ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
Τμήμα Ναυτιλίας και

&

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
Τμήμα Μηχανικών Βιομηχανικής

ΔΙΙΔΡΥΜΑΤΙΚΟ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

«ΝΕΕΣ ΤΕΧΝΟΛΟΓΙΕΣ ΣΤΗ ΝΑΥΤΙΛΙΑ ΚΑΙ ΤΙΣ ΜΕΤΑΦΟΡΕΣ»

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Έλεγχος θερμοκρασίας χώρου με PID

(Calorifier system)

Προτεινόμενος Τίτλος Αγγλικά:

Temperature control using PID

(Calorifier system)

Ονοματεπώνυμο Σπουδαστή:

Γκούφας Γεώργιος

Ονοματεπώνυμο Υπεύθυνου Καθηγητή:

Γρηγόρης Νικολάου

Φεβρουάριος 2019

ΤΙΤΛΟΣ

Έλεγχος θερμοκρασίας με PID

(Calorifier system)

ΟΝΟΜΑ ΦΟΙΤΗΤΗ

ΓΚΟΥΦΑΣ ΓΕΩΡΓΙΟΣ

Μεταπτυχιακή Διατριβή που υποβάλλεται στο καθηγητικό σώμα για την μερική εκπλήρωση των υποχρεώσεων απόκτησης του μεταπτυχιακού τίτλου του Διιδρυματικού Προγράμματος Μεταπτυχιακών Σπουδών «Νέες Τεχνολογίες στη Ναυτιλία και τις Μεταφορές» του Τμήματος Ναυτιλίας και Επιχειρηματικών Υπηρεσιών του Πανεπιστημίου Αιγαίου και του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής του Πανεπιστημίου Δυτικής Αττικής.

Δήλωση συγγραφέα διπλωματικής διατριβής

Ο/Η κάτωθι υπογεγραμμένος / η ΓΚΟΥΦΑΣ ΓΕΩΡΓΙΟΣ……., του ΕΥΑΓΓΕΛΟΥ……, με αριθμό μητρώου ……36…….. φοιτητής / τρια του. Διιδρυματικού Προγράμματος Μεταπτυχιακών Σπουδών «Νέες Τεχνολογίες στη Ναυτιλία και τις Μεταφορές» του Τμήματος Ναυτιλίας και Επιχειρηματικών Υπηρεσιών του Πανεπιστημίου Αιγαίου και του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι: *«Είμαι συγγραφέας αυτής της μεταπτυχιακής διπλωματικής διατριβής και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην διατριβή. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η διατριβή προετοιμάστηκε από εμένα προσωπικά ειδικά για τη συγκεκριμένη μεταπτυχιακή διπλωματική διατριβή».*

Ο δηλών Ημερομηνία

Γκούφας Γεώργιος 2/22019

ΠΕΡΙΛΗΨΗ

Ένα από τα βασικότερα εργαλεία στην επιστήμη του αυτοματισμού είναι ο PID ελεγκτής. Πρόκειται για ένα σύστημα βρόγχου ανάδρασης που χρησιμοποιείται πολύ συχνά στα συστήματα αυτομάτου ελέγχου. Σκοπός του ελεγκτή είναι ελέγχοντας τις εισόδους του συστήματος μας να ελαχιστοποιήσει το σφάλμα στ η λειτουργία του. Με έναν ελεγκτή PID μπορούμε να ρυθμίσουμε τρείς διαφορετικές παραμέτρους για να πετύχουμε την ιδανική λειτουργία του: το αναλογικό (P), το ολοκληρωτικό (I) και το διαφορικό (D).Στην εργασία αυτή θα επιχειρήσουμε να κάνουμε έλεγχο θερμοκρασίας με PID μέσω ενός PLC της scneider electric χρησιμοποιώντας μια ψηφιακή έξοδο PWM. Στα κεφάλαια 3.4 και 5 θα δούμε θεωρητικά τον ελεγκτή PID τις δράσεις του ελεγκτή και τους τρόπους επιλογής των παραμέτρων του. Στο κεφάλαιο 6 θα δούμε την μέθοδο PWM. Ενώ στα κεφάλαια 7 και 8 θα δούμε αναλυτικά τα στάδια της κατασκευής μας, από τη επιλογή των υλικών και την καλωδίωση αυτών μέχρι φυσικά στο βασικότερο κομμάτι που είναι αυτό του προγραμματισμού.

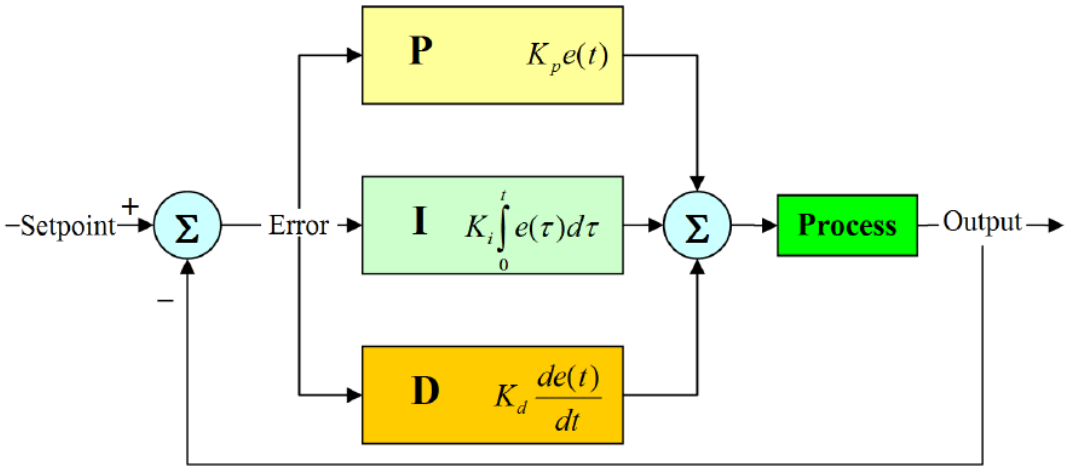
Περιγραφή εφαρμογής

Μια από τις πιο κοινές εφαρμογές που μπορούμε να συναντήσουμε με έλεγχο θερμοκρασίας στη ναυτιλία είναι το calorifier system. Με το συγκεκριμένο σύστημα διατηρούμε το γλυκό νερό του καραβιού, το οποίο προορίζεται για καθημερινή χρήση, στην επιθυμητή θερμοκρασία καθόλη την διάρκεια του εικοσιτετράωρου. Το σύστημα μας αποτελείται από ένα μεγάλο δοχείο νερού του οποίου τη θερμοκρασία μετράμε με ένα αισθητήριο(pt 100). Η θέρμανση του νερού γίνετε με αντίσταση η οποία θα ελέγχεται από τον PID μας με PWM. Στην εξομοίωση της εφαρμογής που θα κάνουμε σε αυτή την εργασία το μέσο το οποίο θα θερμάνουμε θα είναι ατμοσφαιρικός αέρας μέσα σε ένα δοχείο και το θερμαντικό μας σώμα θα είναι ένας κοινός λαμπτήρας. Το βασικό μέρος της εφαρμογής μας είναι ο έλεγχος του θερμαντικού σώματος.

ΚΕΦΑΛΑΙΟ 1

Ενότητα 1.1 Ο PID ελεγκτής

Ο ελεγκτής PID είναι ένα πολύ σημαντικό κομμάτι των συστημάτων αυτομάτου ελέγχου. Αποτελεί στην ουσία ένα μικρότερο σύστημα μέσα στο κυρίως σύστημα το οποίο προσπαθεί να μηδενίσει την διαφορά ανάμεσα στη πραγματική και στην ιδεατή τιμή της εξόδου έτσι ώστε το τελικό σφάλμα να είναι, εάν όχι μηδενικό, ελάχιστα αντιληπτό από το κεντρικό μας σύστημα. Τα αρχικά PID μεταφράζονται ως εξής: Proportional(αναλογικός), Integral(ολοκληρωτικός) και derivative(διαφορικός) δηλαδή το σφάλμα εισόδου πολλαπλασιάζεται με μια σταθερά, ολοκληρώνεται και παραγωγίζεται. Τα αποτελέσματα αυτών των πράξεων στη συνέχεια αθροίζονται και αποτελούν την πραγματική είσοδο του συστήματος όπως φαίνετε στο παρακάτω σχήμα:



Μέχρι την δεκαέτια του ’40 ο PID είχε χρησιμοποιηθεί σε πολλές απαιτητικές εφαρμογές και είχαν δημιουργηθεί τρία βασικά θέματα προς επίλυση:

1) Απλοί τρόποι για την επιλογή των παραμέτρων του ελεγκτή.
2) Να γίνουν πιο ανεξάρτητες οι δράσεις του ελεγκτή.
3) Να πεισθούν οι σχεδιαστές για την αναγκαιότητα της χρήσης των ελεγκτών.

Ανάλογα με το σύστημα που θέλουμε να ελέγξουμε συναντάμε και διαφορετικές μορφές PID. Όταν για παράδειγμα έχουμε μικρές καθυστερήσεις χρόνου ή το κλειστό σύστημα δεν χρειάζεται να είναι αρκετά γρήγορο συνήθως χρησιμοποιούμε ελεγκτή τύπου PI ενώ όταν έχουμε συστήματα τύπου ένα, λόγω του μηδενικού σφάλματος στην μόνιμη κατάσταση χρησιμοποιούμε ελεγκτή τύπου PD.

Ένας ελεγκτής PID έχει σημαντικά προτερήματα: έχει μηδενικό σφάλμα στη μόνιμη κατάσταση μεσω της ολοκληρωτικής του δράσης, μπορεί να προβλέψει μια κατάσταση μέσω της διαφορικής του δράσης και φυσικά παρέχει ανάδραση.

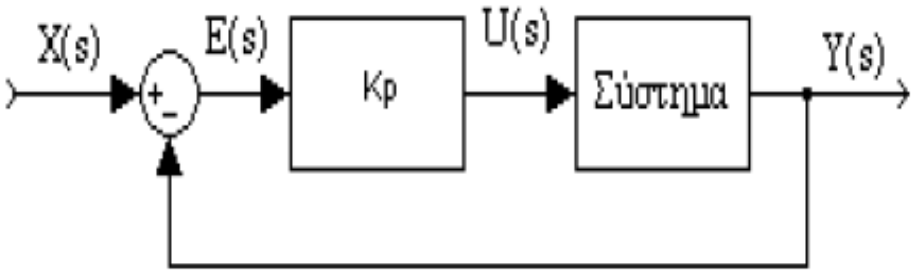
Η χρήση για την κάθε δράση του ελεγκτή ξεχωριστά φαίνεται καλύτερα στο παρακάτω σχέδιο:

Τύπος Ελεγκτή	Χρόνος Ανόδου	Υπερύψωση	Χρόνος Αποκατάστασης	Μόνιμο σφάλμα
P	Μείωση	Αύξηση	Μικρή Αλλαγή	Μείωση
I	Μείωση	Αύξηση	Αύξηση	Εξάλειψη
D	Μικρή Αλλαγή	Μείωση	Μείωση	Μικρή Αλλαγή

Ενότητα 1.2 Είδη ελεγκτών

1)Αναλογικός (P) ελεγκτής

Ο αναλογικός ελεγκτής στην ουσία ενισχύει το σφάλμα που δέχεται το σύστημα στην εισοδό του με κέρδος Kp. Ο αναλογικός ελεγκτής έχει τη μορφή του παρακάτω σχήματος:



Εικ 1. (Συστήματα αυτομάτου ελέγχου:Παντελής Μαλατέστας)

Η συνάρτηση μεταφοράς του αναλογικού ελεγκτή είναι η εξής:

$$P_{(s)} = K_p$$

Ο αναλογικός ελεγκτής αυξάνει την ταχύτητα απόκρισης του συστήματος αλλά μπορεί να δημιουργεί σφάλμα μόνιμης κατάστασης.Για μηδενικό σφάλμα θα έχει και μηδενική έξοδο.

Όταν το σύστημα έχει συνάρτηση μεταφοράς:

$$G_{(s)} = \frac{1}{s + A}$$

τότε η συνάρτηση μεταφοράς του κλειστού βρογχου με P ελεγκτή θα είναι:

$$G_{c(s)} = \frac{\frac{K_p}{s+A}}{1+\frac{K_p}{s+A}} \rightarrow G_{c(s)} = \frac{K_p}{s+A+K_p}$$

Άρα εάν διεγείρουμε το σύστημα με μοναδιαία βηματική είσοδο θα βρούμε το σφάλμα της μόνιμης κατάστασης από τη σχέση:

$$e_{ss} = 1 - \lim_{s\to 0}\left(s * G_{c(s)} * \frac{1}{s}\right) = 1 - \lim_{s\to 0}\frac{k_p}{s+A+k_p} \rightarrow e_{ss} = 1 - \frac{kp}{A+k_p}$$

Επομένως το σφάλμα μόνιμης κατάστασης θα μειωθεί όσο αυξάνουμε το κέρδος του αναλογικού ελεγκτή όμως στη πραγματικότητα αυτό δεν είναι εφικτό.

Εάν η συνάρτηση μεταφοράς μας ήταν η εξής:

$$G_{(s)} = \frac{1}{s^2 + s * A + B}$$

τότε η συνάρτηση του κλειστού συστήματος με P ελεγκτή θα ήταν η παρακάτω:

$$G_{c(s)} = \frac{\frac{K_p}{s^2 + s * A + B}}{1+\frac{K_p}{s^2 + s * A + B}} \rightarrow G_{c(s)} = \frac{K_p}{s^2 + s * A + B + K_p}$$

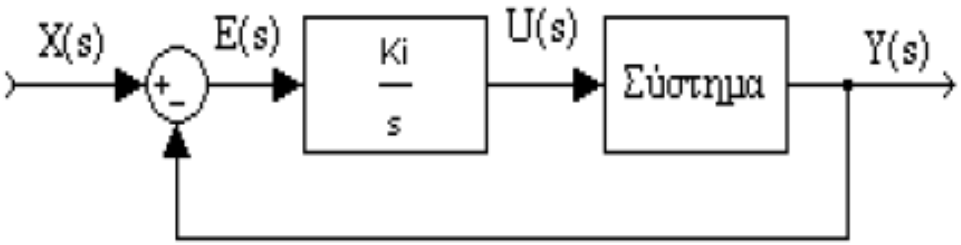
Αντιστοιχα με μια μοναδιαία βηματική είσοδο θα βρίσκαμε πάλι το σφάλμα μόνιμης κατάστασης από τη σχέση:

$$e_{ss}=1-\lim_{s\to 0}\left(s*G_{c(s)*}\frac{1}{s}\right)=1-\lim_{s\to 0}\frac{k_p}{s^2+S*A+B+Kp}\to e_{ss}=1-\frac{Kp}{B+Kp}$$

Σε αυτή τη περίπτωση το σφάλμα του συστήματος μειώνετε όσο αυξάνετε το κέρδος του ελεγκτή και η φυσική συχνότητα $\omega_n(=\sqrt{B+k_p}$ ενώ χωρίς τον ελεγκτή θα ήταν $\omega_n=\sqrt{B}$) κάτι που δίνει ταχύτερη απόκριση στο σύστημα.Όταν έχουμε αναλογικό ελεγκτή το A είναι ίσο με 2*ζ*ω_n όπως και σε σύστημα χωρίς ελεγκτή και παραμένει σταθερό με τη διαφορά ότι αυξάνεται η φυσική συχνότητα ενώ μειώνεται ι συντελεστής απόσβεσης ζ με αποτέλεσμα να έχουμε αύξηση της υπερύψωσης στη βηματική απόκριση.Έτσι όταν αυξάνουμε το κέρδος του ελεγκτή Kp μπορεί να μειώνουμε το σφάλμα στο σύστημα αλλά δημιουργούμε μεγάλη υπερύψωση και σημαντικές ταλαντώσεις στο σύστημα.

2)Ολοκληρωτικός (I) ελεγκτής

Ο ολοκληρωτικός ελεγκτής (Integral) μας δίνει στην έξοδο του συστήματος ένα σήμα ανάλογο του ολοκληρώματος του σφάλματος που δέχεται στην είσοδο.Ο ολοκληρωτικός ελεγκτής περιέχει και τον συντελεσυή ολοκλήρωσης ο οποίος συμβολίζεται με Ki . Η μορφή του ολοκληρωτικού ελεγκτή είναι η εξής:



Εικ.2 (Συστήματα αυτομάτου ελέγχου: Παντελής Μαλατέστας)

Η συνάρτηση μεταφοράς του ελεγκτή είναι:

$$I_{(s)}=\frac{K_i}{s}$$

Για βηματική διέγερση στον ολοκληρωτικό ελεγκτή έχουμε κάποια καθυστέρηση στο χρόνο μέχρι να φτάσουμε την είσοδο και όταν το σφάλμα γίνει μηδενικό τότε έχουμε σταθερή έξοδο.

