



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ
ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

**Κατασκευή Συστήματος Ελέγχου Στάθμης Δεξαμενών
Ελεγχόμενο από PLC**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Διονύσης Δ. Μαρίνος

Επιβλέπων : Μαρία Γ. Ιωαννίδου
Καθηγήτρια Ε.Μ.Π.

Αθήνα, Ιούλιος 2011



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ
ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

**Κατασκευή Συστήματος Ελέγχου Στάθμης Δεξαμενών
Ελεγχόμενο από PLC**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Διονύσης Δ. Μαρίνος

Επιβλέπων : Μαρία Γ. Ιωαννίδου
Καθηγήτρια Ε.Μ.Π.

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

.....
Μ. Γ. Ιωαννίδου
Καθηγήτρια Ε.Μ.Π.

.....
Ν. Θεωδόρου
Καθηγητής Ε.Μ.Π.

.....
Π. Τσαραμπάρης
Λέκτορας Ε.Μ.Π.

Αθήνα, Ιούλιος 2011

.....

Διονύσης Δ. Μαρίνος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Διονύσης Δ. Μαρίνος, 2011

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ' ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθούν ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

ΠΕΡΙΛΗΨΗ

Στην κατασκευή αυτή το πρόβλημα που καλούμαστε να επιλύσουμε αφορά τον έλεγχο στάθμης δυο δεξαμενών που περιέχουν νερό. Δηλαδή έχουμε τοποθετήσει δυο δεξαμενές σε διαφορετικό ύψος και πρέπει το νερό από την κάτω δεξαμενή να μετακινείται μέσω μιας αντλίας στην πάνω δεξαμενή. Κατόπιν το νερό ελευθερώνεται μέσω μιας ηλεκτροβαλβίδας και επιστρέφει στην κάτω δεξαμενή και ο κύκλος επαναλαμβάνεται μέχρι να το διακόψει ο χρήστης. Επιπλέον αναφέρεται ότι κατά την ενεργοποίηση του προγράμματος έχει πάντα προτεραιότητα η αντλία εκτός από την περίπτωση που η πάνω δεξαμενή είναι γεμάτη. Ο έλεγχος στάθμης υλοποιείται με αισθητήρια τα οποία συνδέονται σε έναν επιτηρητή στάθμης, οι έξοδοι του οποίου αποτελούν την είσοδο του PLC LOGO! SIEMENS και πιο συγκεκριμένα τύπου LOGO!Basic 230RC. Για τις ανάγκες τροφοδοσίας χρησιμοποιήθηκε ένας μετασχηματιστής 230 V AC/12 V DC με μέγιστο ρεύμα 4 A. Το κάθε εξάρτημα της κατασκευής εξετάζεται σε ξεχωριστή παράγραφο ενώ ακολουθεί αναλυτική περιγραφή του κύκλου εργασίας και των σημάτων που χρησιμοποιούνται σ' αυτόν.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ

Προγραμματιζόμενοι λογικοί ελεγκτές, PLC SIEMENS, γλώσσες προγραμματισμού PLC, LOGO! 230RC, επιτηρητής στάθμης, αισθητήρια στάθμης, αντλία, ηλεκτροβαλβίδα.

ABSTRACT

In this experiment the required issue is required is to discuss the control of the level of the water that is contained in two small tanks. We have two tanks in different heights and the water should move from the lower tank to the higher tank, with the aid of a small pump. The flow of water from the higher tank to the lower tank is controlled by a solenoid valve. The cycle is repeated again and again until the operator, who controls the system stops the power supply. In addition, as the program starts to run the pump has priority against the solenoid valve except when the upper tank is thought to be full of water (from the user and the PLC). The level of water is controlled by sensors, connected to a liquid level monitor, the output of which are inputs for PLC Logo! SIEMENS (PLC type: LOGO!Basic 230RC). A voltage adaptor should be used with characteristics 230 V AC / 12V DC, $I_{\max} = 4$ A. Every component of this construction is discussed in a separate paragraph, along with analytical description of the work / procedure cycle and the signals that are used in each cycle.

KEYWORDS

Programmable Logic Controllers, PLC SIEMENS, programming languages of PLC, LOGO! 230 RC, liquid level supervisor, level sensors, electric pump, solenoid valve.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να εκφράσω την ειλικρινή ευγνωμοσύνη μου στην επιβλέπων καθηγήτρια μου Μαρία Γ. Ιωαννίδου, Καθηγήτρια της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών Ε.Μ.Π., για την εμπιστοσύνη που μου έδειξε με την ανάθεση αυτής της διπλωματικής εργασίας, καθώς και για την προσπάθεια που καταβάλει να μεταδώσει τις γνώσεις και τη νοοτροπία του Μηχανικού προς τους φοιτητές της.

Τέλος, οφείλω να ευχαριστήσω την οικογένεια μου που με στήριξε και με στηρίζει όλα αυτά τα χρόνια και που χωρίς την πολύτιμη παρουσία της δε θα είχα καταφέρει τίποτα από όσα έχω μέχρι στιγμής επιτύχει.

ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη.....	5
Λέξεις κλειδιά.....	5
Abstract.....	6
Keywords.....	6
Περιεχόμενα.....	8
Περιεχόμενα εικόνων.....	13
Κεφάλαιο 1.....	15
Προγραμματιζόμενοι Λογικοί Ελεγκτές	
1.1.Εισαγωγή.....	15
1.2.Γενικά για τους προγραμματιζόμενους λογικούς ελεγκτές.	17
1.2.1.Πλεονεκτήματα προγραμματιζόμενων λογικών ελεγκτών.....	17
1.2.2.Στάδια εργασίας στον προγραμματιζόμενο αυτοματισμό.	18
1.2.3. Δομή προγραμματιζόμενου λογικού ελεγκτή.	19
1.2.3.1. Πλαίσιο τοποθέτησης μονάδων.....	19
1.2.3.2. Μονάδα τροφοδοσίας.....	22
1.2.3.3. Κεντρική μονάδα επεξεργασίας (CPU).....	23
1.2.3.4. Μονάδες εισόδων / εξόδων (I/O)	26
1.2.3.5. Κάρτα επικοινωνίας.....	31
1.2.4. Χρόνος κύκλου.....	32
1.2.5. Λειτουργικό σύστημα ελεγκτή προγραμματιζόμενης λογικής.....	32
1.2.6. Προσπέλαση προγράμματος.....	33
1.2.6.1. Μέθοδοι προσπέλασης προγράμματος.....	33
1.2.7. Δομή προγράμματος.....	35

1.2.7.1. Τύποι μπλοκ.....	35
1.2.8. Δομή μπλοκ.....	37
1.2.9. Μορφές προγραμματισμού.....	38
1.2.10. Γλώσσες προγραμματισμού.....	39
1.2.11. Σύγκριση μορφών προγραμματισμού.....	40
1.2.11.1. Πλεονεκτήματα λίστας εντολών (STL) σε σχέση με τις γραφικές μορφές (LAD, FBD)	40
1.2.11.2. Μειονεκτήματα λίστας εντολών (STL) σε σχέση με τις γραφικές μορφές (LAD, FBD)	41
1.2.11.3. Συμπεράσματα.	41
1.2.12. Θέση σε λειτουργία Προγραμματιζόμενου λογικού ελεγκτή.....	42
1.2.12.1. Θέση σε λειτουργία και έλεγχος των μονάδων ενός ελεγκτή (HARDWARE).....	43
1.2.13. Σύγκριση μεταξύ κλασσικού αυτοματισμού και μοντέρνου (PLC).....	46
1.3. Προγραμματισμός του PLC που εγκαταστήσαμε.....	48

Κεφάλαιο 2.....49

Περιγραφή της εφαρμογής

2.1.Εξήγηση λειτουργίας και κύκλος εργασίας.....	49
2.2.Αντλία.....	51
2.3.Δεξαμενές-Σωλήνωση-Καλωδίωση.....	52
2.4.Τροφοδοσίες.....	54
2.5.Έλεγχος στάθμης δεξαμενών με χρήση επιτηρητή.....	56
2.5.1.Περιγραφή Λειτουργίας.....	56
2.5.2.Τεχνικά Χαρακτηριστικά.....	57
2.5.3.Διάγραμμα Λειτουργίας.....	58
2.6.Ηλεκτροβαλβίδα	60

2.6.1.Γενικά.....	60
2.6.2.Αρχή λειτουργίας.....	60

Κεφάλαιο 3.....64

Προγραμματιζόμενος Λογικός Ελεγκτής LOGO!

3.1.Γενικά.....	64
3.2.Εγκατάσταση και καλωδίωση.....	67
3.3.Καταστάσεις λειτουργίας (RUN) και διακόπτης (STOP)	68
3.4.Προγραμματισμός LOGO!	70
3.5.Εισαγωγή προγράμματος στο LOGO!	72
3.6.Πρόγραμμα ελέγχου στάθμης υγρού δεξαμενών.....	80

Συμπεράσματα.....83

Βιβλιογραφία.....84

Παράρτημα Α.....86

LOGO! functions

4.1. Constants and connectors – Co.....	87
4.2. Basic functions list – GF.....	90
4.2.1. AND.....	91
4.2.2. AND with edge evaluation.....	91
4.2.3. NAND (not AND)	92
4.2.4. NAND with edge evaluation.....	93
4.2.5. OR.....	94
4.2.6. NOR (not OR)	94

4.2.7. XOR (exclusive OR)	95
4.2.8. NOT (Negation, Inverter)	96
4.3. Special functions.....	96
4.3.1. Designation of the inputs.....	97
4.3.2. Time response.....	98
4.3.3. Backup of the real – time clock.....	99
4.3.4. Retentivity.....	99
4.3.5. Parameter protection.....	99
4.3.6. Calculating the gain and offset of analog values.....	100
4.4. Special functions list – SF.....	102
4.4.1. On – delay.....	105
4.4.2. Off – delay.....	108
4.4.3. On/off – delay.....	109
4.4.4. Retentive on – delay.....	111
4.4.5. Wiping relay (pulse output)	112
4.4.6. Edge triggered wiping relay.....	113
4.4.7. Asynchronous pulse generator.....	115
4.4.8. Random generator.....	117
4.4.9. Stairway lighting switch.....	119
4.4.10. Multiple function switch.....	121
4.4.11. Weekly timer.....	124
4.4.12. Yearly timer.....	128
4.4.13. Up/down counter.....	134
4.4.14. Hours counter.....	137
4.4.15. Threshold trigger.....	141
4.4.16. Analog threshold trigger.....	144

4.4.17. Analog differential trigger.....	147
4.4.18. Analog comparator.....	150
4.4.19. Analog watchdog.....	155
4.4.20. Analog amplifier.....	158
4.4.21. Latching relay.....	159
4.4.22. Pulse relay.....	160
4.4.23. Message texts.....	162
4.4.24. Softkey.....	174
4.4.25. Shift register.....	177
4.4.26. Analog multiplexer.....	179
4.4.27. Analog ramp.....	181
4.4.28. PI controller.....	185
4.4.29. Pulse Width Modulator (PWM)	191
4.4.30. Analog math.....	194
4.4.31. Analog math error detection.....	197
Παράρτημα Β.....	199

Sirai L 133

ΠΕΡΙΕΧΟΜΕΝΑ ΕΙΚΟΝΩΝ

Εικ. 1.1 : Δομή προγραμματιζόμενου ελεγκτή.....	27
Εικ. 1.2 : Συμβατότητα μεταξύ των γλωσσών.....	42
Εικ. 2.1 : Σχηματική αναπαράσταση της κατασκευής ελέγχου στάθμης.....	50
Εικ. 2.2 : Σύστημα ελέγχου στάθμης υγρού με χρήση δεξαμενών.....	50
Εικ. 2.3 : Αντλία.....	52
Εικ. 2.4 : Δεξαμενές.....	53
Εικ. 2.5 : Σωλήνωση ηλεκτροβαλβίδας.....	53
Εικ. 2.6 : Ασφάλεια Schneider 2A.....	54
Εικ. 2.7 : Μετασχηματιστής εξόδου.....	55
Εικ. 2.8 : Βραχυκυκλωμένες κλέμες εισόδου.....	55
Εικ. 2.9 : Βραχυκυκλωμένες κλέμες εξόδου.....	56
Εικ. 2.10 : Πίνακας τεχνικών χαρακτηριστικών επιτηρητή.....	57
Εικ. 2.11 : Απεικόνιση αισθητήριων – επαφών.....	58
Εικ. 2.12 : Διάγραμμα λειτουργίας.....	58
Εικ. 2.13 : Επιτηρητές στάθμης.....	59
Εικ. 2.14 : Αισθητήρια.....	59
Εικ. 2.15 : Τεχνικά μέρη ηλεκτροβαλβίδας.....	61
Εικ. 2.16 : Ηλεκτροβαλβίδα.....	63
Εικ. 3.1 : PLC LOGO! 230RC.....	64
Εικ. 3.2 : Μορφή LOGO!	65
Εικ. 3.3 : Οι τύποι του LOGO!	66
Εικ. 3.4 : Ράγα τοποθέτησης LOGO!	66
Εικ. 3.5 : Εγκατάσταση LOGO!	67
Εικ. 3.6 : Καλωδίωση εισόδων και τροφοδοσίας PLC.....	68
Εικ. 3.7 : Καλωδίωση εξόδων PLC.....	68

Εικ. 3.8 : Περιπτώσεις επαναφοράς προγράμματος μετά από διακοπή τροφοδοσίας.....	70
Εικ. 3.9 : Πύλη OR.....	71
Εικ. 3.10 : Απεικόνιση block στο LOGO!	71
Εικ. 3.11 : LOGO!'s menus.....	73
Εικ. 3.12 : Basic functions.....	76
Εικ. 3.13 : Special functions.....	79

1. Προγραμματιζόμενοι Λογικοί Ελεγκτές

1.1.Εισαγωγή

Η τεχνική του αυτοματισμού μέχρι πριν τρεις δεκαετίες περίπου, βασιζόταν στο μεγαλύτερο μέρος της στα συστήματα συρματωμένης λογικής. Δηλαδή, η λειτουργία του αυτοματισμού στηριζόταν στην συρμάτωση ανάμεσα στα διάφορα στοιχεία (π.χ. επαφές, πηνία, ρελέ, χρονικά κ.λπ., για αυτοματισμούς με ρελέ, πύλες AND, πύλες OR κ.λπ., για αυτοματισμούς με ηλεκτρικές πλακέτες).

Επομένως, ένας πίνακας αυτοματισμού μπορούσε να κατασκευασθεί, μόνον όταν είχε τελειώσει οριστικά η μελέτη, δηλ. το συνδεσμολογικό σχέδιο. Επιπλέον, κάθε μετέπειτα αλλαγή είχε σαν αποτέλεσμα αντίστοιχη αλλαγή στην συρμάτωση (τροποποίηση του πίνακα), με τα γνωστά προβλήματα, π.χ. αν φτάνουν ή όχι οι ελεύθερες επαφές των ρελέ, αν υπάρχει χώρος για προσθήκη νέου υλικού κ.λπ.