Όταν η συνάρτηση μεταφοράς του συστήματος είναι η εξής:

$$G_{(s)} = \frac{1}{s + A}$$

τότε η συνάρτηση μεταφοράς του συστήματος κλειστού βρόγχου με ελεγκτή I είναι η εξής:

$$G_{c(s)} = \frac{\frac{K_i}{s*(s+A)}}{1+\frac{K_i}{s*(s+A)}} \rightarrow G_{c(s)} = \frac{K_i}{s*(s+A)+K_i} \rightarrow G_{c(s)} = \frac{K_i}{s^2 + A*s + K_i}$$

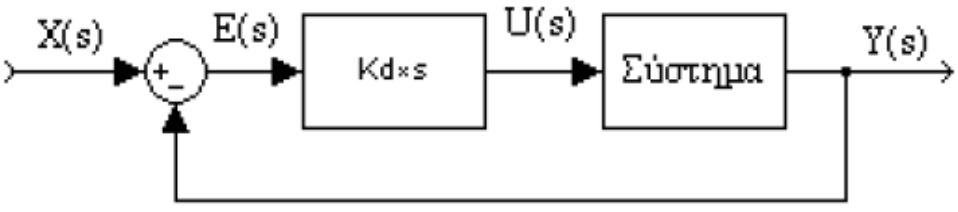
Οπότε εάν είχαμε μοναδιαία βηματική είσοδο το σφάλμα μόνιμης κατάστασης θα δινόταν από τη σχεση:

$$e_{ss} = 1 - \lim_{s\to 0}(s*G_{c(s)}*\frac{1}{s}) = 1 - \lim_{s\to 0}\frac{K_i}{s^2 + A*s + K_i} \rightarrow e_{ss} = 1 - \frac{K_i}{K_i} \rightarrow e_{ss} = 0$$

Που σημαίνει ότι το σφάλμα μόνιμης κατάστασης μηδενίζεται με τη χρήση ολοκληρωτικού ελεγκτή. Παρατηρούμε ότι με την αύξηση του συντελεστή ολοκλήρωσης αυξάνεται η φυσική συχνότητα του συστήματος με αποτέλεσμα να έχουμε ταχύτερη απόκριση του συστήματος και μειώνεται ο συντελεστής απόσβεσης του συστήματος αφού το γινόμενο 2*ζ*ω_n παραμένει σταθερό και ίσο με A, με συνέπεια την αύξηση της υπερύψωσης στη βηματική απόκριση. Όταν έχουμε σταθερή είσοδο αναφοράς το σφάλμα μόνιμης κατάστασης εξαλείφετε.

3)Διαφορικός (D) ελεγκτής

Στον διαφορικό ελεγκτή (differential) όταν έχουμε σήμα σφάλματος εισόδου με μορφή βηματικής διέγερσης τότε η έξοδος του ελεγκτή είναι η κρουστική συνάρτηση με θεωρητικά άπειρο πλάτος για t=0. Όταν το σφάλμα είναι σταθερό η έξοδος του ελεγκτή είναι μηδέν.
Η μορφή του διαφορικού ελεγκτή είναι η εξής:



Εικ 3. (Συστήματα αυτομάτου ελέγχου: Αναστασία Βελώνη)

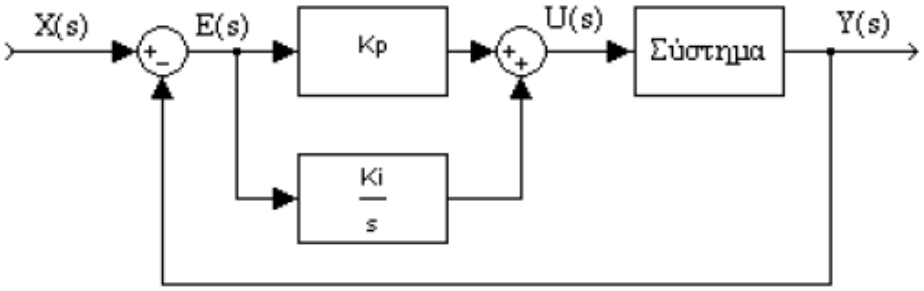
Η συνάρτηση μεταφοράς του διαφορικού ελεγκτή είναι:

$$D_{(s)} = K_d * s$$

Ο διαφορικός ελεγκτής περιορίζει το σφάλμα στη μόνιμη κατάσταση όμως στη πράξη πότε δεν χρησιμοποιείτε αποκλειστικά μόνος του.

4)Αναλογικός – Ολοκληρωτικός (PI) ελεγκτής

Η μορφή ενός αναλογικού – ολοκληρωτικού ελεγκτή είναι η εξής:



Εικ. 5 (Συστήματα αυτομάτου ελέγχου : Αναστασία Βελώνη)

Η συνάρτηση μεταφοράς του ελεγκτή που προκύπτει από το παραπάνω σχέδιο είναι η εξής:

$$PI_{(s)} = K_p + \frac{K_i}{s} = \frac{s*K_p + K_i}{s} \rightarrow PI_{(s)} = K_p * \frac{s + \frac{K_i}{K_p}}{s}$$

Ο ολοκληρωτικός έλεγχος έχει ως σκοπό την εξάλειψη του σφάλματος στη μόνιμη κατάσταση ενώ ο αναλογικός αυξάνει την ευστάθεια του συστήματος μας και αυξάνει την ταχύτητα απόκρισης.
Εάν έχουμε ένα σύστημα πρώτης τάξης με συνάρτηση μεταφοράς:

$$G_{(s)} = \frac{1}{s + A}$$

τότε η συνάρτηση μεταφοράς του κλειστού βρόγχου θα είναι:

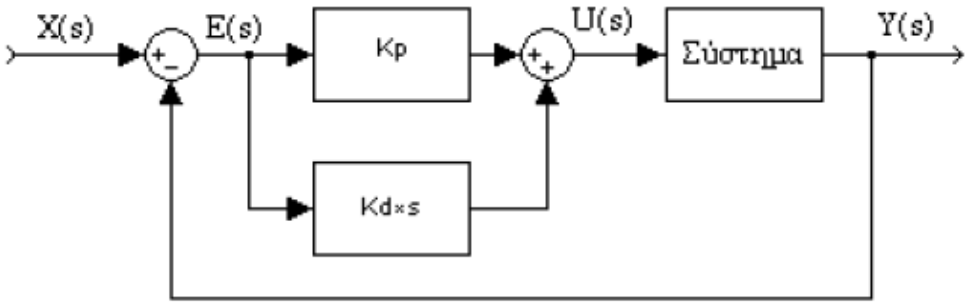
$$G_{c(s)} = \frac{\frac{s*K_p + K_i}{s} * \frac{1}{s+A}}{1 + \frac{s*K_p + K_i}{s} * \frac{1}{s+A}} \rightarrow G_{c(s)} = \frac{\frac{s*K_p + K_i}{s*(s+A)}}{\frac{s*(s+A) + s*K_p + K_i}{s*(s+A)}} \rightarrow$$

$$\rightarrow G_{c(s)} = \frac{s*K_p + K_i}{s^2 + A*s + s*K_p + K_i} \rightarrow G_{c(s)} = \frac{s*K_p + K_i}{s^2 + (A + K_p)*s + K_i}$$

Οπότε τώρα έχουμε: ω_n=K_ι και 2*ζ*ω_n=A+K_p
Άρα χρησιμοποιώντας έναν PI ελεγκτή αυξάνωντας τον ολοκληρωτικό όρο Ki αυξανεται και η φυσική συχνότητα του συστήματος ενώ αυξάνωντας τον αναλογικό συντελεστή Kp αυξάνεται και το γινόμενο 2*ζ*ω_n χωρίς να μεταβάλλεται η ω_n οπότε έχουμε μεγαλύτερο συντελεστή απόσβεσης δηλαδή μικρότερη υπερύψωση.Το πόσο θα επηρρεάσει το σύστημα ο ελέγκτής εξαρτάται από το μηδενικό s=$\frac{Ki}{Kp}$ που εισάγει ο ελεγκτής στο σύστημα.Για βηματική είσοδο έχουμε μηδενικό σφάλμα μόνιμης κατάστασης.

5)Αναλογικός – Διαφορικός (PD) ελεγκτής

Η μορφή του αναλογικου διαφορικού ελεγκτή είναι η εξής:



Εικ. 4 (Συστήματα αυτομάτου ελέγχου : Αναστασία Βελώνη)

Η συνάρτηση μεταφοράς του ελεγκτή που πρόκυπτει είναι:

$$PD_{(s)} = K_p + s * K_d \rightarrow PD_{(s)} = s + \frac{K_p}{K_d}$$

Κάνοντας PD έλεγχο μειώνουμε την υπερύψωση αφού έχουμε καλυτερη απόσβεση του συστήματος χωρίς να μειώνουμε άμεσα το σφάλμα μόνιμης κατάστασης.Μέσω του διαφορικού όρου έχουμε μια αύξηση στην απόσβεση του συστήματος η οποία επιτρέπει τη διόρθωση του σφάλματος μόνιμης κατάστασης και κατά συνέπεια την καλύτερη ταχύτητα απόκρισης του συστήματος.Αυτού του είδους ο έλεγχος μπορεί να προκαλέσει θόρυβο στην απόκριση του συστήματος.Επίση απότομες αλλάγες στο σήμα αναφοράς μπόρει να προκαλέσουν προβλήματα γιατί ο διαφορικός όρος παράγει έξοδο κρουστικής μορφής.
Εάν το σύστημα έχει συνάρτηση μεταφοράς:

$$G_{(s)} = \frac{1}{s^2 + A * s + B}$$

τότε η συνάρτηση μεταφοράς του κλειστου βρόγχου με PD ελεγκτή θα είναι:

$$G_{c(s)} = \frac{(K_p + s * K_d) * \frac{1}{s^2 + A * s + B}}{1 + (K_p + s * K_d) * \frac{1}{s^2 + A * s + B}} \rightarrow$$

$$\rightarrow G_{c(s)} = \frac{\frac{(K_p + s * K_d)}{s^2 + A * s + B}}{\frac{s^2 + A * s + B + K_p + s * K_d}{s^2 + A * s + B}} \rightarrow$$

$$\rightarrow G_{c(s)} = \frac{K_p + s * K_d}{s^2 + (A + K_d) * s + (B + K_p)}$$

Οπότε τώρα ισχύουν τα εξής:

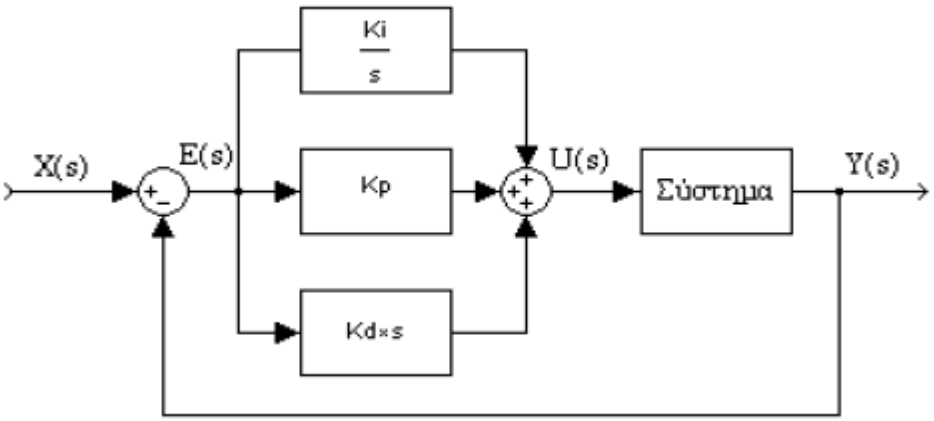
$$\omega_n = B + K_p \text{ και } 2 * \zeta * \omega_n = A + K_d$$

Άρα όταν αυξάνεται το αναλογικό κέρδος Kp έχουμε αύξηση της φυσικής συχνότητας άρα ταχύτερη απόκριση του συστήματος.Η αύξηση τοθ διαφορικού συντελεστή Kd p προκαλεί αύξηση του γινομένου $2*\zeta*\omega_n$ κρατόντας σταθερή τη φυσική συχνότητα δλδ αυξημένος συντελεστής απόσβεσης και μείωση της υπερύψωσης της απόκρισης του συστήματος.Αυξάνοντας το αναλογικό κέρδος μειώνεται το σφάλμα μόνιμης κατάστασης χωρίς να επηρρεάζεται από τη μεταβολή του διαφορικού συντελεστή.

6)Αναλογικός - Ολοκληρωτικός – Διαφορικός (PID) ελεγκτής

Ο PID ελεγκτής είναι ευρέως διαδεδομένος σε πολλά είδη εφαρμογών λόγω της απλότητας του η οποία επιτρέπει ευκολό και αξιόπιστο χειρισμό αλλά και της σωστής συμπεριφοράς του σε μεγάλη κλίμακα συνθηκών.

Η μορφή ενός PID ελεγκτή έχει ως εξής:



Εικ. 6 (Συστήματα αυτομάτου ελέγχου : Αναστασία Βελώνη)

Η συνάρτηση μεταφοράς του ελεγκτή που προκύπτει ειναι:

$$PID_{(s)} = K_p + s*K_d + \frac{K_i}{s} \rightarrow PID_{(s)} = \frac{K_d*s^2 + K_p*s + K_i}{s}$$

Ένας PID ελεγκτής έχει την αρτιότερη απόδοση σε ένα σύστημα καθώς παρουσιάζει την ταχύτερη απόκριση,μηδενική υπερύψωση και μηδενικο σφάλμα μόνιμης κατάστασης.Πολύ σημαντικό στην λειτουργία ενός PID ελεγκτή είναι η σωστή επιλογή και των τριών παραμέτρων Kp,Ki και Kd έτσι ώστε να μην επιδρά αρνητικά η μια πάνω στην άλλη.

Κεφαλαιο 2

2.1 ΕΛΕΓΧΟΣ ΘΕΡΜΟΚΡΑΣΙΑΣ ΜΕ PLC

Με τη συγκεκριμένη εφαρμογή θα ελέγχουμε τη θερμοκρασία ενός χώρου μέσω PID που μας παρέχει το ίδιο το PLC. Ο PID μας θα έχει τέσσερις διαφορετικές λειτουργίες οι οποίες θα εμφανίζονται σε μια interractive οθόνη.Στην οθόνη ο χρήστης περάν των λειτουργιών θα μπορεί να επέμβει πλήρως στο σύστημα είτε αυτό είναι setpoint είτε είναι μεταβλητές του ελεγκτή.Επίσης θα έχει προσβαση σε σελίδα με τα trend του συστήματος αλλα και σε σελίδα που θα αναγράφονται οι κρίσιμες καταστάσεις του συστήματος.Η έξοδος του pid μας θα είναι σε μορφή παλμου PWM.

Στην ναυτηλία είναι πάρα πολλές οι εφαρμογές που γίνετε έλεγχος PID μέσω PLC. Τέτοιου είδους έλεγχο μπορούμε να συνατήσουμε σε boiler ,σε VFD , σε Coollers κλπ. Πολύ συνηθισμένη είναι η χρήση του στα Calorifier (στο σύστημα που διατηρεί το νερό του καραβιού ζεστό). Η διαφορά της κατασκευής μας σε σχέση με ένα calorifier system είναι ότι συνήθως στο καράβι χρησιμοποιούν αντίσταση ενώ εμείς θα χρησιμοποιήσουμε λαμπτήτα και φυσικά στο καράβι ο χώρος είναι ένα δοχείο με νερό.

2.2 ΠΕΡΙΓΡΑΦΗ ΚΑΤΑΣΚΕΥΗΣ ΚΑΙ ΥΛΙΚΩΝ

Τα υλικά που χρησιμοποιήθηκαν για την εργασία μας είναι τα εξής:

- Τροφοδοτικό 24VDC

Το τροφοδοτικό που χρησιμοποιούμε είναι το ABL8REM24030 και είναι της scneider electric. Τα 24V στην συγκεκριμένη εργασία στην ουσία τα χρησιμοποιούμε για να τροφοδοτήσουμε την CPU μας, την αναλογική κάρτα μας, την οθόνη μας και τα ενδεικτικά λαμπάκια μας. Το τροφοδοτικό φαίνεται στην παρακάτω φώτο:

Εικ .6 www.scneiderelectric .com

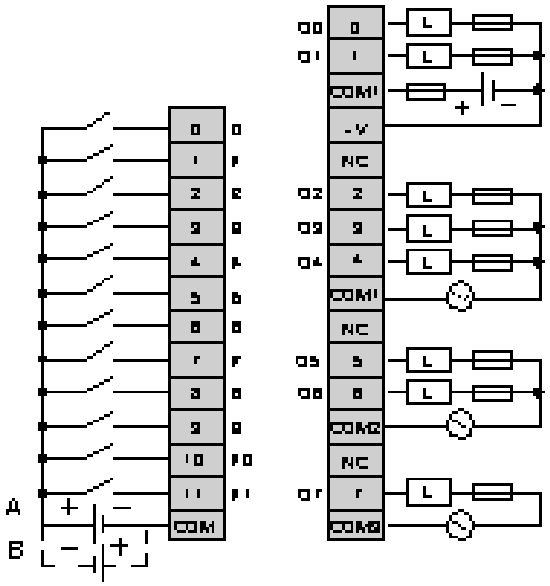
Στις πάνω κλέμμες του παίρνει την τροφοδοσία δικτύου(220V) και από τις κάτω κλέμμες του παιρνουμε τα 24V.

- CPU

Η CPU που χρησιμοποιούμε είναι η TWDLMDA20DRT της scneider electric η οποία φαίνεται παρακάτω:

Η συγκεκριμένη cpu έχει 12 εισόδους 24VDC και 8 εξόδους (2 εξόδους τρανζίστορ και 6 εξόδους relay oyt). Η τροφοδοσία της είναι 24VDC.