Σήμερα, **τα συστήματα προγραμματιζόμενης λογικής**, με κύριους εκπρόσωπους στη βιομηχανία τους προγραμματιζόμενους λογικούς ελεγκτές, έχουν πια εκτοπίσει τελείως τα προηγούμενα συστήματα αυτοματισμού.

Οι ελεγκτές προγραμματιζόμενης λογικής έκαναν την εμφάνιση τους στην αγορά στα τέλη του 1960, επιβαλλόμενοι τόσο από τις παραγωγικές διαδικασίες που απαιτούσαν όλο και πιο πολύπλοκη λογική στα συστήματα ελέγχου, όσο και από την πρόοδο στην βιομηχανία της ηλεκτρονικής που μπορούσε να προμηθεύσει τους κατασκευαστές με τα απαραίτητα στοιχεία (μνήμες, επεξεργαστές, A/D μετατροπείς, κτλ.).

Τα πρώτα PLC είχαν την δυνατότητα επεξεργασίας μόνο ψηφιακών σημάτων και βέβαια ο προγραμματισμός τους δεν ήταν, όπως συνηθίσαμε να λέμε, φιλικός στον χρήστη. Για παράδειγμα, η αποθήκευση του προγράμματος γινόταν σε κασέτες. Η δεκαετία του 70 ήταν η εποχή όπου τα PLC άρχιζαν να εγκαθίστανται δειλά - δειλά στην βιομηχανία, ενώ η καθιέρωση τους ήρθε την επόμενη δεκαετία (1980). Η παρουσία του προσωπικού υπολογιστή από το 1982 και μετά, από την μία μεριά προσέφερε απεριόριστες δυνατότητες και ευκολίες στην δημιουργία και τον έλεγχο

προγραμμάτων, ενώ από την άλλη απείλησε τα PLC που κινδύνευαν προσωρινά από συστήματα ελέγχου με προσωπικούς υπολογιστές (PC based automation). Όμως το τοπίο ξεκαθάρισε οριστικά στα τέλη της δεκαετίας όπου τόσο τα PLC, όσο και οι Η/Υ απέκτησαν την θέση τους στις παραγωγικές διαδικασίες με σαφές πλεονέκτημα υπέρ των PLC.

Στα συστήματα προγραμματιζόμενης λογικής, η κατασκευή και συρμάτωση του πίνακα είναι ανεξάρτητη από τη λειτουργία που πρόκειται να εκτελέσει ο αυτοματισμός, με άλλα λόγια η μελέτη δεν αποτελεί προϋπόθεση. Πάνω στις κλέμες του ελεγκτή συνδέονται όλα τα στοιχεία, που δίνουν εντολές (τερματικοί διακόπτες, μπουτόν κλπ.), καθώς και όλα τα στοιχεία που δέχονται εντολές (πηνία, ρελέ ισχύος κινητήρων, λυχνίες κλπ.).

Η λειτουργία του αυτοματισμού προγραμματίζεται στην μνήμη του ελεγκτή, ακόμα και την τελευταία στιγμή, πριν από τη θέση σε λειτουργία. Επομένως η μελέτη (πρόγραμμα) μπορεί να γίνεται παράλληλα με την επιλογή του υλικού και την κατασκευή του πίνακα.

Αν στην συνέχεια χρειαστεί να γίνουν αλλαγές στη λειτουργία, γεγονός σύνηθες στον αυτοματισμό, τότε αυτές γίνονται πολύ απλά «διορθώνοντας» το πρόγραμμα, χωρίς να χρειαστεί να επέμβουμε τη συρμάτωση του πίνακα.

Αυτή η ευελιξία στις μετατροπές και οι πολλές τους δυνατότητες αποτελούν το σπουδαιότερο πλεονέκτημα των ελεγκτών προγραμματιζόμενης λογικής στην τεχνική του αυτοματισμού.

Τέλος, από τις αρχές του 1990 έως σήμερα η ανάπτυξη των ελεγκτών προγραμματιζόμενης λογικής είναι αλματώδης και χαρακτηρίζεται από δίκτυα υψηλών ταχυτήτων, ταχύτατους και πανίσχυρους κεντρικούς επεξεργαστές.

1.2. Γενικά για τους Προγραμματιζόμενους λογικούς ελεγκτές

1.2.1. Πλεονεκτήματα Προγραμματιζόμενων λογικών ελεγκτών.

- Μεγιστοποιούν την ταχύτητα της διαδικασίας παραγωγής και κατά συνέπεια μειώνουν πολύ γρήγορα το χρόνο απόσβεσης της εγκατάστασης.
- Στο στάδιο της μελέτης δεν υπάρχει πρόβλημα αν επαρκούν ή όχι οι επαφές των ρελέ, των χρονικών ή των εξωτερικών τερματικών.
- Η λειτουργία του αυτοματισμού μπορεί να αλλάξει πολύ εύκολα σε οποιοδήποτε στάδιο (μελέτη, κατασκευή, θέση σε λειτουργία ή αργότερα).
- Ο εντοπισμός βλαβών διευκολύνεται γιατί για κάθε εξωτερική εντολή υπάρχει αντίστοιχο LED. Επίσης, η ροή του αυτοματισμού μπορεί να παρακολουθείται άνετα, με τη βοήθεια μιας συσκευής προγραμματισμού.
- Έχουν τη δυνατότητα επέκτασης. Δηλαδή αν προστεθεί κάποια καινούρια διαδικασία ή χρειάζεται κάποιος επιπλέον έλεγχος, μπορεί με την προσθήκη των κατάλληλων καρτών εισόδων/εξόδων (I/O), το ίδιο το PLC να την πραγματοποιήσει.
- Ο αυτοματισμός παραδίνεται συντομότερα σε λειτουργία, επειδή η μελέτη μπορεί να γίνεται παράλληλα με την τοποθέτηση και συρμάτωση του ελεγκτή.
- Δεν υπάρχει το γνωστό πρόβλημα των «μη ενημερωμένων» σχεδίων του πίνακα αυτοματισμού μετά από λίγο καιρό. Ο ελεγκτής έχει πάντα κρατημένο «μέσα του» το τελευταίο πρόγραμμα, το οποίο μπορεί να διαβαστεί με μία συσκευή προγραμματισμού ή να εκτυπωθεί σε χαρτί.
- Υπάρχει σημαντική οικονομία στο χώρο, τη συντήρηση (δεν υπάρχουν μηχανικές επαφές) και την κατανάλωση ενέργειας.
- Η τοποθέτηση μπορεί να γίνει χωρίς κίνδυνο και μέσα σε πεδία ισχύος.
- Ένας ελεγκτής μπορεί να συνδεθεί με περιφερειακές μονάδες για επιτήρηση - έλεγχο της εγκατάστασης (οθόνη, εκτυπωτής, πληκτρολόγιο), καταργώντας το κλασικό μιμικό διάγραμμα και τον πίνακα χειρισμών. Επίσης, μπορεί να συνδεθεί με ηλεκτρονικό υπολογιστή για ανταλλαγή στοιχείων.

- Η γλώσσα προγραμματισμού είναι προσαρμοσμένη στο βιομηχανικό αυτοματισμό και άρα είναι προσιτή στο προσωπικό που μέχρι σήμερα συντηρούσε τους κλασσικούς πίνακες αυτοματισμού.
- Όλες οι κάρτες είναι τοποθετημένες στον ίδιο δίαυλο επικοινωνίας (bus), πράγμα που επιταχύνει την διαδικασία ανταλλαγής πληροφοριών.
- Τα PLC έχουν σχεδόν απεριόριστη διάρκεια ζωής λόγω των ασθενών ρευμάτων που χρησιμοποιούν. Ταυτόχρονα συμβάλουν στην οικονομικότερη, από πλευράς κατανάλωσης ηλεκτρικής ενέργειας, λειτουργία της εγκατάστασης.

1.2.2. Στάδια εργασίας στον προγραμματιζόμενο αυτοματισμό.

Στην επίλυση ενός προβλήματος αυτοματισμού με ένα ελεγκτή συναντάμε τρία στάδια. Παρακάτω περιγράφονται τα στάδια αυτά, ενώ παράλληλα παρουσιάζεται η αντιστοιχία που έχουν με σύστημα συρματωμένης λογικής (π.χ. αυτοματισμός με ρελέ).

Προγραμματιζόμενη λογική

1. Επιλογή ελεγκτή προγραμματιζόμενης λογικής.
2. Τοποθέτηση ελεγκτή σε πίνακα.

Συρμάτωση εξωτερικών στοιχείων της εγκατάστασης (τερματικοί, μπουτόν, διακόπτες, ενδ. λυχνίες, ρελέ ισχύος, κ.λπ.), πάνω στις κλέμες του ελεγκτή.

3. Προγραμματισμός.

Συρματωμένη λογική

1. Επιλογή βοηθητικού ρελέ, χρονικών, κ.λπ. που χρειάζονται για τον αυτοματισμό, βάσει του συνδεσμολογικού σχεδίου.
2. Απλή τοποθέτηση του υλικού (βοηθητικά ρελέ, χρονικά) μέσα στον πίνακα.

Συρμάτωση των εξωτερικών στοιχείων της εγκατάστασης ελεγκτή (τερματικοί, μπουτόν, διακόπτες, ενδ. λυχνίες, ρελέ ισχύος, βαλβίδες κ.λπ.) πάνω στην κλεμοσειρά του πίνακα.

3. Συρμάτωση του υλικού αυτοματισμού (βοηθητικά ρελέ, χρονικά) μέσα στον πίνακα μεταξύ τους, σύμφωνα με το συνδεσμολογικό σχέδιο.

1.2.3. Δομή Προγραμματιζόμενου λογικού ελεγκτή.

Η επιλογή ενός ελεγκτή προγραμματιζόμενης λογικής (τύπος -μέγεθος -κόστος) εξαρτάται κυρίως από το πλήθος των στοιχείων που δίνουν εντολή σ' αυτόν (είσοδοι) και το πλήθος των στοιχείων που δέχονται εντολή απ' αυτόν (έξοδοι), καθώς και από το πλήθος των λειτουργιών που απαιτείται να κάνει ο αυτοματισμός (μέγεθος προγράμματος, απαιτούμενη μνήμη και δυνατότητες της κεντρικής μονάδας).

Ανεξάρτητα, όμως, από τον τύπο και το μέγεθος, σε κάθε ελεγκτή συναντάμε τα εξής απαραίτητα στοιχεία:

- Πλαίσιο για τοποθέτηση των μονάδων
- Μονάδα τροφοδοσίας
- Κεντρική μονάδα (CPU) με τον μικροεπεξεργαστή και την μνήμη για το πρόγραμμα
- Μονάδες εισόδων
- Μονάδες εξόδων

1.2.3.1. Πλαίσιο τοποθέτησης μονάδων

Όλες οι μονάδες, από τις οποίες αποτελείται ένας ελεγκτής, πρέπει να τοποθετηθούν σε κάποιο πλαίσιο. Σ' αυτό είναι ενσωματωμένο το σύστημα ζυγών, δηλ. το σύστημα αγωγών, μέσω των οποίων επικοινωνούν οι διάφορες μονάδες μεταξύ τους για την ανταλλαγή πληροφοριών και για την τροφοδοσία τους.

Κεντρικό πλαίσιο (μόνο ένα)

Χαρακτηριστικό του είναι ότι μόνο σ' αυτό τοποθετείται η κεντρική μονάδα (CPU) και μάλιστα σε καθορισμένη θέση. Επίσης, εδώ τοποθετείται και η μονάδα τροφοδοσίας για ορισμένους τύπους ελεγκτών. Στις υπόλοιπες θέσεις του πλαισίου τοποθετούνται μονάδες εισόδων - εξόδων ή άλλες ειδικές μονάδες.

Σε μεγαλύτερα συστήματα ελεγκτών η μονάδα τροφοδοσίας δεν είναι σε μορφή κάρτας αλλά ερμαρίου και αποτελεί βασικό στοιχείο του κεντρικού πλαισίου.

Πλαίσιο επέκτασης (ένα ή περισσότερα)

Σ' αυτό τοποθετούνται μόνο μονάδες εισόδων - εξόδων. Χρησιμοποιείται αν δεν φτάνουν οι θέσεις του κεντρικού πλαισίου, για να χωρέσουν όλες οι απαιτούμενες μονάδες.

Συνδέεται με το κεντρικό (ή με άλλα πλαίσια επέκτασης), μέσω ειδικής μονάδας διασύνδεσης και καλωδίου.

Συγκεκριμένα το πλαίσιο δημιουργείται τοποθετώντας όσες «μονάδες πλαισίου» απαιτούνται τη μία πλάι στην άλλη και συνδέοντας τις. Κάθε τέτοια μονάδα πλαισίου έχει συγκεκριμένες θέσεις για τοποθέτηση μονάδων εισόδων - εξόδων.

Χωροταξική διάταξη πλαισίων ελεγκτή

Όπως είπαμε προηγουμένως, ένας ελεγκτής αποτελείται από ένα κεντρικό πλαίσιο και τα απαιτούμενα πλαίσια επέκτασης.

Ανάλογα με την απόσταση από το κεντρικό πλαίσιο, στην οποία τοποθετούνται τα πλαίσια επέκτασης, τα συστήματα διακρίνονται σε δύο περιπτώσεις:

- Κεντρικό σύστημα
- Αποκεντρωμένο σύστημα

Κεντρικό σύστημα

Το κεντρικό πλαίσιο και όλα τα πλαίσια επέκτασης βρίσκονται χωροταξικά σε μια θέση (π.χ. τοποθετημένα μέσα σ' έναν πίνακα). Η απόσταση του πιο μακρινού πλαισίου επέκτασης από το κεντρικό δεν επιτρέπεται να είναι μεγαλύτερη από π.χ. 2 - 2,5m.

Εκτός από τις καθορισμένες θέσεις των μονάδων τροφοδοσίας, CPU και διασύνδεσης, σε όλες τις άλλες θέσεις μπορούν να τοποθετηθούν μονάδες εισόδων - εξόδων.

Αποκεντρωμένο σύστημα

Στο αποκεντρωμένο σύστημα ορισμένα πλαίσια επέκτασης βρίσκονται σε μεγάλη απόσταση από το κεντρικό.

Ένα αποκεντρωμένο σύστημα έχει νόημα, αν η προς αυτοματισμό εγκατάσταση έχει τα στοιχεία της (π.χ. τερματικοί, μπουτόν, λυχνίες, ρελέ ισχύος, κ.λπ.) «μοιρασμένα» σε περισσότερα «κέντρα βάρους».

Σε μια τέτοια περίπτωση «αντιστοιχούμε» ένα ή περισσότερα πλαίσια επέκτασης σε κάθε ένα τέτοιο βάρος. Αποτέλεσμα είναι η μεγάλη οικονομία καλωδίωσης από τα στοιχεία της εγκατάστασης προς τις μονάδες εισόδων - εξόδων.

Η διασύνδεση δύο μακρινών μεταξύ τους πλαισίων επιτυγχάνεται μέσω ειδικών μονάδων διασύνδεσης σε κάθε πλαίσιο και ειδικού καλωδίου. Στα πλαίσια αυτά επίσης είναι απαραίτητο να υπάρχει και η μονάδα τροφοδοσίας.