Η συνδεσμολογία της cpu έχει ως εξής:



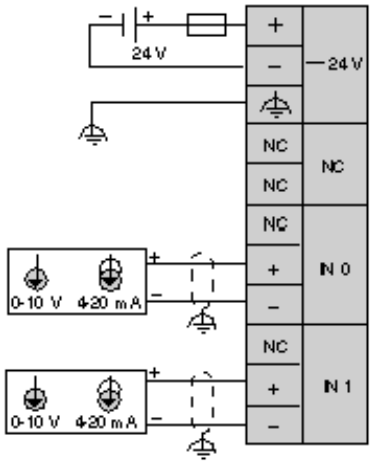
Εικ .7 www.scneiderelectric .com

- Αναλογική κάρτα

Την αναλογική κάρτα την χρησιμοποιούμε μόνο για να πάρουμε το αναλογικό σήμα από το PT100.Η κάρτα που χρησιμοποιούμε είναι η TM2AMI2HT της scneider electric.Η συγκεκριμένη κάρτα έχει δύο αναλογικές εισόδους που σημαίνει ότι μπορούν να συνδεθούν μέχρι δύο σένσορες επάνω της.Στο εκάστοτε κανάλι της μπορεί να διαβάσει σήματα 0-10 V ή 4-20mA.Η αναλογική μας κάρτα απεικονίζεται στην παρακάτω:

Εικ .7 www.scneiderelectric .com

Η συνδεσμολογία της κάρτας έχει ως εξής:



Εικ .8 www.scneiderelectric .com

- Οθόνη

Η οθόνη αφής που χρησιμοποιούμε είναι η XBTZ9780 της scneider electric με τροφοδοσία 24VDC,μεγέθους 5,7 inch. Η οθόνη απεικονίζεται παρακάτω:

Εικ .9 www.scneiderelectric .com

- Καλώδιο επικοινωνίας

Το καλώδιο επικοινωνίας που χρησιμοποιούμε είναι το XBTZ9780 και είναι ένα καλώδιο σειριακής επικοινωνίας modbus το οποίο μας εξασφαλίζει την επικοινωνία μεταξύ της οθόνης και του plc μας.

- PT 100

Το PT 100 είναι το αισθητήριο το οποίο θα μας μεταφέρει το σήμα από το χώρο τον οποίο μετράμε την θερμοκρασία μας στην αναλογική μας κάρτα έτσι ώστε να είναι διαθέσιμο προς επεξεργασία.

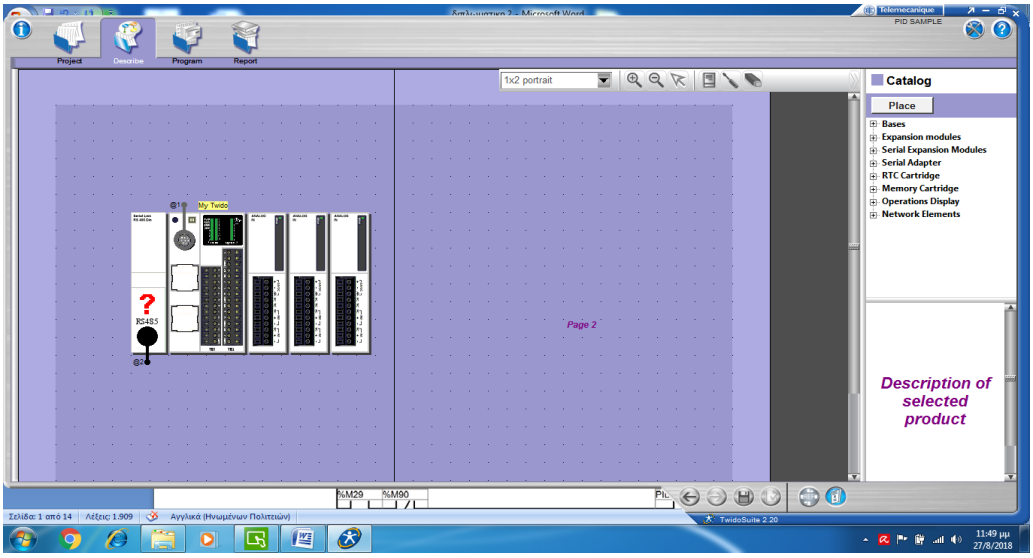
- Ενδεικτικά led

Τα ενδεικτικά led τα οποία χρησιμοποιούμε είναι τα ZBV-B4 χρώματος κόκκινου και τροφοδοσίας 24VDC.Τα led που χρησιμοποιούμε φαίνονται παρακάτω:

2.3 Ανάλυση software PLC.

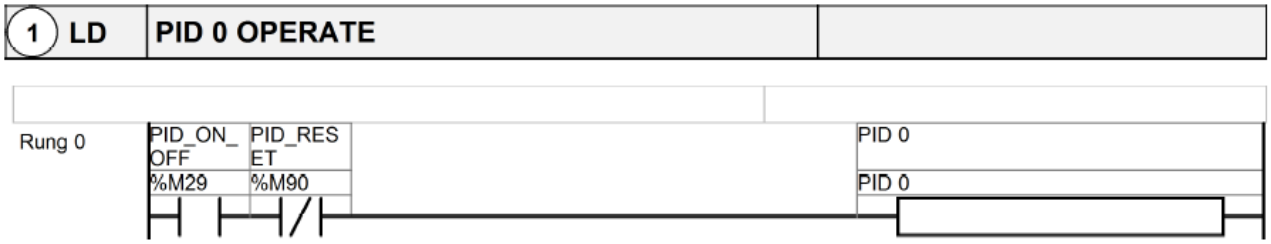
Το πρόγραμμα που θα χρησιμοποιήσουμε για να προγραμματίσουμε το PLC μας είναι το twido suite της scneider electric.Η γλώσσα προγραματισμού που θα χρησιμοποιήσουμε είναι η ladder.

Αφού ανοίξουμε το twido suite κάνουμε <create a new project> τότε θα επιλέξουμε από το πάνω αριστερά μέρος της οθόνης το κουμπί <describe>.Στη σελίδα που θα μας εμφανιστεί πρέπει να ορίσουμε το plc που θα χρησιμοποιήσουμε(cpu και κάρτες) καθώς και τον τρόπο επικοινωνίας που θα χρησιμοποιήσουμε π.χ σε περίπτωση remote χειρισμού. Όλες οι cpu και οι κάρτες είναι διαθέσιμες στην δεξιά μεριά της οθόνης, αρκεί να τις επιλέξουμε και να τις σύρουμε στο κέντρο της σελίδας. Στην συγκεκριμένη εργασία σε αυτή την σελίδα θα εμφανίζονται οι παρακάτω κάρτες:

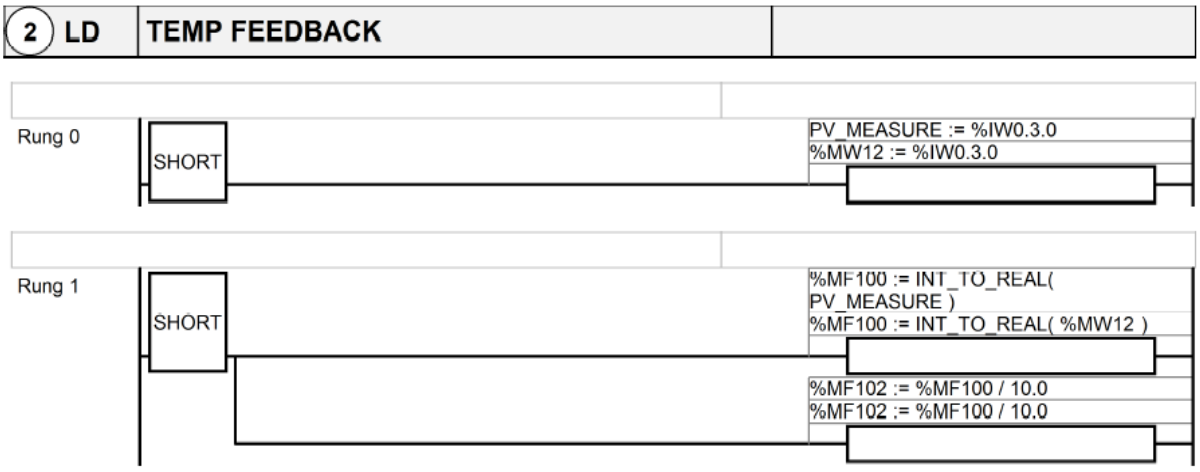


Στη συνέχεια επιλέγοντας το ακριβώς διπλανό εικονίδιο το <program> θα μεταφερθούμε σε μια σελίδα όπου θα μπορούμε να γράψουμε τον κώδικα μας.

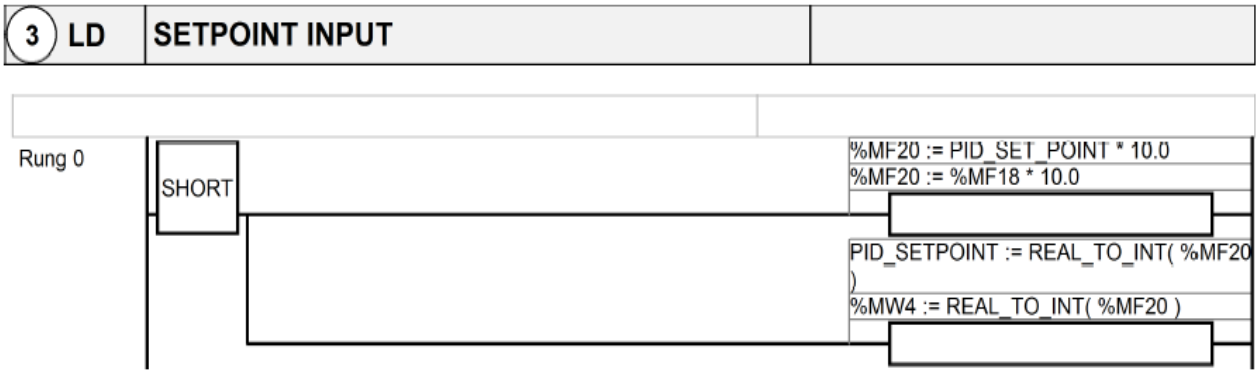
Αναλυτική περιγραφή του κώδικα ακολουθεί παρακάτω:



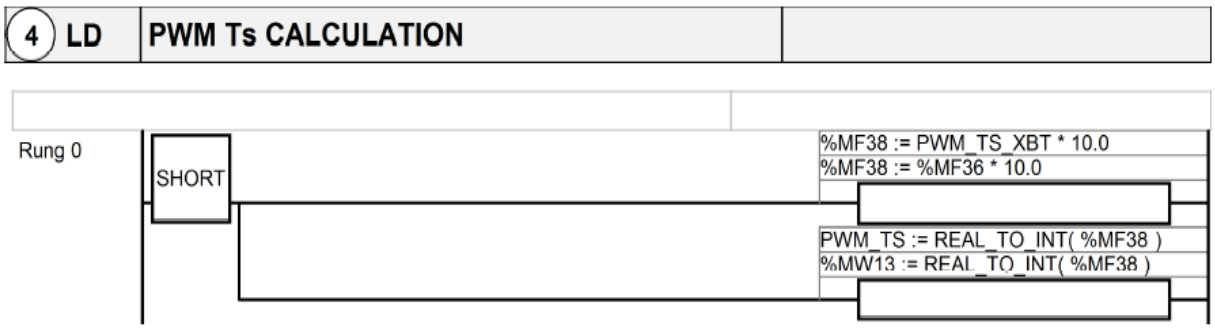
Ξεκινάμε το πρόγραμμα μας θέτοντας πότε θα είναι σε κατάσταση on ή off ο pid μας αλλά και πότε θα γίνετε το reset σε περίπτωση δυσλειτουργίας. Χρησιμοποιούμε μια ανοιχτή επαφή M29 για το on off του ελεγκτή και μια κλειστή M90 για το reset.Μέσω του vijeo designer ορίζουμε την M29 ως toggle,αυτό σημαίνει ότι πατώντας το button on-off η κατάσταση του bit μας θα αλλάζει από μηδέν σε ένα ή από ένα σε μηδέν. Ενώ την M90 την ορίζουμε ως momentary on, το οποίο σημαίνει ότι το bit μας αλλάζει κατάσταση μόνο όσο πατάμε το buttonενώ μόλις αφήσουμε το button το bit επανέρχεται στην αρχική του κατάσταση.



Στο input word %IW0.3.0 έχουμε την μέτρηση της θερμοκρασίας που παίρνουμε από τον PT αισθητήρα μας. Στη συνέχεια έχοντας τη μέτρηση μας στο m word 12 μετατρέπουμε αυτή την τιμή μας σε πραγματικό αριθμό για να μπορέσουμε να κάνουμε τις πράξεις μας και να έχουμε ακρίβεια σε αυτές. Τον πραγματικό μας αριθμό τώρα τον βάζουμε στο MF 100.Το MF σαν χωρητικότητα έχει ίση με 2 m word και πάντα καταλαμβάνει δύο θέσεις στην μνήμη του plc, δλδ οι θέσεις MFO και MF1 αναφέρονται στο ίδιο MF.Η πραγματική μας τιμή τοποθετείται στο MF102 και προκύπτει ύστερα απο τη διαίρεση της τιμής του MF100 με το 10.Η αναλογική μας κάρτα διαβάζει απο -200 εως 6000 και εχει σαν unit το 0,1.Έτσι διαιρούμε με το 10 για να έχουμε την πραγματική μας τιμή μέτρησης(αναλυτικότερα θα το δούμε στο κεφάλαιο που αναλύουμε το configuration τη κάρτας).



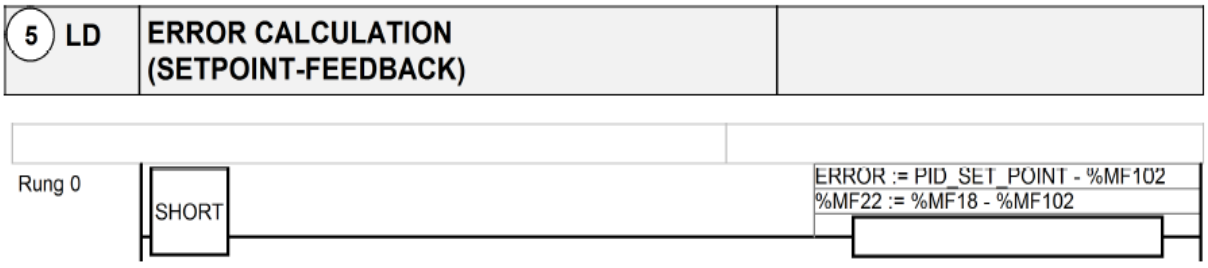
Για να θέσουμε εμείς το setpoint του PID μας μέσω της οθόνης μας πρέπει να κάνουμε ακριβώς την αντίστροφη διαδικασία από αυτήν που κάναμε στο προηγούμενο rung (tempfeedback).Την τιμή που θα του βάλουμε στο setpoint της οθόνης δλδ στο MF18 την πολλαπλασιάζουμε με το 10 έτσι ώστε να την διαβάσει σωστά το PLCκαι το αποτέλεσμα της πράξης την βάζουμε στο MF20.Στη συνέχεια για να βάλουμε το αποτέλεσμα αυτό το οποίο είναι πραγματικός αριθμός σε m word πρέπει να το μετατρέψουμε σε ακέραιο. Αυτό το κάνουμε με την εντολή real to int και το αποτέλεσμα αυτής το βάζουμε στο MW4.



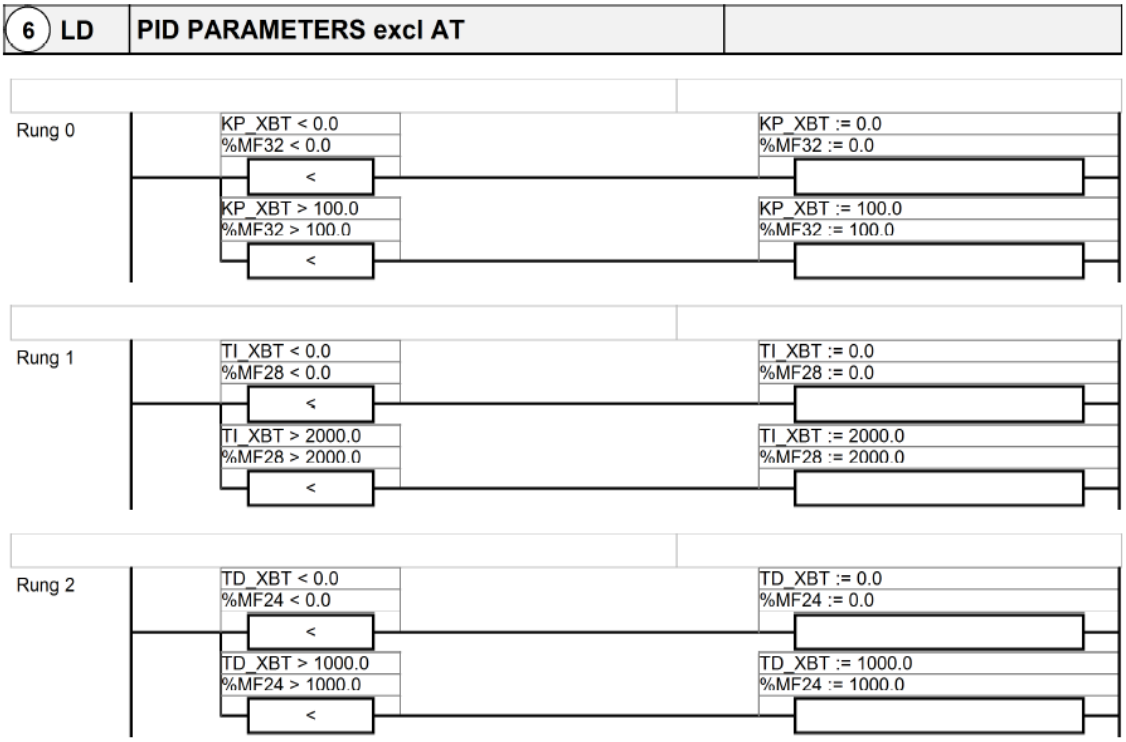
Εάν πάμε στο help του twido soft και επιλέξουμε το PID characteristics θα δούμε τα χαρακτηριστικά του PID μας. Ανάμεσα σε αυτά μας δίνει και τα χαρακτηριστικά της εξόδου του PWM(pwm output).Το unit που μας δίνει για την μέτρηση της περιόδου του pwm είναι 0.1s και το εύρος των τιμών που μπορεί να πάρει είναι από 1 έως 500.Αυτό σημαίνει ότι η περίοδος κυμαίνετε μεταξύ 1 και 50 ms.