Πάνω σε κάθε κλάδο της «αλυσίδας» επιτρέπεται να υπάρχουν μέχρι 4 «σημεία», στα οποία θα τοποθετηθούν τα πλαίσια επέκτασης. Στα 4 αυτά «σημεία» μπορούν να τοποθετηθούν περισσότερα από ένα πλαίσια με τη μέθοδο του κεντρικού συστήματος.

Σημείωση 1. Αν και οι διάφορες είσοδοι - έξοδοι δεν είναι χωροταξικά στην ίδια θέση, αυτό δεν ενοχλεί πουθενά το χρήστη, όταν φτιάχνει το πρόγραμμα. Προγραμματιστικά δηλαδή δεν γίνεται καμία διάκριση σε «κοντινές» και «μακρινές» I/O.

Σημείωση 2. Εκτός από τη μέθοδο αποκεντρωμένου συστήματος με τη μορφή της «αλυσίδας» υπάρχει και η λεγόμενη «ακτινική» μορφή. Σ' αυτήν μπορούν να συνδεθούν ακτινικά με το κεντρικό πλαίσιο μέχρι 3 αποκεντρωμένες θέσεις - κάθε μια σε απόσταση 1 Km. Σε κάθε αποκεντρωμένη θέση μπορεί να τοποθετηθούν περισσότερα από ένα πλαίσια με τη μέθοδο του κεντρικού συστήματος.

Σύγκριση κεντρικού - αποκεντρωμένου συστήματος

Σε μια εγκατάσταση με χωροταξικά «μαζεμένα» όλα τα στοιχεία της, χρησιμοποιούμε πάντα το κεντρικό σύστημα.

Αντίθετα, σε μια εγκατάσταση με κατανεμημένα χωροταξικά τα στοιχεία της σε περισσότερα «κέντρα βάρους», πρέπει να εξεταστεί οικονομοτεχνικά, αν συμφέρει το αποκεντρωμένο σύστημα.

1.2.3.2. Μονάδα τροφοδοσίας

Η μονάδα τροφοδοσίας χρησιμεύει για να δημιουργηθούν από την τάση του δικτύου οι απαραίτητες εσωτερικές τάσεις για την τροφοδοσία αποκλειστικά των ηλεκτρικών στοιχείων, που υπάρχουν μέσα στον ελεγκτή (τρανζίστορς, ολοκληρωμένα κ.λπ.). Επίσης για να διατηρηθεί το περιεχόμενο της μνήμης RAM σε μια διακοπή τάσης με τη βοήθεια μπαταρίας, που ενσωματώνεται σ' αυτή.

Τα σπουδαιότερα τεχνικά χαρακτηριστικά μιας μονάδας τροφοδοσίας είναι τα εξής:

Είσοδος

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

Έξοδος

Ονομαστική τάση, ονομαστικό ρεύμα, προστασία βραχυκυκλώματος.

Διάφορα

Μπαταρία για διατήρηση μνήμης RAM.

Αν το πρόγραμμα ενός ελεγκτή πρόκειται να αποθηκευτεί σε μνήμη RAM, τότε απαραίτητα πρέπει να υπάρχει στο σύστημα και μια μπαταρία για τη διατήρηση του περιεχομένου της μνήμης σε μια διακοπή τάσης του δικτύου. Αυτή ή μπαταρία, που είναι συνήθως λιθίου τοποθετείται στη μονάδα τροφοδοσίας και μπορεί να κρατήσει το πρόγραμμα της μνήμης RAM για πολύ μεγάλο χρονικό διάστημα. Πρέπει, όμως, να προσεχτούν τα εξής σημεία:

α) Η μπαταρία θα πρέπει να αλλάζεται με την συχνότητα που ορίζει ο κατασκευαστής.

β) Το SOFTWARE του ελεγκτή πρέπει να παρέχει στο χρήστη τη δυνατότητα να αξιολογήσει το γεγονός, ότι η μπαταρία έχει πέσει κάτω από το όριο ασφαλείας.

γ) Η αντικατάσταση της μπαταρίας πρέπει πάντοτε να γίνεται με τον ελεγκτή υπό τάση για να μην χαθεί το πρόγραμμα

1.2.3.3. Κεντρική μονάδα επεξεργασίας (CPU).

Η CPU έχει τα εξής χαρακτηριστικά:

- Ενσωματωμένη RAM εργασίας (Working Memory)
- Ενσωματωμένη RAM φορτώματος (Load memory)
- Εξωτερική Flash EPROM φορτώματος (Load memory) που επεκτείνει την ενσωματωμένη.

Η Load μνήμη περιλαμβάνει όλα τα Block Λογικής (συμπεριλαμβανομένων και Block που δεν απαιτούνται για την εκτέλεση του προγράμματος πχ. Block Header), μπλοκ Δεδομένων και Δεδομένων παραμετροποίησης (4 Kbytes) που δεν χάνονται ούτε με το Reset της μνήμης ούτε με την απώλεια μπαταρίας του τροφοδοτικού. Με την μεταγωγή της CPU από κατάσταση Stop - κατάσταση εκτέλεσης του προγράμματος μεταφέρονται από την Load μνήμη στην Working μνήμη μόνο τα κομμάτια των μπλοκ λογικής και δεδομένων που είναι απαραίτητα για την εκτέλεση του προγράμματος. Η working μνήμη είναι γρηγορότερη από την Load μνήμη και σβήνει με το μπουτόν Reset memory της CPU ή αν πέσει η μπαταρία του τροφοδοτικού.

Η CPU εμπεριέχει Status Leds και Leds σφαλμάτων ενώ ο τρόπος λειτουργίας επιλέγεται με κλειδί (KEY). Όταν το κλειδί μετακινηθεί ο τρόπος λειτουργίας της CPU δεν μπορεί να αλλάξει. Αυτή η δυνατότητα προστατεύει το πρόγραμμα της εφαρμογής από μη εξουσιοδοτημένη αλλαγή ή διαγραφή του.

Η CPU περιλαμβάνει διαγνωστική μνήμη που δεν σβήνεται ούτε με την πτώση τάσης ούτε με το Reset της μνήμης και καταγράφονται με ώρα και ημερομηνία γεγονότα που συνδέονται με:

- Σφάλματα της CPU.
- Σφάλματα συστήματος της CPU.
- Σφάλματα περιφερειακών modules.
- Μεταγωγή από κατάσταση Stop-Εκτέλεση προγράμματος (RUN) -Stop.
- Προγραμματιστικά λάθη στο πρόγραμμα εφαρμογής.

Η διαγνωστική μνήμη μπορεί να διαβαστεί ON-LINE τοπικά με τον ηλεκτρονικό υπολογιστή.

Επίσης η CPU περιλαμβάνει Διαγνωστικό Alarm μπλοκ στο οποίο προγραμματίζοντας την Διεύθυνση μιας οποιασδήποτε κάρτας εισόδου/εξόδου λαμβάνονται διαγνωστικά bit για την κάρτα όπως:

- Βλάβη κάρτας
- Εσωτερικό εξωτερικό σφάλμα
- Πρόβλημα σε κάποιο κανάλι της κάρτας
- Έλλειψη εξωτερικής τάσης
- Έλλειψη φίσσας καλωδίων, Bit που μπορούν συνολικά να ενημερώσουν τοπική λυχνία. Ειδικά στις κάρτες αναλογικών εισόδων αν στο στάδιο αρχικής παραμετροποίησης της κάρτας ενεργοποιήσει ο χρήστης την ανίχνευση κομμένου καλωδίου τότε είτε με την ενεργοποίηση του διαγνωστικού Alarm μπλοκ είτε με την μη ενεργοποίηση του αλλά οπτικά σε εξωτερικό LED της κάρτας (System Fault) ειδοποιείται τοπικά ή remote το σύστημα για το κομμένο καλώδιο οποιοδήποτε αναλογικού οργάνου(4.20mA)
- Υπάρχει ενσωματωμένο ρολόι πραγματικού χρόνου
- Υποστηρίζονται Γλώσσες προγραμματισμού όπως LAD (LADDER) FBD (Πύλες) STL (λίστα εντολών) σύμφωνα με τα διεθνή Standards IEC 1131-3 Part 3 αλλά και επιπλέον γλώσσες προγραμματισμού με την χρήση Optional Software πακέτων.

- Όλες οι επιπλέον γλώσσες προγραμματισμού με μικρό ποσό μετάφρασης (Compilation) μεταφράζονται στις γλώσσες LAD. FBD. STL.

Υποστηρίζεται δομημένος προγραμματισμός με την ύπαρξη ειδικών μπλοκ οργάνωσης (OB) Block δεδομένων (DB, Block λειτουργία (FC, FB) Block Λειτουργιών συστήματος (SFC, SFB) και Block δεδομένων συστήματος (SDB).

- Υποστηρίζονται οι παρακάτω εντολές

- Λογικής bit BOOLEAN (AND, OR)

- Λογικής Word boolean (AND, OR) με 16 bit- Σταθερές.

- Λογικής Double Boolean (AND,OR) με 32 bit- Σταθερές

- Εντολές παλμού.

- Set / Reset bit (πχ. Inputs, Outputs, Memorys)

- Εντολές ολίσθησης Δεξιά, αριστερά και κυκλικής ολίσθησης.

- Set /Reset bit (π.χ. Inputs, Outputs, Memorys)

- Εντολές ολίσθησης δεξιά, αριστερά και κυκλικής ολίσθησης

- Εντολές χρονικών και απαριθμητών

- Αποθήκευσης και μεταφοράς τιμών από και προς καταχωρητές byte, Word, Doubleword.

- Εντολές σύγκρισης (16bit, 32 bit ακέραιων αριθμών, 32 bit δεκαδικών αριθμών).

- Αριθμητικές πράξεις όπως α) Πρόσθεση, β) πολλαπλασιασμό, 16bit ακέραια, γ)

Πρόσθεση/πολλαπλασιασμό 32 bit ακέραια, δ) Πρόσθεση/πολλαπλασιασμό 32 bit δεκαδικών

- Εύρεση τετραγωνικής ρίζας, Λογαριθμικές πράξεις, τριγωνομετρικές λειτουργίες.

- Εντολές αλλαγής ελέγχου του προγράμματος από μπλοκ σε μπλοκ και από εντολή σε εντολή μέσα στο ίδιο μπλοκ .
- Εντολές μετατροπής κώδικα (πχ BCD σε 16 bit Ακέραια)
- Διάφοροι τρόποι εκτέλεσης του προγράμματος όπως κυκλικός, ελεγχόμενος από γεγονός ή από χρόνο
- Ένδειξη μεγίστου - ελαχίστου- μέσου κύκλου εκτέλεσης προγράμματος

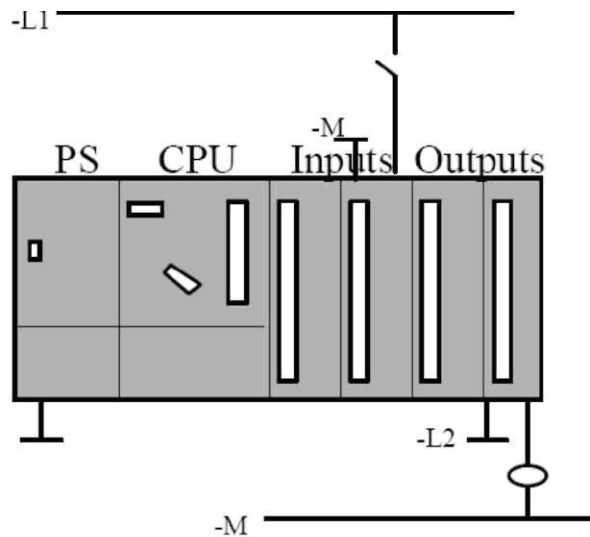
Στην κεντρική μονάδα (CPU) ο μικροεπεξεργαστής προσπελαύνει συνεχώς (κυκλικά) το πρόγραμμα, που είναι γραμμένο στην μνήμη. Ρωτάει συνεχώς, αν οι διάφορες είσοδοι έχουν ή δεν έχουν τάση (επαφές κλειστές ή ανοιχτές), επεξεργάζεται τις εντολές του προγράμματος και βάσει αυτών εξαναγκάζει τις εξόδους να διεγερθούν ή όχι (δηλ. να αποκτήσουν ή όχι τάση, οπότε διεγείρονται ή όχι τα συνδεδεμένα σ' αυτές ρελέ κ.λπ.).

Το πρόγραμμα γράφεται στη μνήμη με τη βοήθεια ενός ηλεκτρονικού υπολογιστή εξοπλισμένου με υλικό προγραμματισμού (CX-Programmer). Αυτό συνδέεται στην κεντρική μονάδα μόνο όταν πρόκειται να γραφτεί ή να μεταφερθεί το πρόγραμμα στην μνήμη, ή αν πρόκειται να γίνουν αλλαγές. Επίσης, χρησιμοποιείται για τον έλεγχο των διαφόρων σημάτων κατά την εξέλιξη του προγράμματος και για την ανεύρεση σφαλμάτων.

1.2.3.4. Μονάδες εισόδων / εξόδων (I/O)

Τα καλώδια που έρχονται από τα αισθητήρια της παραγωγικής διαδικασίας (τερματικοί, μπουτόν, διακόπτες), συνδέονται στις κλέμες των μονάδων εισόδων (είσοδοι του ελεγκτή).

Αντίστοιχα, τα καλώδια που πηγαίνουν προς τα ρελέ ισχύος, βαλβίδες, λυχνίες κ.λπ. συνδέονται στις κλέμες των μονάδων εξόδου (έξοδοι του ελεγκτή).



Εικ. 1.1 Δομή προγραμματιζόμενου ελεγκτή. PS = τροφοδοτικό, CPU = κεντρική μονάδα επεξεργασίας, Inputs = εισόδοι, Outputs = εξόδοι., -M = ο ακροδέκτης M του τροφοδοτικού (αρνητικός πόλος)

Ο ελεγκτής αντιλαμβάνεται ότι ένα αισθητήριο είναι ανοιχτό ή κλειστό από το αν εμφανίζεται τάση L1 στην αντίστοιχη κλέμα εισόδου. Επίσης, αν από το πρόγραμμα δοθεί εντολή για διέγερση π.χ. ενός ρελέ, τότε εμφανίζεται η τάση L2 στην αντίστοιχη κλέμα εξόδου.

Οι τάσεις L1, L2 δεν παρέχονται από την μονάδα τροφοδοσίας του ελεγκτή και πρέπει να δημιουργηθούν από εμάς με κατάλληλο τροφοδοτικό (για DC) ή μετασχηματιστή τάσης χειρισμού (για AC).

Μονάδες ψηφιακών εισόδων

Ένας ελεγκτής αντιλαμβάνεται ότι μια εξωτερική επαφή (π.χ. τερματικός) έκλεισε, όταν στην αντίστοιχη κλέμα εισόδου εμφανίζεται τάση. Η τάση αυτή ονομάζεται τάση εισόδων.