Έτσι στο rung 0 αυτό που ορίζουμε το pwm το βάζουμε στο MF36.Πολλαπλασιάζουμε αυτή την τιμή με το 10 αφού το unit μας είναι 0,1 έτσι ώστε να διαβαστεί σωστά από το plc και το βάζουμε στο MF38.Στη συνέχεια μετατρέπουμε τον πραγματικό αριθμό σε ακέραιο και τον βάζουμε στο m word MW13.

Στο rung 1 οριοθετούμε τις τιμές που μπορεί να πάρει το PWM βάσει χαρακτηριστικών του. Ορίζουμε ότι εάν το MF36 είναι μικρότερο της μονάδας τότε να γίνει ίσο την μονάδα. Ενώ εάν είναι μεγαλύτερο από 50s να γίνει ίσο με 50s.



Στο error calculation υπολογίζουμε τη διαφορά μεταξύ του setpoint που έχουμε ορίσει στην οθόνη μας και της πραγματικής τιμής που παίρνουμε από τον αισθητήρα μας. Το MF18 είναι η τιμή του setpoint και το MF102 είναι η πραγματική μας τιμή. Τη διαφορά τους την ορίζουμε ως MF22.

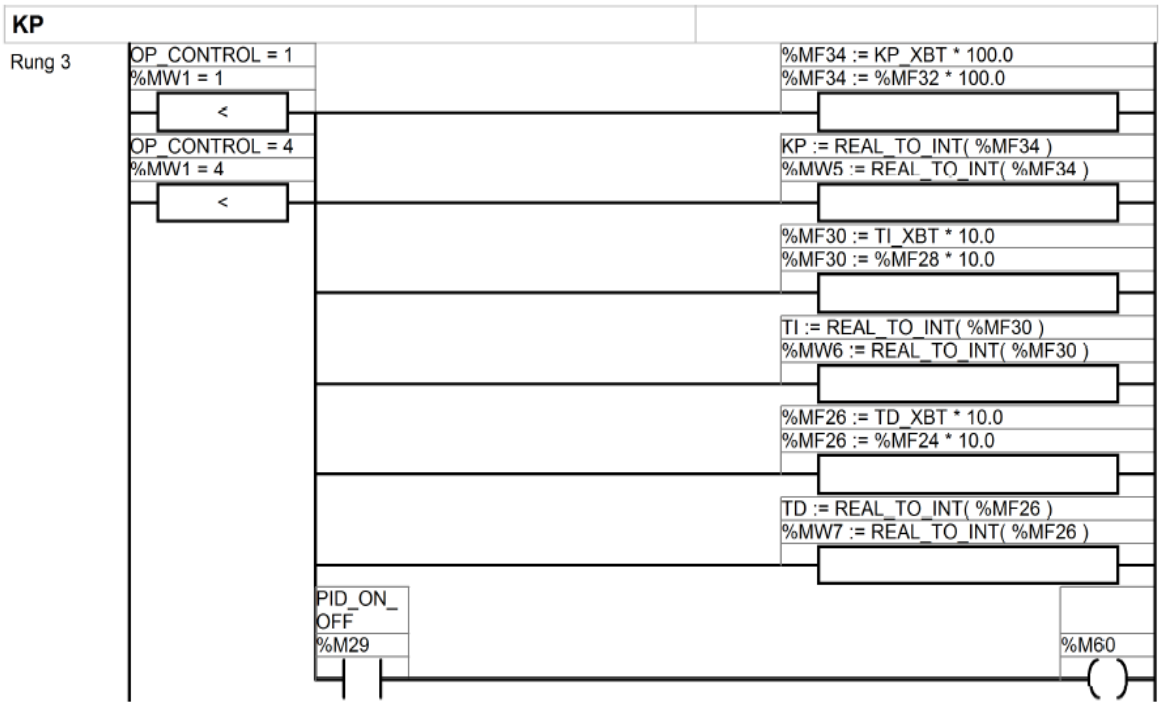


Πάλι βάση των χαρακτηριστικών του PID μας τα οποία μας δίνει το ίδιο το πρόγραμμα πρέπει να ορίσουμε το εύρος των τιμών μεταξύ των οποίων θα κινούνται οι τιμές των Kp,Ti και Td όταν τις ορίζουμε αυτές χειροκίνητα. Το εύρος των τιμών που μας δίνει το twidosoft είναι για το Kp : 0 εως 100 για το Ti :0 εως 2000 και για το Td :0 εως 1000.

Έτσι στο rung 0 κάνουμε τη σύγκριση της MF32(στην οποία ορίζουμε το Kp) έτσι ώστε εάν είναι μικρότερη του μηδενός να γίνει ίση με μηδεν και εαν είναι μεγαλύτερη του 100 να γίνει ίση με 100.

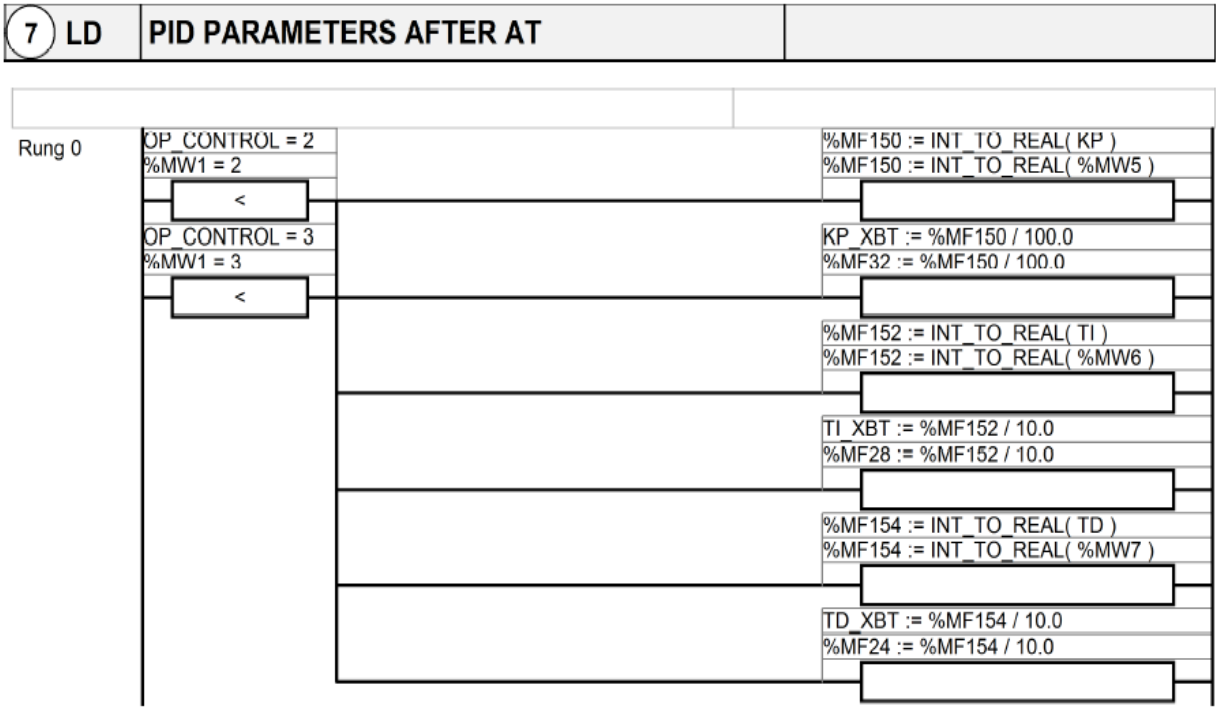
Στο rung 1 κάνουμε τη σύγκριση της MF28(στην οποία ορίζουμε το Ti) έτσι ώστε εάν είναι μικρότερη του μηδενός να γίνει ίση με μηδέν και εαν είναι μεγαλύτερη του 2000 να γίνει ίση με 2000.

Στο rung 2 κάνουμε τη σύγκριση τηςMF24(στην οποία ορίζουμε το Td) έτσι ώστε εάν είναι μικρότερη του μηδενός να γίνει ίση με μηδέν και εάν είναι μεγαλύτερη του 1000 να γίνει ίδη με 1000



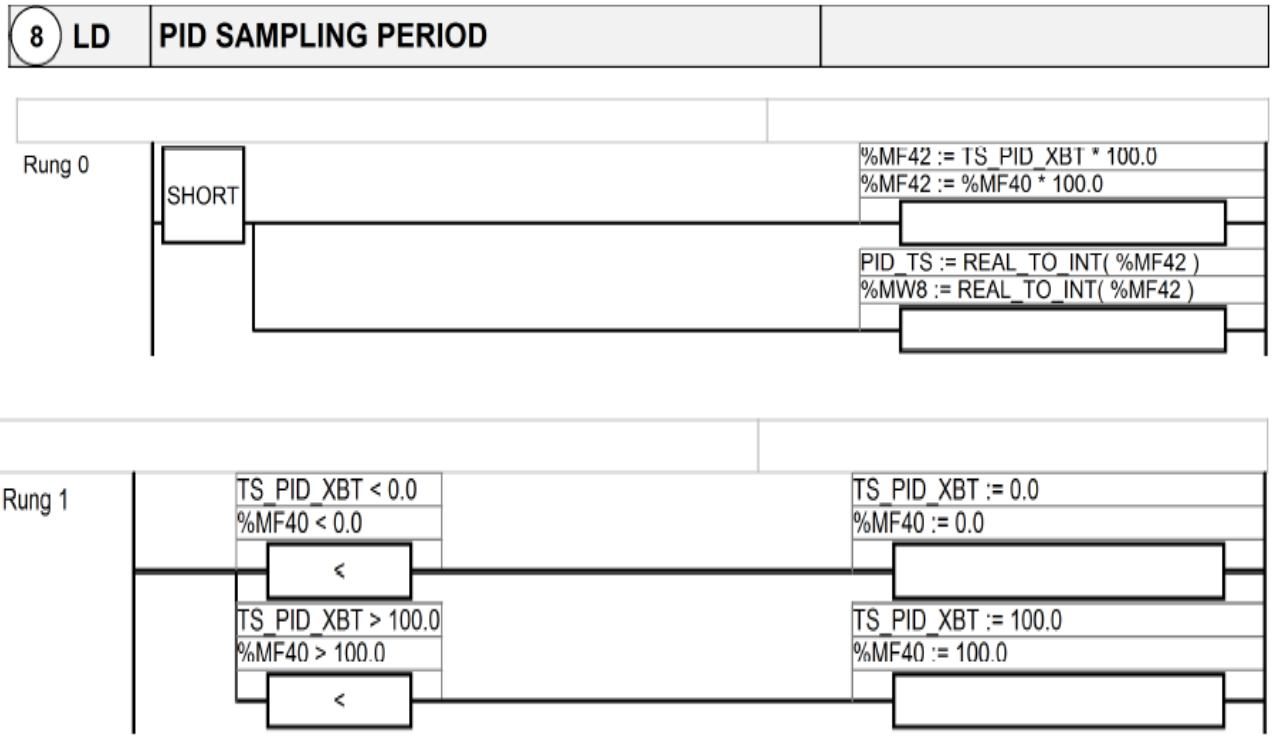
Στο rung 3 υπολογίζουμε τις παραμέτρους Kp ,Ti και Td βάση των χαρακτηριστικών τους μόνο όταν βρισκόμαστε σε PID ή PI λειτουργία. Από το general configurationτου PID μας μπορούμε να επιλέξουμε τη λειτουργία που θέλουμε να κάνει π.χ. μόνο pid, autotuning και pid ή μόνο autotuning.Υπάρχει όμως και η επιλογή του word address με την οποία μέσω ενός m word ορίζουμε εμείς την λειτουργία του. Βάση του help για το word address όταν το MW είναι ίσο με 1 έχουμε μόνο PID λειτουργία ενώ όταν είναι ίσο με 4 έχουμε μόνο PI λειτουργία. Αυτό του ορίζουμε και στο rung 3 με το operation control.Στη δεξιά μεριά του rung πολλαπλασιάζουμε με τους παράγοντες που μας δίνονται για κάθε παράμετρο. Για το Kp πολλαπλασιάζουμε με 100 αφού αυτός είναι ο παράγοντας μέτρησης βάση του προγράμματος και το μετατρέπουμε σε ακέραιο για να το στρογγυλοποιήσουμε βάζοντας το στο MW5.Το ίδιο κάνουμε και στις άλλες δύο παραμέτρους απλά εκεί πολλαπλασιάζουμε με το 10 αφού η μονάδα μέτρησης είναι τα 0,1 second

Το bit M60 ενεργοποιεί ένα button στην οθόνη με το οποίο μπορούμε να εναλλασσόμαστε μεταξύ heating και cooling λειτουργίας του PID.



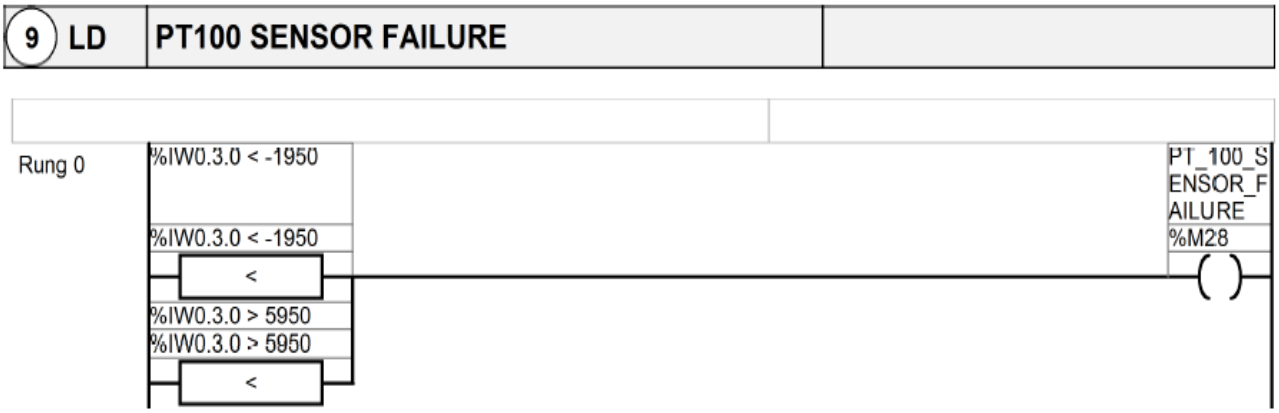
Σε αυτό το rung βλέπουμε πάλι στο αριστερό κομμάτι τη λειτουργία του PID μας .Όταν στο MW1 έχουμε το 2 βρισκόμαστε σε λειτουργία auto tuning και PID ενώ όταν έχουμε το 3 έχουμε μόνο auto tuning λειτουργία. Στη πρώτη περίπτωση το PLC υπολογίζει τις παραμέτρους του PID και τον βάζει σε λειτουργία ενώ στην δεύτερη περίπτωση υπολογίζει τις παραμέτρους αλλά εμείς έχουμε την τελική επιλογή για τις τελικές τιμές.

Τη δεξιά μεριά του rung κάνουμε ακριβώς την αντίστροφη διαδικασία απο αυτήν που κάναμε όταν ορίζαμε χειροκίνητα τις παραμέτρους του PID.Μετατρέπουμε τους ακέραιους που έχουμε στα MW5,MW6 και MW7 σε πραγματικούς και διαιρούμε με την εκάστοτε σταθερά βάση των χαρακτηριστικών του PID για να μπορέσουμε να έχουμε σωστή απεικόνιση των τιμών του autotuning στην οθόνη μας.