Η τάση για την τροφοδοσία των εισόδων δεν δημιουργείται από τη μονάδα τροφοδοσίας του ελεγκτή, αλλά πρέπει να τη δημιουργήσουμε εμείς με κατάλληλο τροφοδοτικό (για DC) ή μετασχηματιστή τάσης χειρισμού (για AC). Εξαιρέση αποτελούν συνήθως οι πολύ μικροί ελεγκτές, στους οποίους ο κατασκευαστής μπορεί να έχει ενσωματώσει ένα μικρό τροφοδοτικό.

Σημείωση.

Μια μονάδα εισόδων έχει συνήθως 4, 8, 16 ή 32 εισόδους, ανάλογα με τον τύπο του ελεγκτή και τη τάση. Υποχρεωτικό είναι να χρησιμοποιείται η ίδια τάση για όλες τις εισόδους μιας μονάδας εισόδων. Για μια άλλη όμως μονάδα εισόδων μπορεί να χρησιμοποιηθεί διαφορετική τάση.

Μονάδες ψηφιακών εξόδων

Οι μονάδες ψηφιακών εξόδων χρησιμεύουν για τη διέγερση των εξωτερικών στοιχείων της εγκατάστασης, όπως ρελέ κινητήρων, ενδεικτικές λυχνίες κλπ. Όταν από το πρόγραμμα δοθεί εντολή για τη διέγερση ενός π.χ. εξωτερικού ρελέ, τότε κλείνει ο αντίστοιχος «διακόπτης» της εξόδου. Η τάση εμφανίζεται στην κλέμα εξόδου και το ρελέ οπλίζει. Η τάση αυτή ονομάζεται τάση εξόδων. Ο «διακόπτης» εξόδου είναι συνήθως ηλεκτρονικός (τρανζίστορ, triac), αλλά μπορεί να είναι και μηχανική επαφή μικρορελέ.

Η τάση για την τροφοδοσία των μονάδων εξόδων δεν δημιουργείται από τη μονάδα τροφοδοσίας του ελεγκτή, αλλά πρέπει να τη δημιουργήσουμε εμείς με κατάλληλο τροφοδοτικό (για DC) ή μετασχηματιστή τάσης χειρισμού (για AC).

Σημείωση.

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. Συνήθως μια μονάδα εξόδων περιλαμβάνει 4 ή 8 ή 16 ή 32 εξόδους, ανάλογα με τον τύπο του ελεγκτή και την τάση.

Τα σπουδαιότερα τεχνικά χαρακτηριστικά μίας μονάδας εξόδων είναι: το πλήθος των εξόδων, η γαλβανική απομόνωση, η ονομαστική τάση, οι ανοχές τάσης, το ονομαστικό ρεύμα κάθε εξόδου, το ελάχιστο ρεύμα φορτίου, η ταυτόχρονη φόρτιση εξόδων μιας ομάδας, η προστασία εξόδων, η μέγιστη συνολική διαδρομή καλωδίων,

η συχνότητα ζεύξεων, το ρεύμα που απορροφά η μονάδα συνολικά από τα εσωτερικά 5 V, η απαιτούμενη πρίζα καλωδίων (τύπος).

Αν γίνει κάποιο βραχυκύκλωμα στο εξωτερικό κύκλωμα μιας εξόδου, τότε χρειάζεται η «επέμβαση» της προστασίας που υπάρχει μέσα στη μονάδα. Στην απλούστερη περίπτωση, η προστασία αυτή είναι μια ασφάλεια υπερταχείας τήξης και μάλιστα μία ανά μονάδα εξόδων. Αντίθετα, στις μονάδες DC 24V, είναι συνηθισμένη η «ηλεκτρονική» προστασία. Σ' αυτή την περίπτωση, όταν συμβεί ένα εξωτερικό βραχυκύκλωμα, η προστασία μειώνει ή και μηδενίζει ακόμα την τάση εξόδου της μονάδας, ώστε το ρεύμα εξόδου να μην υπερβεί το μέγιστο επιτρεπόμενο.

Μονάδες αναλογικών εισόδων

Η αναλογική κάρτα εισόδων μπορεί να επεξεργασθεί αισθητήρια με

δυνατότητα μετρήσεων βασικών περιοχών τάσης

$\pm 1V / 200 K\Omega$ Αντίσταση εισόδου

$1.5V/200 K\Omega$ Αντίσταση εισόδου και περιοχών ρεύματος

$4.20 mA/80\Omega$ Αντίσταση εισόδου

$\pm 20mA/ 80\Omega$ Αντίσταση εισόδου

- Η ανάλυση του A/D μετατροπέα της κάρτας είναι 12 bits
- Ο κύκλος ολοκλήρωσης/μετατροπής για κάθε κανάλι 2.5/100 msec
- Το μήκος καλωδίου μέχρι το αισθητήριο θα είναι τουλάχιστον 200m με μπλενταρισμένο καλώδιο
- Έχει γαλβανική απομόνωση
- Προστασία έναντι ανάστροφου πολικότητας
- Επιτρεπτή τάση εισόδου για κανάλι τάσης 20V

- Επιτρεπτό ρεύμα εισόδου για κανάλι ρεύματος 40mA
- Αντιστάθμιση Θερμοκρασίας : εσωτερική ή εξωτερικό με Module αντιστάθμισης.
- Όριο σφάλματος λειτουργίας (πάνω από την περιοχή θερμοκρασίας που αναφέρεται στην περιοχή εισόδου) max +-1%
- Όριο Βασικού σφάλματος (Όριο σφάλματος λειτουργίας στα 25° που αναφέρεται στην περιοχή εισόδου) max +- 0.6 %
- Δυνατότητα διάγνωσης μέσω κόκκινου Led για σφάλματα καναλιών
- Φίσα καλωδίων με στοιχείο κωδικοποίησης. Όταν η φίσα τοποθετείται για πρώτη φορά στην κάρτα τότε το στοιχείο κωδικοποίησης επιδρά στο να μπορεί να τοποθετηθεί η φίσα σε κάρτες της ίδιας περιοχής τάσης ή ρεύματος

Μονάδες αναλογικών εξόδων

Η αναλογική κάρτα εξόδων έχουν:

- Τάση τροφοδοσίας 24VDC
- Γαλβανικά απομονωμένη
- Περιοχές εξόδου τάσης $\pm 10V$, 0-10V 1-5V και περιοχές εξόδου ρεύματος 4.20mA, $\pm 20mA$, 0-20mA που μπορούν να αλλαχθούν με μηχανικά Jumpers πάνω στην κάρτα ενώ διάφορες ρυθμίσεις μπορούν να γίνουν από το Software.
- Αντίσταση φορτίου για τα κανάλια τάσης min 1 K Ω
- Αντίσταση φορτίου για τα κανάλια ρεύματος max 0.5 K Ω .
- Χωρητικά φορτία max 1 μF
- Επαγωγικά φορτία max 1 mH
- Προστασία από βραχυκύκλωμα με ρεύμα βραχυκύκλωσης 25mA για εξόδους τάσης.