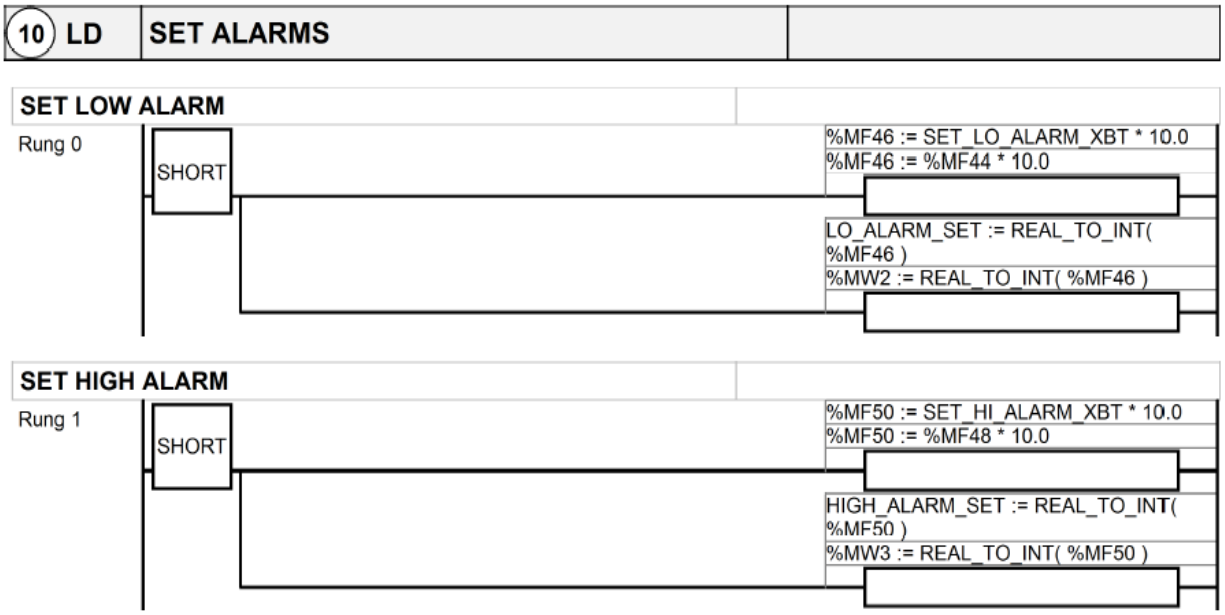
Σε αυτό το κομμάτι του προγράμματος ορίζουμε τη περίοδο δειγματοληψίας του PID μας, η οποία βρίσκετε στο block της οθόνης που ονομάζετε Ts. Το MF40 είναι η τιμή που βάζουμε στην οθόνη μας. Την πολλαπλασιάζουμε με 100, γιατί ο χρόνος που μετράει το PLC στο sampling period είναι 10ms (0,01s)και στη συνέχεια την μετατρέπουμε σε ακέραιο και την τοποθετούμε στο MW8 από το οποίο θα την διαβάσει το PLCμας.

Στο Rung1 οριοθετούμε το εύρος τιμών που θα ορίζουμε στο sampling period το οποίο πρέπει να είναι μεταξύ 0 και 100 (twido soft help).Έτσι εδώ κάνουμε τη σύγκριση και του λέμε οτι εάν είναι μικρότερο από 0 τότε κάντο ίσο με μηδέν και εαν είναι μεγαλύτερο του 100 τότε κάντο ίσο με 100.

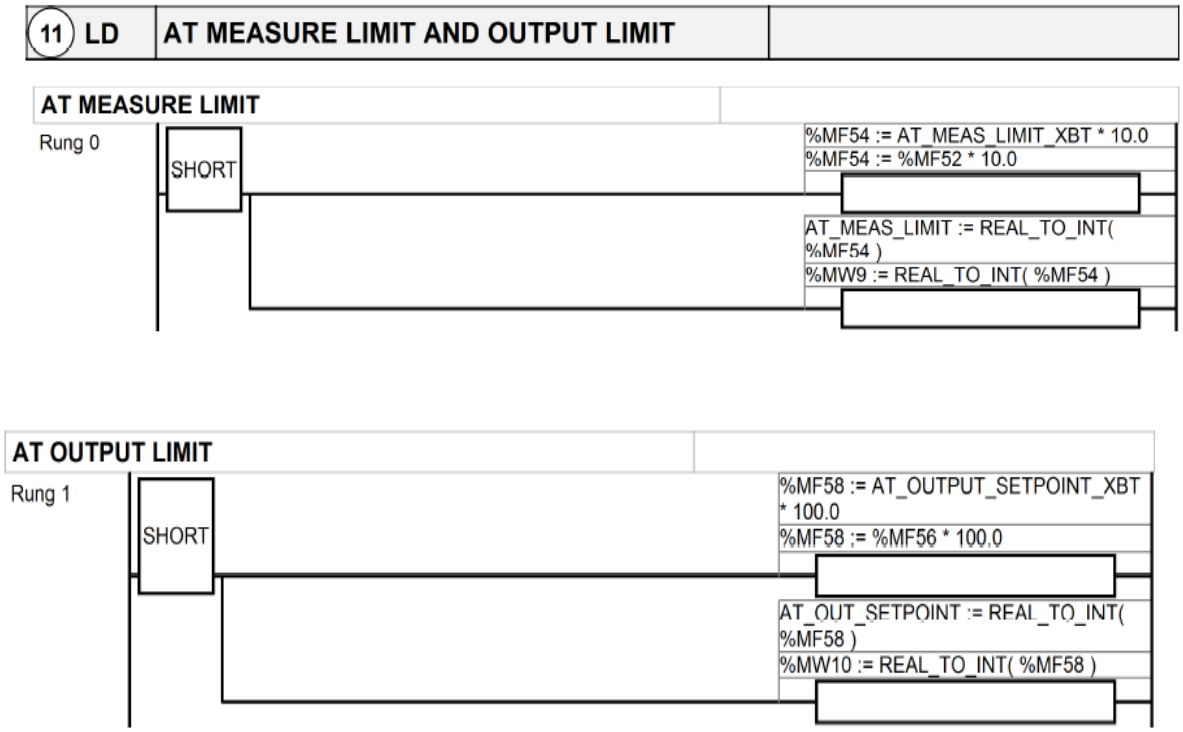


Το Pt100 που χρησιμοποιούμε για να μετρήσουμε την θερμοκρασία μας το συνδέουμε σε μια αναλογική είσοδο του PLC.Σε αυτή την είσοδο έχουμε ένα εύρος τιμών από -2000 έως 6000. Σε αυτό το κομμάτι του προγράμματος κάνουμε έναν έλεγχο στη περίπτωση που το PTμας δώσει ακραίες τιμές. Αυτό συμβαίνει συνήθως σε περίπτωση βραχυκυκλώματος .Έτσι συγκρίνοντας το input word που παίρνουμε τη μέτρηση του PT,

εάν είναι μικρότερο του 1950 ή μεγαλύτερο του 5950,ενεργοποιούμε ένα bit,το %M28,το οποίο θα μας δώσει ένα sensor failure στην οθόνη μας.



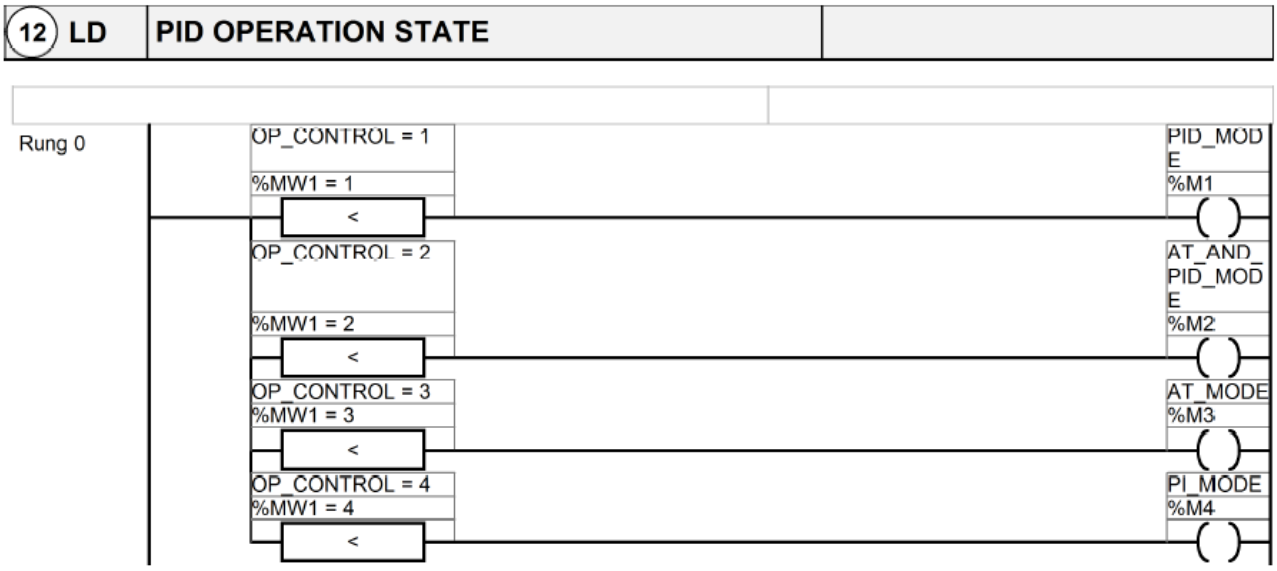
Σε αυτό το κομμάτι του προγράμματος ορίζουμε τα low και τα high alarm του process value. Πολλαπλασιάζουμε με το 10 όπως ακριβώς κάναμε και στο setpoint input γιατί τα units που μετράει το PLC είναι 0,1.Στη συνέχεια τα μετατρέπουμε σε ακεραίους και τα βάζουμε σε ένα m word για να το διαβάσει το PLC μας.



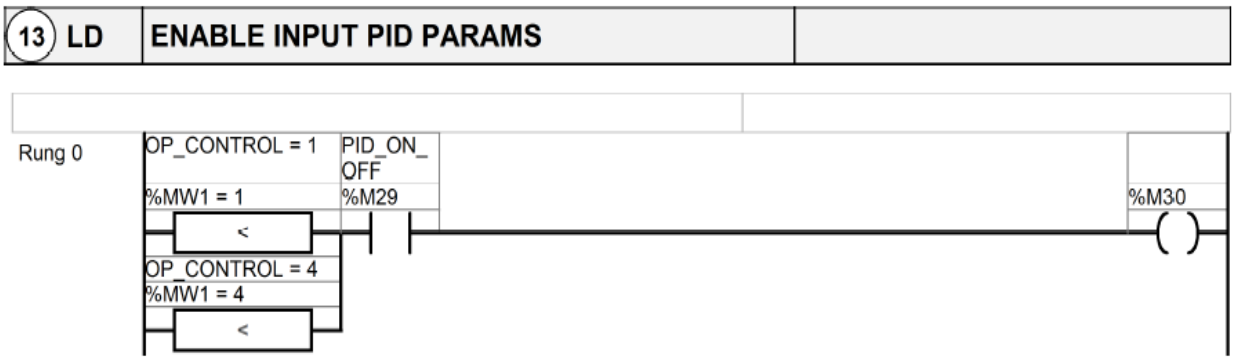
Εδώ ελέγχουμε δύο καταστάσεις. Στο rung 0 ελέγχουμε το όριο του process value μέσα στα οποία θα γίνετε το autotuning. Γίνετε με την ίδια διαδικασία με την οποία οποία ορίσαμε και το setpoint(πολλαπλασιάζουμε τη τιμή με 10 και την μετατρέπουμε σε ακέραιο).

Στο rung 1 ένα ελέγχουμε ποσοστιαία το όριο της εξόδου του PID μας, στη συγκεκριμένη περίπτωση μέχρι πόσο θα ανάψει η λάμπα με την οποία ελέγχουμε την θερμοκρασία. Η τιμή που διαβάζει το PLC μας για το AT OUTPUT LIMIT είναι μεταξύ 0 και 10000.

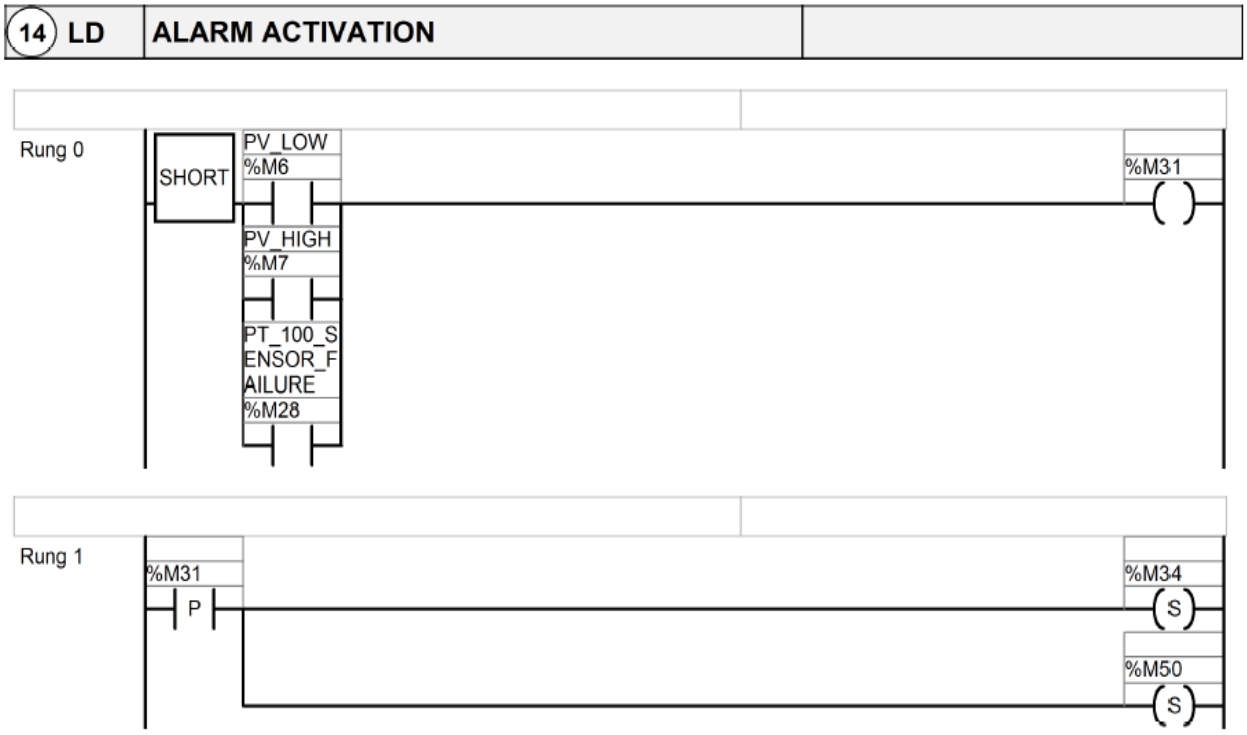
Και τα δύο rung αναφέρονται μόνο στην περίπτωση της autotuning λειτουργίας και στην οθόνη μας εμφανίζονται σαν popup window πατώντας το κουμπί της πρώτης σελίδας <<set at limits>>.



Εδώ στο RUNG 0 ελέγχουμε τα ενδεικτικά λαμπάκια της οθόνης μας. Όταν τοMW1 γίνει ίσο με 1 ενεργοποιούμε ένα Boolean το M1 το οποίο αντιστοιχεί στο λαμπάκι που είναι μόνο PID mode. Αντίστοιχα το M2 είναι για AT και PID mode το M3 μόνο για AT mode και το M4 μόνο για PI mode.



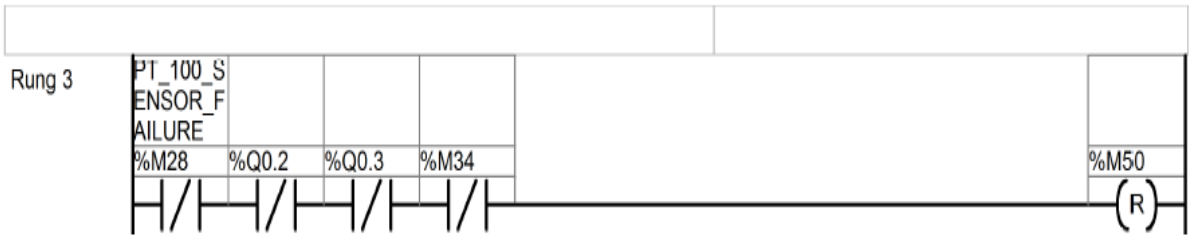
Σε αυτό το rung ελέγχουμε πότε θα μπορούμε να επεμβαίνουμε στις παραμέτρους του PID.Έτσι όταν είμαστε σε mode PID ή PI δλδ έχουμε στο MW1 1 ή 4 και είναι φυσικά και σε λειτουργία on τότε ενεργοποιούμε ένα Boolean το M30 το οποίο μας επιτρέπει να επέμβουμε στις παραμέτρους του PID.Αυτό γίνετε εάν πάμε στην εκάστοτε παράμετρο του PIDκάνοντας διπλο κλίκ πάνω στο κουτάκι της παραμέτρου και επιλέγοντας απο το εικονίδιο που θα μας εμφανιστεί advanced>enable interlock.



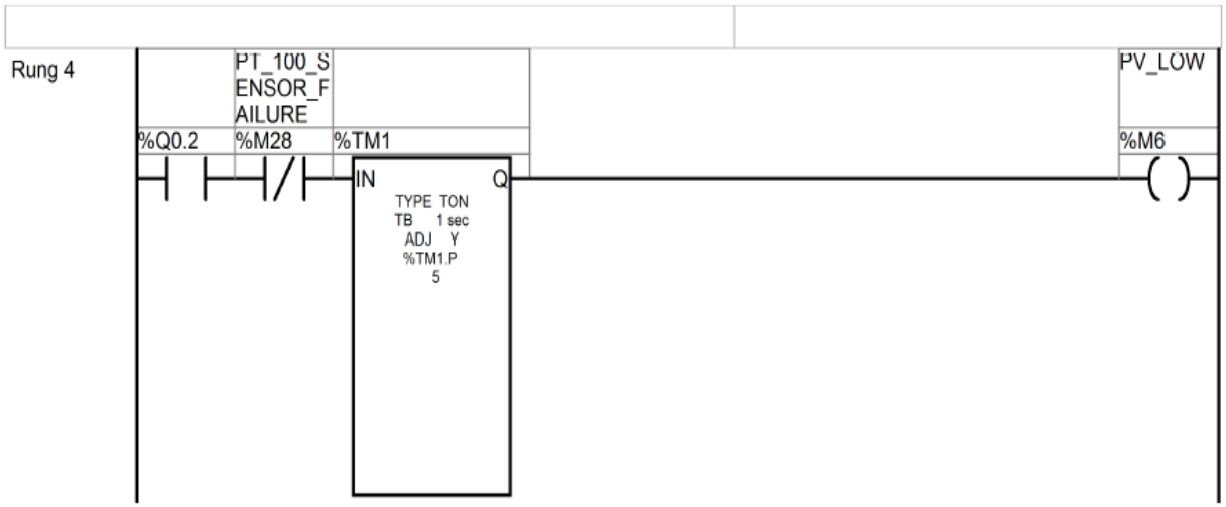
Εδώ στο RUNG 0 όταν ενεργοποιηθούν κάποιο από τα alarm PV LOW PV HIGH ή SENSOR FAILURE ενεργοποιούμε ένα το M31.

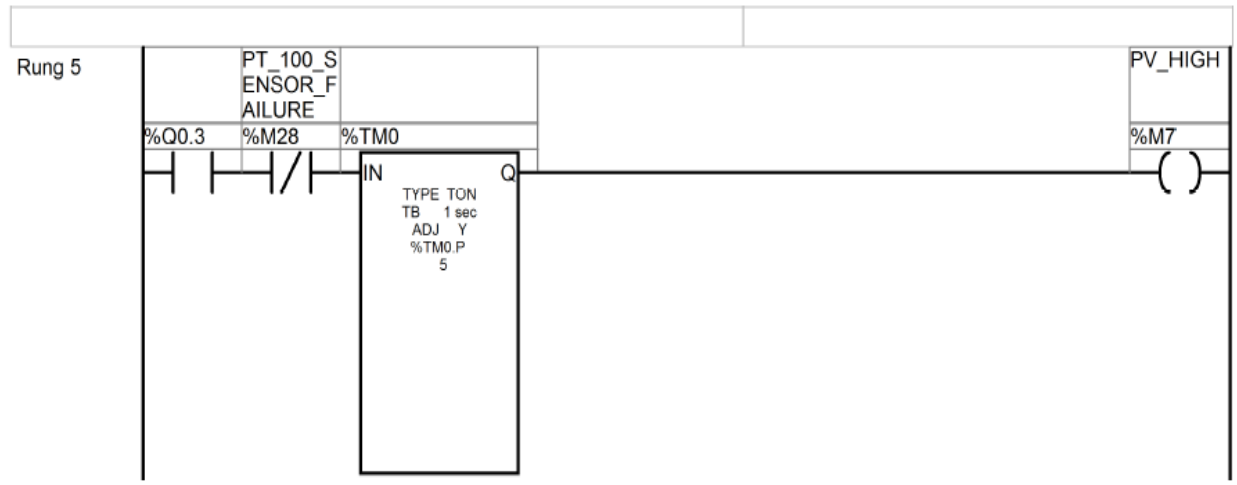
Στο RUNG1μόλις ενργοποιειθεί το M31 έστω και στιγμιαία(αυτό το δηλώνουμε με την εντολή P) τότε ενεργοποιούμε δυο bit το M34 το οποίο το χρησιμοποιούμε για το flick στο alarm και το M50 το οποίο μας κάνει το alarm κόκκινο.

Στο RUNG 2 χρησιμοποιούμε το M33 για να κάνουμε το acknowledge στο alarm μας. Μόλις πατηθεί δλδ το button ACN από την οθόνη μας το alarm που μας έχει έρθει σταματάει να αναβοσβήνει και παραμένει κόκκινο.

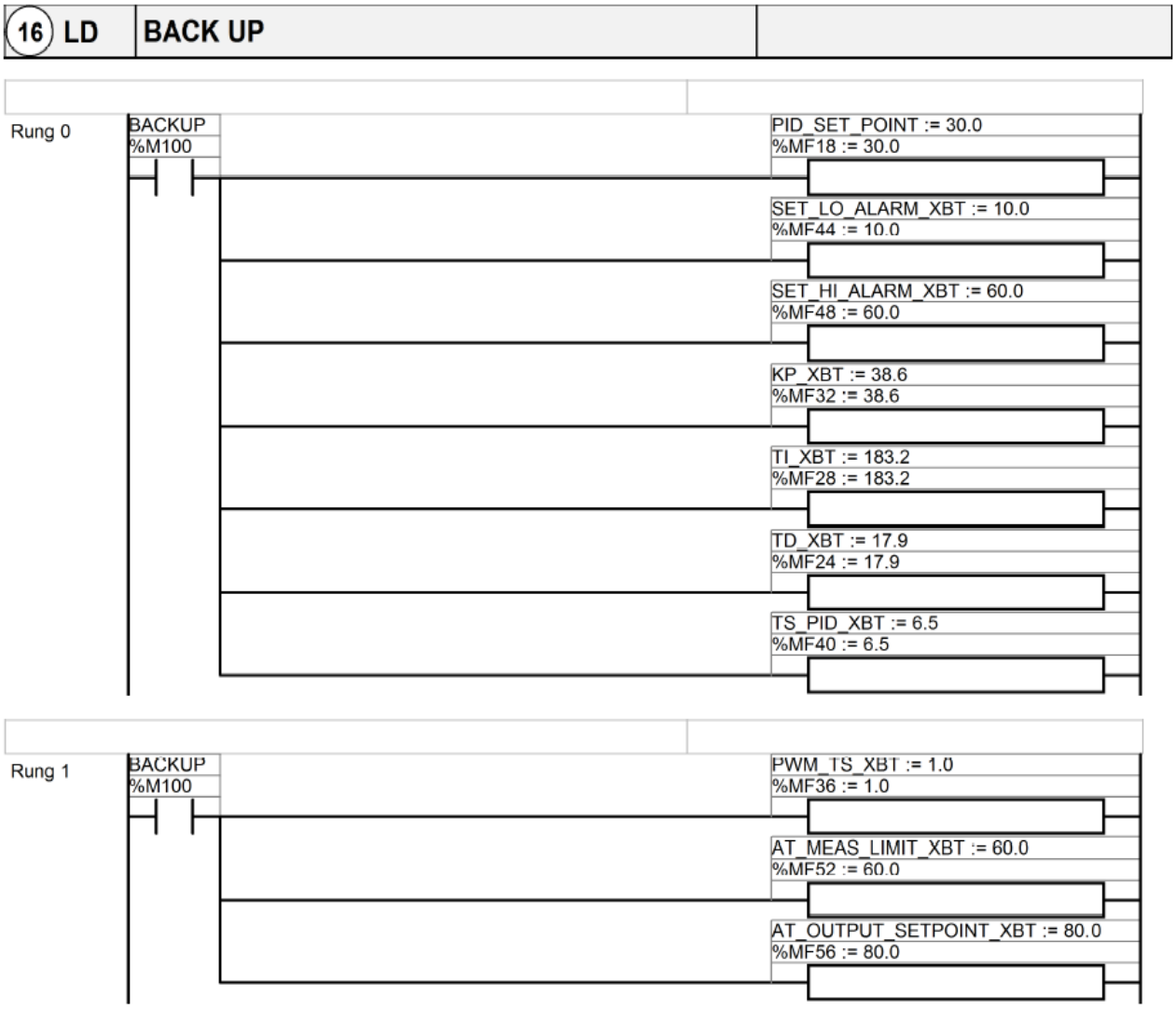


Στο rung 3 μόλις όλα τα alarm από το σύστημα μας απενεργοποιηθούν τότε φεύγει και το κόκκινο χρώμα από το alarm μας. Το M28 είναι για το sensor failure, τα Q0.2 και Q0.3 είναι τα ενδεικτικά λαμπάκια στο panel μας για τα pv high και pv low και το M34 είναι για το flicking.





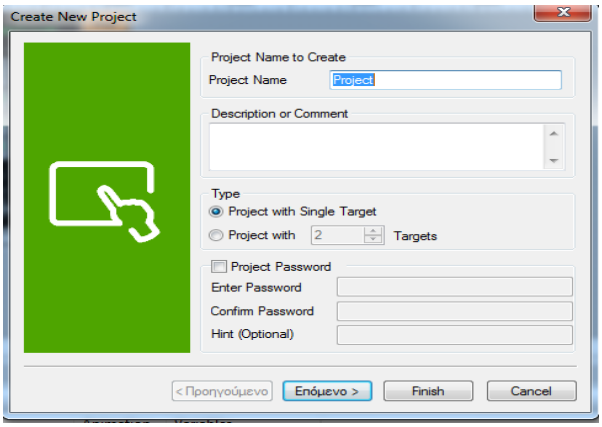
Στο rung 4 και 5 χρησιμοποιώντας δύο timerελέγχουμε τα alarmτων pv high και low έτσι ώστε να υπάρχει μια καθυστέρηση μέχρι να μας έρθουν τα alarmστην οθόνη μας. Αυτό γίνετε για να αποφύγουμε τα alarmπου θα μας έρθουν από πιθανόν στιγμιαίες τιμές.



Σε αυτό το κομμάτι ορίζουμε όλες τις τίμες του συστήματος μας με ιδανικές τιμές και τις συνδέουμε με ένα bit.Έτσι μόλις ενεργοποιηθεί το M100 ότι αλλαγές έχουν γίνει στις παραμέτρους μας επανέρχονται στις τιμές που του έχουμε ορίσει σαν ιδανικές.Το M100 το συνδέουμε στην οθόνη μας με το button backup.

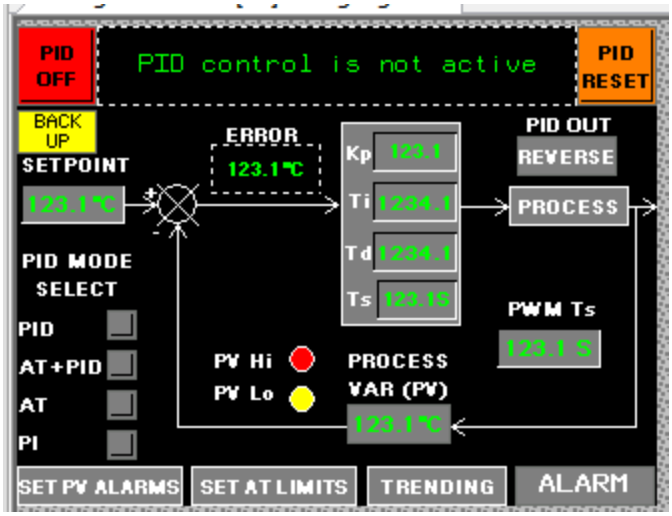
2.4 Ανάλυση software οθόνης

Το πρόγραμμα που θα χρησιμοποιήσουμε για να προγραμματίσουμε την οθόνη μας είναι το vijeo designer 6.2.Ανοίγωντας το προγραμμά μας θα επιλέξουμε file>new project και θα μας εμφανίσει την παρακάτω οθόνη:



Σε αυτή την οθόνη θα βάλουμε το όνομα της εφαρμογής μας,θα επιλέξουμε το project with single target επειδή θα είναι μια η οθόνη η οποία θα χειριζόμαστε και θα επιλέξουμε εάν η εφαρμογή μας θα προστατεύεται από κάποιο κωδικό.Πατώντας το κουμπί <επόμενο> θα ορίσουμε τον κωδικό της οθόνης που θέλουμε να χρησιμοποιήσουμε και μετά θα πατήσουμε finish για να ολοκληρώσουμε τη δημιουργία μας.

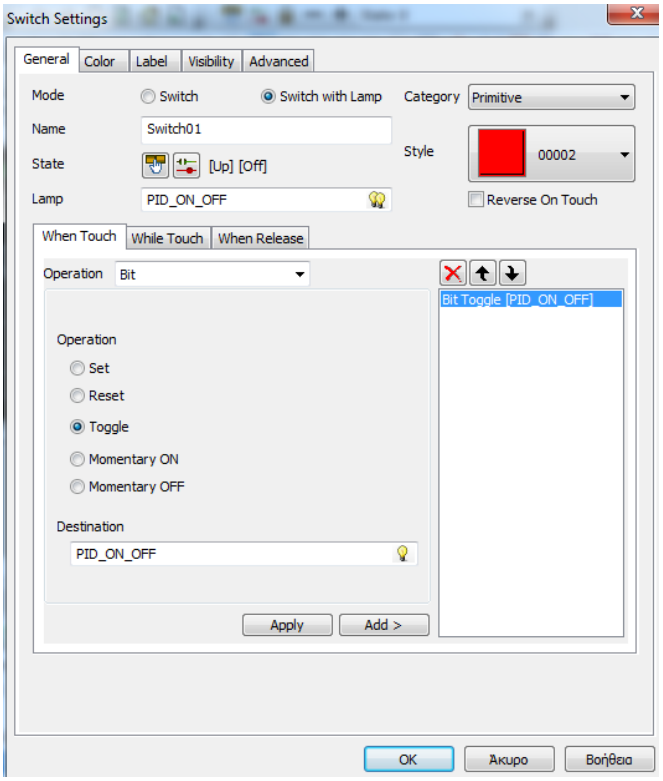
Η κεντρική οθόνη της εφαρμογής μας θα έχει την εξής μορφή:



Ας εξηγήσουμε αναλύτικα ένα ένα τα κομμάτια που συναντάμε στην οθόνη μας.

1)Switches

Ας ασχοληθούμε αρχικά με τους διακόπτες μας(switches).Οι διακόπτες που συναντάμε σε αυτή την οθόνη είναι οι εξής:pid off,pid reset,back up,pid,at+pid,at,pi,back up,set pv alarm,set at limit,trending,alarm.Για να σχεδιάσουμε έναν διακόπτη επιλέγουμε το εικονίδιο από τη μπάρα εργαλείων που γράφει switch.Αφού σχεδιάσουμε τον διακόπτη μόλις αφήσουμε το ποντίκι μας ανοίγει η ακόλουθη σελίδα:



General

Στην επιλογή general των ιδιοτήτων του διακόπτη μπορούμε να ορίσουμε τα εξής:

- Στο mode επιλέγουμε επιλέγουμε εάν ο διακόπτης μας θα είναι απλός η φωτεινός
- Στο state ορίζουμε την θέση του διακόπτη,τι θα κάνει όταν είναι δλδ πατημένος ή σε αδράνεια.
- Στο lamp ορίζουμε το bit με το οποίο θα ενεργοποιείται η λάμπα του διακόπτη,π.χ. στην περίπτωση του διακόπτη που ανάβουμε τον pid ορίζουμε το bit <M29 pid on off> που σημαίνει ότι μόλις πατηθεί ο διακόπτης και ενεργοποιηθεί το M29 τότε το χρώμα του θα γίνει πράσινο.Το χρώμα το ορίζουμε από το style.
- Στο when touch,while touch και when release ορίζουμε εάν η λειτουργία του θα γίνετε όταν πατηθεί,όσο πατηθεί ή όταν αφεθεί ο διακόπτης.
- Στο operation ορίζουμε τη λειτουργία του διακόπτη,π.χ. στην εικόνα που βλέπουμε η οποία αφορά τον διακόπτη pid on off η λειτουργία του διακόπτη θα είναι bit.Επιλέγοντας toggle αλλάζουμε μόνιμα την κατάσταση του.Το bit που θα αλλάξουμε την καταστασή του το ορίζουμε στο destination.

Ένας διακόπτης μπορεί να έχει πολλαπλές λειτουργίες οι οποίες εμφανίζονται στην λίστα δέξια και εκτελούνται σύμφωνα με τη σειρά με την οποία είναι τοποθετημένες στη λίστα.

Colour

Στην επιλογή colour ορίζουμε το χρώμα που θα έχει ο διακόπτης ανάλογα με την κατάσταση του π.χ όταν ο διακόπτης είναι off ο διακόπτης θα είναι κόκκινος ενώ όταν είναι on ο διακόπτης θα είναι πράσινος.

Label

Στο label ορίζουμε το label type του διακόπτη,εάν δηλαδή το μήνυμα του διακόπτη θα είναι στατικό ή θα αλλάζει ανάλογα με την κατάσταση του διακόπτη.Επίσης μπορούμε να ορίσουμε το μήνυμα που θα εμφανίζεται πάνω στο διακόπτη πάλι ανάλογα με την κατάσταση του.

Visibility

Στο visibility ορίζουμε με ποιά εντολή θα γίνετε ο διακόπτης μας ορατός ενεργοποιώντας το enable visibility.Επίσης μπορούμε να ορίσουμε με ποια εντολή θα αναβοσβήνει ο διακόπτης

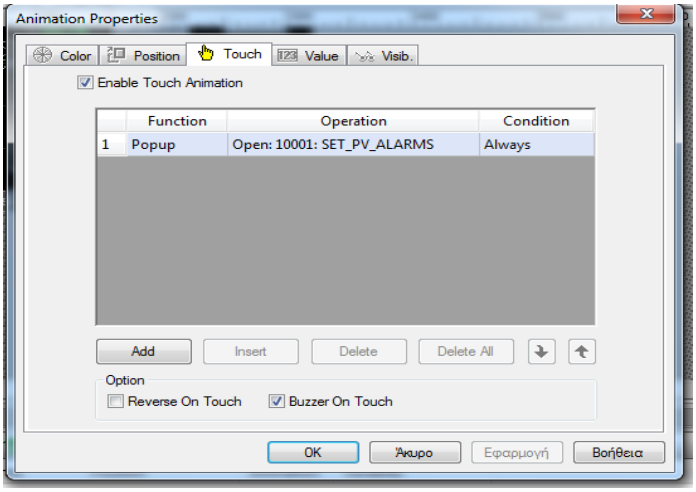
Advanced

Στη σελίδα του advanced μπορούμε επιλέγοντας το enable interlock να δηλωσουμε κάθε πότε θα ενεργοποιείται η λειτουργία του διακόπτη.Με το enable monitor μπορούμε να δηλώσουμε μια εντολή(π.χ. ένα bit) το οποίο να λετουργεί σαν επιβεβαίωση ότι ο διακόπτης έχει πατηθεί.Ενώ με το hold delay μπορείς να ορίσεις μια καθυστέρηση της λειτουργίας για όσο έχεις τον διακόπτη πατημένο.

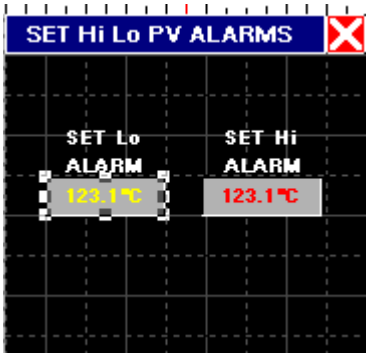
Set pv alarms - set at limits

Τα button που αναφέρονται από πάνω έχουν διαφορετική λειτουργία από τα υπόλοιπα.Δεν κάνουν απλά μια λειτουργία αλλά σε οδηγούν σε μία άλλη σελίδα (pop up window).Η διαδικασία δημιουργίας τους έχει ως εξής:

Αρχικά επιλέγοντας το κούμπι text από το toolbar δημιουργούμε ένα text πλαίσιο στην επιφάνεια εργασίας.Κάνοντας δεξί κλίκ πάνω στο πλαίσιο που φτιάξαμε επιλέγουμε το animation και θα μας εμφανιστεί η παρακάτω σελίδα:



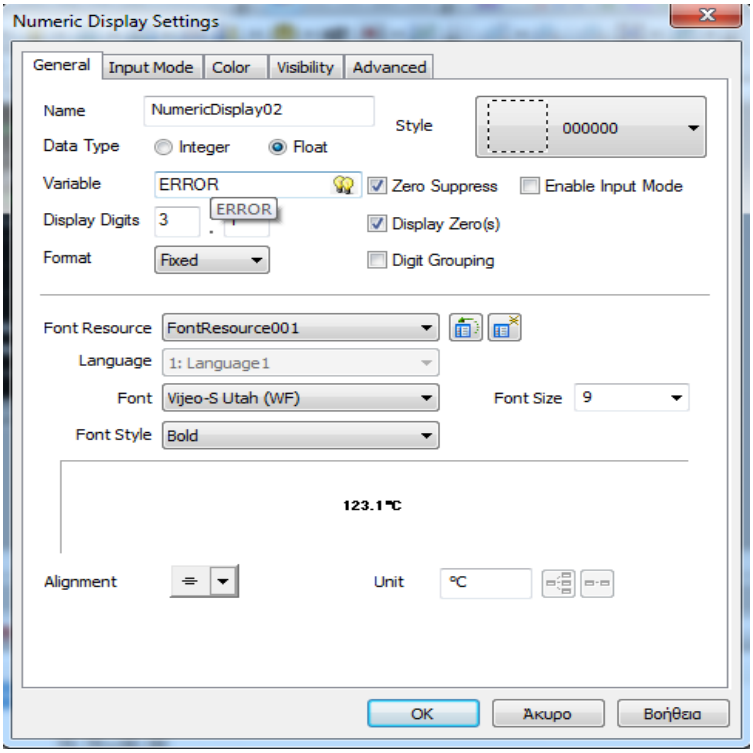
Επιλέγοντας το touch μπορούμε να ορίσουμε το pop up window που θα εμφανίζεται με το πάτημα του μπουτόν.Δηλώνοντας το παράθυρο που θα εμφανίζεται και πατώντας <ok> αυτόματα το παράθυρο θα εμφανιστεί και στη λίστα των pop up windows στο πλαίσιο navigator στο αριστερό μέρος της οθόνης.Στη συγκεκριμένη σελίδα ορίσαμε τη σελίδα των pv larm έτσι κάνοντας διπλό κλικ στο navigator στα pv alarm θα μας εμφανιστεί το παρακάτω:



Αντίστοιχα το ίδιο συμβαίνει και για την σελίδα των at limits.Και στα δύο pop up window χρησιμοποιούμε numeric για τις μεταβλητές μας τα οποία θα εξηγήσουμε αναλυτικότερα στην συνέχεια

2)Numeric display

Όπου εμφανίζονται μεταβλητές στην οθόνη μας τότε χρησιμοποιούμε numeric display.Επιλέγοντας το εικονίδιο που λέει numeric απο το toolbar και σχεδιάζοντας το στην επιφάνεια εργασίας μας εμφανίζεται η εξής σελίδα:



Στη συγκεκριμένη σελίδα μπορούμε να ορίσουμε τα εξης:

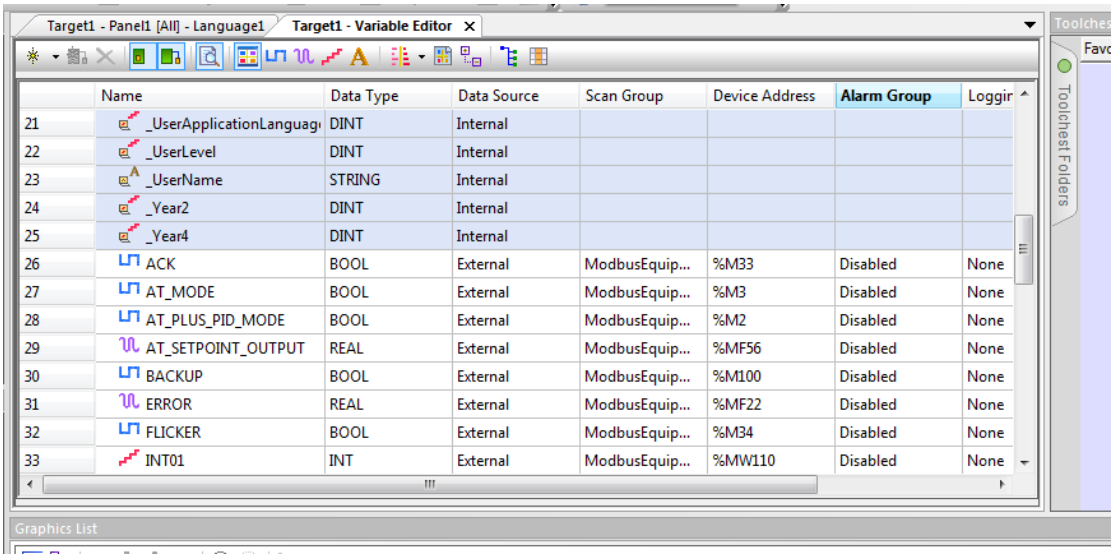
- Data type: ορίζουμε εάν η μεταβλητή μας θα εμφανίζετε σαν ακέραιος (integer) ή σαν δεκαδικός (float).
- Variable: του ορίζεις ποια μεταβλητή θα διαβάζει από το plc.Π.χ. στη συγκεκριμένη περίπτωση θα διάβάζει το error που αντιστοιχεί στο %MF22.
- Display units: καθορίζεις τον αριθμό των ψηφίων που θα εμφανίζονται.Το πρώτο κομμάτι αναφέρεται στα ακέραια και το δεύτερο στα δεκαδικά.

- Format: καθορίζεις τον τρόπο με τον οποίο θα εμφανίζονται τα ψηφία.Π.χ. μόνο ακέραια ή μόνο δεκαδικά.
- Zero supress: επιλέγοντας το μπορείς να σβήσεις τα επιπλέον μηδενικά στη μεταβλητή σου.
- Enable input mode: τα επιλέγεις για να έχεις τη δυνατότητα να επεμβαίνεις στη μεταβλητή σου.
- Advanced>enable interlock: το χρησιμοποιείς για να κλειδώσεις κάποια μεταβλητή σε περίπτωση που δεν θές για κάποιο λόγο να επέμβουν.Π.χ. το χρησιμοποιούμε όταν είμαστε σε mode auto tuning ή auto tuning και pid που δεν θέλουμε να έχουμε πρόσβαση στις μεταβλητες.

Όσο αναφορά τα colour,visibility και input mode έχουν να κάνουν με τον τρόπο που θα εμφανίζεται η μεταβλήτη μας (χρώμα ορατότητα κτλ.).

3)Variables

Δημιουργώντας ένα νέο project στην vijeo,το κομμάτι των μεταβλητών (variables) εμφανίζεται αυτόματα στον navigator της αρχικής οθόνης. Κάνοντας διπλό πάνω στα variables θα δούμε την εξής σελίδα:



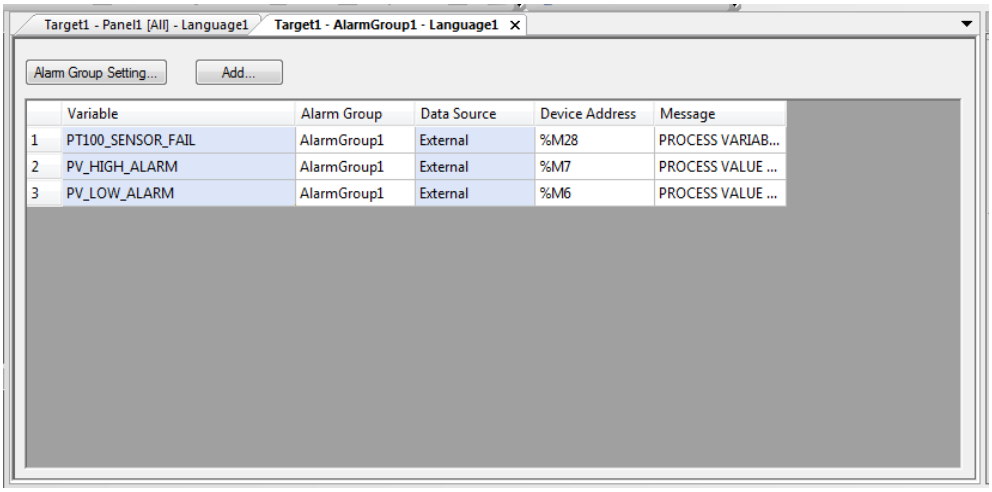
	Name	Data Type	Data Source	Scan Group	Device Address	Alarm Group	Loggir
21	_UserApplicationLanguag	DINT	Internal				
22	_UserLevel	DINT	Internal				
23	_UserName	STRING	Internal				
24	_Year2	DINT	Internal				
25	_Year4	DINT	Internal				
26	ACK	BOOL	External	ModbusEquip...	%M33	Disabled	None
27	AT_MODE	BOOL	External	ModbusEquip...	%M3	Disabled	None
28	AT_PLUS_PID_MODE	BOOL	External	ModbusEquip...	%M2	Disabled	None
29	AT_SETPOINT_OUTPUT	REAL	External	ModbusEquip...	%MF56	Disabled	None
30	BACKUP	BOOL	External	ModbusEquip...	%M100	Disabled	None
31	ERROR	REAL	External	ModbusEquip...	%MF22	Disabled	None
32	FLICKER	BOOL	External	ModbusEquip...	%M34	Disabled	None
33	INT01	INT	External	ModbusEquip...	%MW110	Disabled	None

Σε αυτή τη λίστα μέχρι το 25 έχουμε στοιχεία του συστήματος και από κει και κάτω πρέπει να καταχωρήσουμε τις μεταβλητές που χρησιμοποιούμε σύμφωνα με τον τρόπο υπάρχουν αυτές στο πρόγραμμα μας. Για την καταχώρηση της μεταβλητής μας χρειάζονται¨

- Name:το όνομα της μεταβλητής μας
- Data type:ο τύπος της μεταβλητής π.χ. εάν είναι real ή bool
- Data source: η προέλευση της (στο σύστημα μας όλες οι μεταβλητές μας είναι external αφού προέρχονται όλες από το twido).
- Device address: η διεύθυνση της μέσα στο πρόγραμμα, π.χ. εάν είναι bool θα έχει το M33.

4)Alarms

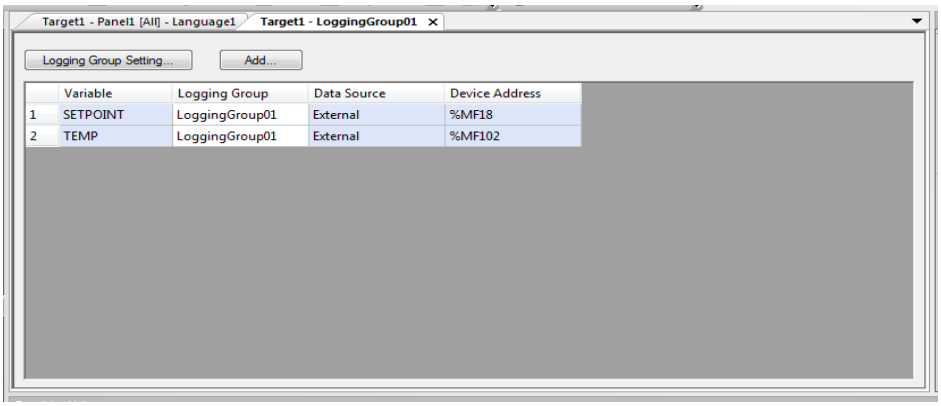
Τα alarm είναι ένα και αυτή μια κατηγορία την οποία μας την εμφανίζει αυτόματα η vijeo με την δημιουργία ενός νέου project.Στη συγκεκριμένη εφαρμογή έχουμε ένα alarm group, κάνοντας διπλό κλίκ επάνω θα δούμε την εξής σελίδα:



Σε αυτή τη σελίδα ορίζουμε τα alarm της εφαρμογής, στη προκειμένη περίπτωση αυτά είναι τρία: τα high και low της process value και το sensor fail του αισθητήρα. Πάλι εδώ χρειάζεται να ορίσουμε την ακριβή διεύθυνση που έχει το alarm στο πρόγραμμα μας.

5)Data logging

Το data logging έχει να κάνει με την καταγραφή των δεδομένων. Και αυτό το κομμάτι εμφανίζεται αυτόματα μετά τη δημιουργία του νέου project.Κάνοντας διπλό κλικ στο data logging θα δούμε την εξής σελίδα:



Δύο είναι οι μεταβλητές στην εφαρμογή μας οι οποίες θα καταγράφονται, οι μεταβολές του setpoint και της θερμοκρασίας. Επιλέγοντας target>options>data locations ορίζουμε που θα γίνετε η καταγραφή του log. Έχουμε τρείς επίλογές:

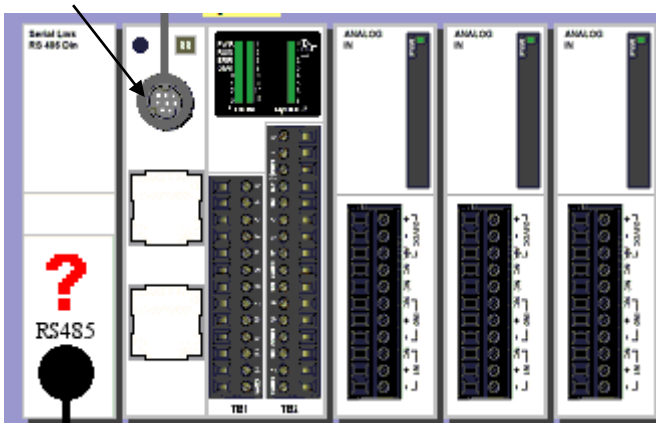
- Main drive: αποθήκευση στην κύρια μνήμη της οθόνης.
- Optional drive: αποθήκευση σε USB.
- Secondary drive: αποθήκευση σε sd card.

Επίσης μέσα από το logging group settings μπορούμε να ορίσουμε κάθε πότε θα γίνετε η δειγματοληψία των μεταβλητών μας (periodic ή trigger) αλλά και την απόκλιση των καταγραφών μεταξύ τους (deviation).Μέσα από το variable storage ορίζουμε τον τρόπο αποθήκευσης π.χ. μπορούμε να επιλέξουμε να αποθηκεύεται και στην sram που είναι η εσωτερική μνήμη της οθόνης και σε file δλδ σε ένα ξεχωριστό αρχειο καθώς επίσης μπορούμε να ορίσουμε και τον ρυθμό ανανέωσης των δεδομένων.

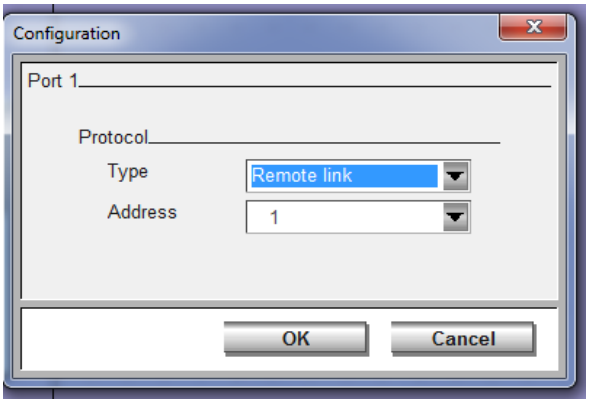
6)I/O Manager

Στο I/O manager ορίζουμε τον τρόπο που θα επικοινωνεί η οθόνη μας με το plc μας. Το πρωτόκολλο επικοινωνίας μας είναι modbus.Μπορούμε να έχουμε modbus RTU και modbusTCP. Όταν έχουμε RTU σημαίνει ένα master station και πολλά slave station ενώ όταν έχουμε TCP σημαίνει ότι έχουμε πολλά master station και πολλά slave station.

Master station είναι ο σταθμός που ρωτάει ενώ slave αυτός που κάνει την ερώτηση. Στη συγκεκριμένη εφαρμογή master είναι η οθόνη μας ενώ slave το plc.Για να ρυθμίσουμε την επικοινωνία στο plc μας πρέπει να πάμε στο twido στη σελίδα που φαίνεται παρακάτω:



Κάνοντας διπλό κλικ εκεί που μας δείχνει το βέλος θα μας ανοίξει το configuration για την επικοινωνία μας:



Στο type θα επιλέξουμε remote link επειδή ο χειρισμός θα γίνετε από την οθόνη μας και στο address θα επιλέξουμε το 1 επειδή είναι ο πρώτος σταθμός slave αφού το μηδέν είναι το master.

Αντίστοιχα και στην οθόνη μας στο equipment configuration πρέπει να δηλώσουμε τον αριθμό του slave

Κεφάλαιο 3

3.1 Ανάλυση της λειτουργίας auto tuning μέσω του plc.

Η λειτουργία του autotuning που υπάρχει στο twido plc είναι ειδικά σχεδιασμένη αυτόματη παραμετροποίηση σε θερμικές διαδικασίες.Καθώς οι τιμές των παραμέτρων του pid ποικίλουν από την μια διαδικασία ελεγχου στην άλλη, το auto tuning του twido μας βοηθάει να καθορίσουμε με ακρίβεια τιμές με μεγάλυτερη ευκολία.

"Όταν χρησιμοποιούμε το auto tuning πρέπει να είμαστε σίγουροι ότι το plc μας και η διαδικασία ελέγχου πληρούν τις εξής προυποθέσεις:

1. Η διαδικασία ελέγχου πρέπει να είναι ένα σταθερό σύστημα ανοιχτού βρόγχου.
2. Όταν ξεκινάμε το auto tuning πρέπει το συστημά μας να βρίσκετε σε μια σταθερή κατάσταση με μηδενική τιμή στην εισοδό του.(π.χ ένα καζάνι νερού θα πρέπει να βρίσκετε σε θερμοκρασία περιβάλλοντος).
3. Κατά τη διάρκεια του auto tuning πρέπει να είμαστε σίγουροι ότι το σύστημα μας δεν θα δέχεται εξωτερικές παρεμβολές γιατί είτε θα υπολογιστούν λάθος οι παράμετροι ή η διαδικασία του auto tuning δεν θα λειτουργεί σωστά (π.χ. εάν σε ένα καζάνι νερού ανοίξει ακόμα και στιγμιαία το καπάκι του καζανιού τότε μεγάλη παρέκκλιση στο auto tuning).

4. Πρέπει να ρυθμιστεί το plc έτσι ώστε να σκανάρει σε μια συγκεκριμένη περίοδο.Εφόσον ρυθμίσουμε την σωστή περίοδο δειγματοληψίας για το auto tuning, η περίοδος σκαναρίσματος πρέπει να ρυθμιστεί έτσι ώστε η περίοδος δειγματοληψίας να είναι ακριβές πολλαπλάσιο της περιόδου του plc.

Για να διασφαλίσουμε την σωστή λειτουργία του pid και κατά συνέπεια του auto tuning είναι πολύ σημαντικό να ρυθμίσουμε το plc να κάνει περιοδικό σκανάρισμα και όχι κυκλικό.Με αυτό τον τρόπο η δειγματοληψία γίνετε ανα τακτά διαστήματα σε αντίθεση με το κυκλικό σκανάρισμα στο οποίο το ένα στάδιο ξεκινάει μόλις τελειώνει το προηγούμενο και μπορεί να προκαλέσει εσφαλμένα αποτελέσματα.

Το auto tuning μπορεί να χρησιμοποιηθεί είτε ανεξάρτητα (AT mode) είτε σε συνδυασμό με τον PID(AT + PID).

- AT MODE: Μετά την ολοκλήρωση της διαδικασίας του auto tuning και τον επιτυχή καθορισμό των παραμέτρων του PID οι αριθμητικές έξοδοι του auto tuning μηδενίζονται και στη λίστα με την κατάσταση του PID εμφανίζεται το μήνυμα <auto tuning complete>.
- AT + PID: Αφού ξεκινήσει και ολοκληρωθεί η διαδικασία του AT τότε αυτόμάτως ξεκινά ο έλεγχος του PID με τις παραμέτρους που έχουν οριστεί από το auto tuning. Εάν σε αυτή την επιλογή εντοπιστεί λάθος από τον αλγόρυθμο του AT τότε δεν θα υπολογιστεί καμία παράμετρος για τον PID, οι μεταβλητές θα επανέλθουν στιις τιμές που είχαν πριν ξεκινήσει το AT,θα εμφανιστεί μήνυμα <drop down list> και ο έλεγχος του PID θα ακυρωθεί.

Μέθοδος υπολογισμού του χρόνου δειγματοληψίας.

Μέθοδος απόκρισης καμπύλης

Σε αυτή τη μέθοδο θέτουμε μια βηματική αλλαγή στην είσοδο του συστήματος μας και καταγράφουμε την εξοδό μας συναρτήσει του χρόνου.Η μέθοδος αυτή μπορεί να περιγραφεί από την εξης σχέση:

$$\frac{S}{U}=\frac{k}{1+tp}*e^{-\theta p}$$

Τα βήματα για να καθορίσουμε την περίοδο δειγματοληψίας με αυτή την μέθοδο είναι τα εξής:

1. Ανοίγουμε τον PID μας στο TWIDO SOFT και επιλέγουμε το output tab.

2. Επιλέγουμε το manual mode και στη συνέχεια το authorize ή address bit.Εφόσον τώρα έχουμε τον PID μας σε manual mode θέτουμε την εξοδό μας σε υψηλά επίπεδα (εύρος 5000-10000).
3. Επιλέγουμε PLC>Transfer PC to PLC για να μεταφέρουμε την εφαρμογή μας στο PLC.
4. Μέσα από το configuration του PID μας επιλέγουμε το trace mode, βάζουμε τον PID σε λειτουργία και ελέγχουμε την απόκριση της καμπύλης μας.
5. Όταν η καμπύλη μας φτάσει σε μία σταθερή κατάσταση σταματάμε την καταγραφή του PID.
6. Για να υπολογίσουμε την σταθερά του χρόνου για την διαδικασία ελεγχου βάση της γραφικής μεθόδου πρέπει να υπολογίσουμε την μεταβλητή εξόδου σε αύξηση 63% (S63%),χρησιμοποιώντας τον τύπο $S_{63\%=S_{initial}}$+(S_{ending}+$S_{initial}$)*63%. Στη συνέχεια υπολογίζουμε γραφικά το $t_{63\%}$ και το $t_{initial}$ και από τον τύπο t=t_{63}-$t_{initial}$ βρίσκουμε την σταθερά του χρόνου.
7. Ο υπολογίσμός του χρόνου δειγματοληψίας θα είναι:

   Ts=t/75
8. Επιλέγουμε Program>Scan the behaviour. Θέτουμε το scan mode ως periodic. Θέτουμε το scan period ως ακριβές πολλαπλάσιο του Ts (sampling period).

3.2 Μέθοδος βαθμονόμησης PID.

Υπάρχουν πολλές μέθοδοι υπολογισμού των παραμέτρων του PID.Εμείς θα χρησιμοποιήσουμε την μέθοδο ziegler-nichols η οποία έχει δυο παραλλαγές:

- Ανοιχτού βρόγχου
- Κλειστού βρόγχου

Πρίν εφαρμόσουμε την μέθοδο θα πρέπει να ορίσουμε την κατεύθυνση δράσης του PID.Εάν η αύξηση στην εξοδό μας προκαλεί αύξηση στο process value τότε θέτουμε τον PID ως inverted (kp>0). Αντίστοιχα εάν προκαλούμε μείωση στο process value τότε θετουμέ τον pid ως direct (kp<0).

Βαθμονόμηση κλειστού βρόγχου.

Σε αυτή την περίπτωση δίνουμε μια τιμή στην kp διατηρώντας το Ti και Td μηδενικά.Αυξομειώνουμαι την kp μέχρι να σχηματίσουμε μια ομαλή ταλάντωση γύρω από το setpoint. Η γραφική παράσταση που θα δημιουργηθεί θα μας δώσει το kpc που προκαλεί αυτήν την ταλάντωση αλλά και την περίοδο της ταλάντωσης.

Αναλόγως το είδος του ελέγχου που κάνουμε (PID ή PI) μπορούμε να υπολογίσουμε τις μεταβλητές μας από τον εξής πίνακα:

-	Kp	Ti	Td
PID	Kpc/1,7	Tc/2	Tc/8
PI	Kpc/2,22	0,83*Tc	-

Ziegler nichols :jens graph

Αυτή η μέθοδος μπορεί να μας οδηγήσει σε μεγάλη υπερπήδηση από την επιθυμητή μας τιμή.Σε αυτή την περίπτωση ρυθμίζουμε τις τιμές μας μέχρι να πάρουμε την επιθυμητή συμπεριφορά.

Βαθμονόμηση ανοιχτού βρόγχου

Καθώς έχουμε την ρυθμισή μας σε manual mode ορίζουμε μια τιμή στην έξοδο έτσι ώστε η διαδικασία εκκίνησης να είναι η ίδια όπως ενός ολοκληρωτή με καθυστέρηση χρόνου.

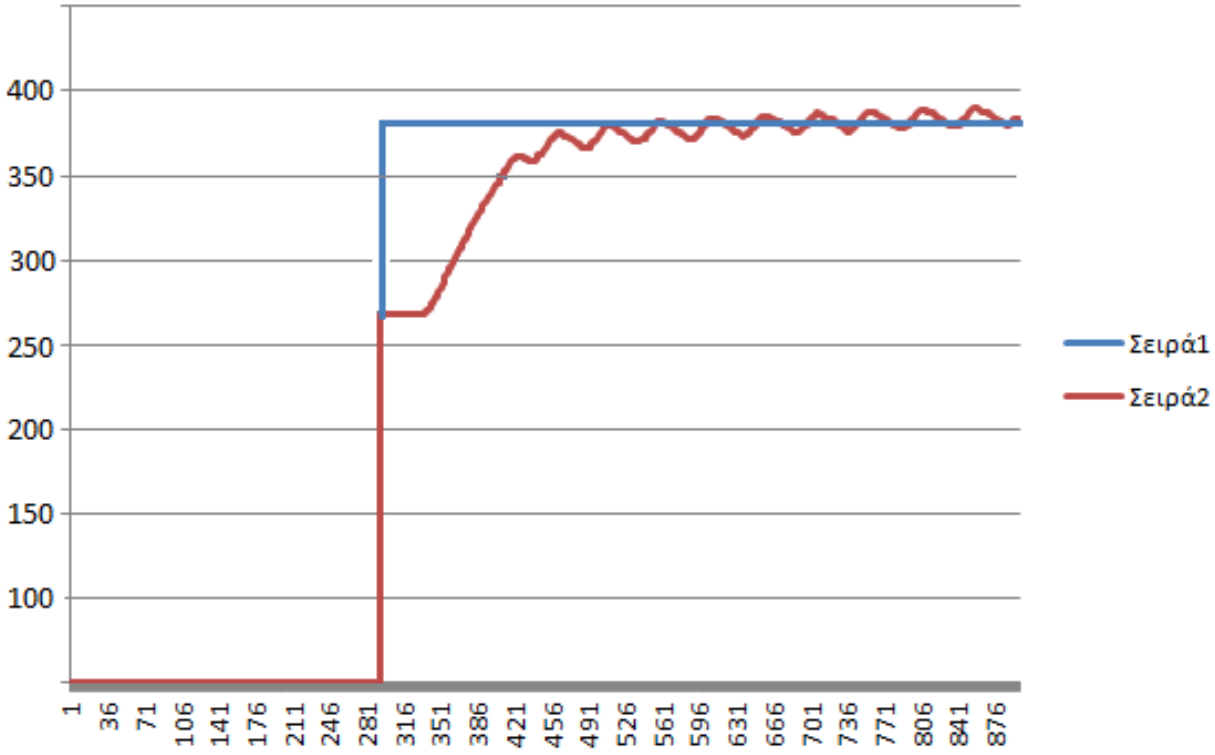
Για τον υπολογισμό των παραμέτρων από την παραπάνω γραφική εφαρμόζουμε τον εξής πίνακα:

-	Kp	Ti	Td
PID	-1,2Tg/Tu	2*Tu	0,5*Tu
PI	-0,9Tg/Tu	3,3*Tu	-

Ziegler – nichols : jens graph

3.3 Υπολογισμός παραμέτρων PID

Στην συγκεκριμένη εφαρμογή η γραφική παράσταση που προκύπτει από την μέθοδο του κλειστού βρόγχου είναι η εξής:



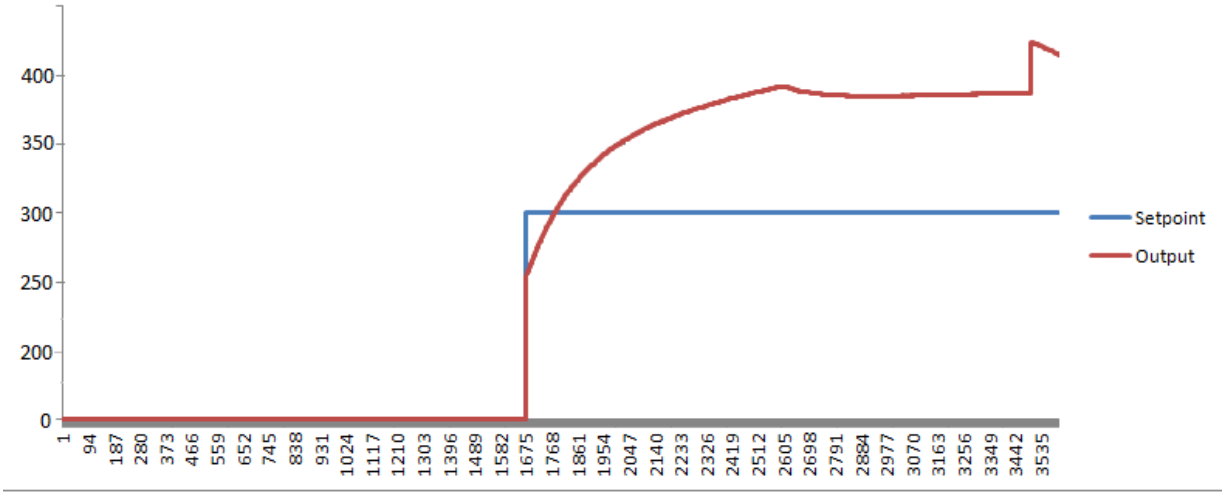
Εικ.9 Extract graph from twido soft

Το Kpc και το TC που προκύπτουν για να φτάσουμε σε αυτό το αποτέλεσμα είναι Kpc=65,62 και Tc=366,4.Ακολουθώντας τον πίνακα για την βαθμονόμηση κλειστού βρόγχου έχουμε τα εξής αποτελέσματα:

Kp=38,6 Ti=183,2 Td=45,8

3.4 Υπολογισμός περιόδου δειγματοληψίας

Όσο αφορά τον υπολογισμό της περιόδου δειγματοληψίας βάζουμε τον ελεγκτή μας σε λειτουργία auto tuning και θέτουμε μια πολύ μεγάλη τιμή στην αναλογική μας έξοδο.Η γραφική παράσταση που θα πάρουμε είναι η εξής:



Εικ. 10 Extract graph from twido soft

Η τιμή που παίρνουμε σαν S initial είναι S(i)=265 ενώ αυτή που παίρνουμε σαν S ending είναι S(e)= 395. Ο υπολογισμός για την περίοδο δειγματοληψίας βάσει του τύπου που μας δίνει το help για το auto tuning είναι:

$$S_{63\%=S_{initial}}+(S_{ending}+S_{initial})*63\%.$$

Άρα

$$S_{63\%}=265 + (395-265) * 63\%=341,9$$

Για την σταθερά του χρόνου έχουμε:

$$t=t_{63}-t_{initial}$$

Άρα $t=473$ sec και $Ts=t/75=6,3$sec

Περιεχόμενα

Βιβλιογραφία

Twido soft general help

Process value : Bela . Liptak

Ziegler-Nichols : Jens Graf

Αναστασία Βελώνη : Συστήματα Αυτομάτου Ελέγχου

Χρήστος Παπαζαχαρίας : Λύσεις στον προγραμματισμο και στην εγκατάσταση PLC.

Χρήστος Παπαζαχαρίας : Βιομηχανικά δίκτυα και ξελιγμένος προγραμματισμός PLC.

Παντελής Μαλατέστας : Συστήματα αυτομάτου ελέγχου

Παντελής Μαλατέστας : Ασκήσεις συστημάτων αυτομάτου ελέγχου.

Σταύρος Ρουμπής : Αυτοματισμός με προγραμματιζόμενους ελεγκτές

Ιωάννης Μπερέτας : Αυτοματισμός με χρήση PLC.

Library.automationdirect.com