- Ισχύς εξόδου με τάση ανοικτού κυκλώματος 18V
- Η ανάλυση του D/A Converter είναι 11 bits+Πρόσημο ($\pm 10V$, 4.20mA, $\pm 20mA$, 1-5V) , 12 Bits (0-10V, 0-20mA)
- Ο κύκλος μετατροπής για κάθε κανάλι είναι max 0.8 msec και ειδικά για Ωμικά φορτία 0.1 ms για χωρητικά φορτία 3.3 ms για επαγωγικά φορτία 0.5. ms
- Όρια λειτουργίας (0.60°C στην περιοχή εξόδου) Τάση $\pm 0,5\%$
Ρεύμα $\pm 0,6\%$
- Βασικό σφάλμα (0-25°C στην περιοχή εξόδου) Τάση $\pm 0,2\%$
Ρεύμα $\pm 0,3\%$
- Το μήκος καλωδίου μέχρι το στοιχείο ενεργοποίησης είναι 200m με μπλενταρισμένο καλώδιο
- Φίσσα καλωδίου με την ίδια λογική όπως της αναλογικής κάρτας εισόδου.
- Δυνατότητα διάγνωσης μέσω κόκκινου Led για σφάλματα καναλιών

1.2.3.5. Κάρτα επικοινωνίας

- Interfaces 1 (RS232)
- ταχύτητα μετάδοσης 2.4 Kbit/sec -19.2 Kbit/sec
- LEDS για SEND-RECEIVE- ERROR
- Πρωτόκολλα : ASCII (μήκος πληροφορίας max 1024 bytes , ταχύτητα 9.6 Kbit/sec), 3964R(μήκος πληροφορίας max 1024 bytes , ταχύτητα 19.2 Kbit/sec). RK512 (με ειδικό Software εφαρμογή), Printer Driver (ταχύτητα 9.6 Kbit/sec)

Ο χρήστης με την βοήθεια πακέτου παραμετροποίησης που φορτώνεται στο βασικό Software , καθορίζει το πρωτόκολλο και τα χαρακτηριστικά του και αυτά υπό μορφή System Data φορτώνονται στην Load Memory της CPU.

Το PLC ικανοποιεί τα βιομηχανικά Standards λόγω της συμβατότητας τους σε υψηλά ηλεκτρομαγνητικά πεδία, της λειτουργίας του σε μεγάλη θερμοκρασία 0.60°C και της αντοχής του σε κραδασμούς, δεν απαιτεί κατά την λειτουργία του επιπλέον ανεμιστήρες.

1.2.4. Χρόνος κύκλου

Σαν χρόνος κύκλου ορίζεται ο χρόνος που απαιτείται για την επεξεργασία και εκτέλεση (μία φορά) των εντολών που είναι γραμμένες στην μνήμη, από την πρώτη μέχρι την τελευταία. Είναι αυτονόητο, ότι όσο πιο μεγάλο είναι το πρόγραμμα τόσο μεγαλύτερος είναι και ο χρόνος κύκλου.

Για να μπορούν να συγκριθούν διαφορετικοί τύποι ελεγκτών, οι κατασκευαστές δίνουν συνήθως τον μέσο χρόνο κύκλου για 1 K (= 1024) εντολές προγράμματος. Όσο μικρότερος είναι αυτός ο χρόνος, τόσο ταχύτερος είναι ο ελεγκτής. Η έννοια μέσος χρόνος κύκλου οφείλεται στο γεγονός ότι όλες οι εντολές δεν απαιτούν ίδιο χρόνο για την επεξεργασία τους. Έτσι, ανάλογα με τις εντολές που θα χρησιμοποιήσουμε, διαφοροποιείται και ο χρόνος αυτός.

Συνήθως ο χρόνος κύκλου για ένα 1 K εντολές είναι της τάξεως μερικών msec.

1.2.5. Λειτουργικό σύστημα ελεγκτή προγραμματιζόμενης λογικής

Ορισμένες ενέργειες του ελεγκτή γίνονται αυτόματα, χωρίς καμία απολύτως εντολή από το χρήστη, π.χ.

- Όταν επανέρχεται η τάση μετά από μία διακοπή του δικτύου γίνεται μηδενισμός των βοηθητικών που ανήκουν στην περιοχή μνήμης χωρίς συγκράτηση.
- Πριν από την έναρξη κάθε κύκλου προγράμματος γίνεται μεταφορά σημάτων από τις κλέμες των μονάδων εισόδων στη μνήμη απεικόνισης καταστάσεων εισόδων (Process - Image Input Register).

- Μετά το τέλος κάθε κύκλου προγράμματος γίνεται μεταφορά της μνήμης απεικόνισης καταστάσεων εξόδων (Process - Image Output Register) στις αντίστοιχες κλέμες των μονάδων εξόδων.

Όλες αυτές οι απαραίτητες ενέργειες, οι οποίες προσδιορίζουν τι άλλο πρέπει να κάνει ο ελεγκτής παράλληλα με το κυρίως πρόγραμμα του χρήστη, το οποίο εμείς προγραμματίζουμε, αποτελούν το λειτουργικό πρόγραμμα του ελεγκτή. Το πρόγραμμα αυτό είναι συνήθως αποθηκευμένο σε μια μνήμη ROM μέσα στην κεντρική μονάδα και περιέχει εντολές που δεν μπορούν να διαβαστούν από εμάς, και οι οποίες καθορίζουν τις παραπάνω αντιδράσεις.

1.2.6. Προσπέλαση προγράμματος

Το γενικό πρόγραμμα μιας κεντρικής μονάδας αποτελείται από το λειτουργικό σύστημα και το πρόγραμμα του χρήστη.

Το λειτουργικό σύστημα όπως προαναφέραμε, αποτελεί το σύνολο που περιέχει όλες τις εντολές και τις δηλώσεις που ελέγχουν τις πηγές του συστήματος, τις διαδικασίες που χρησιμοποιούν αυτές τις πηγές, καθώς και περιοχές λειτουργίας όπως αποθήκευση δεδομένων στην περίπτωση πτώση της τάσης του δικτύου, ενεργοποίηση τάσεων προτεραιότητας, κλπ. Το λειτουργικό σύστημα αποτελεί ένα μέρος της κεντρικής μονάδας, στο οποίο ο χρήστης δεν έχει πρόσβαση γραφής. Εντούτοις, μπορούμε να φορτώσουμε ξανά το σύστημα αυτό από μια μονάδα μνήμης, π.χ. στην περίπτωση της ενημέρωσης με τις τελευταίες αλλαγές του προγράμματος.

Το πρόγραμμα του χρήστη αποτελεί το σύνολο όλων των εντολών και δηλώσεων, στην περίπτωση αυτή τα στοιχεία του προγράμματος, για την οδήγηση των σημάτων, μέσα από την οποία η όλη διαδικασία επηρεάζεται ανάλογα με τη προκαθορισμένη εργασία ελέγχου.

1.2.6.1. Μέθοδοι προσπέλασης προγράμματος

Το πρόγραμμα του χρήστη μπορεί να αποτελείται από διάφορα μέρη τα οποία η κεντρική μονάδα επεξεργασίας εκτελεί ανάλογα με το αν λαμβάνουν χώρα κάποια

συγκεκριμένα γεγονότα. Ένα τέτοιο γεγονός μπορεί να είναι η εκκίνηση του αυτόματου συστήματος, μια διακοπή ή η ανίχνευση σφάλματος. Τα προγράμματα που εξαρτώνται από τέτοια γεγονότα χωρίζονται σε τάξεις προτεραιότητας που καθορίζουν την σειρά εκτέλεσης των μερών του προγράμματος όταν συμβαίνουν συγκεκριμένα γεγονότα.

Το χαμηλής τάξης προτεραιότητας πρόγραμμα είναι το κυρίως πρόγραμμα, το οποίο προσπελαύνεται κυκλικά από την κεντρική μονάδα. Όλα τα άλλα γεγονότα μπορούν να διακόψουν το κυρίως πρόγραμμα σε οποιοδήποτε σημείο. Η CPU τότε εκτελεί την ανάλογη ρουτίνα διακοπής ή ρουτίνα αποσφαλμάτωσης και επιστρέφει στο κυρίως πρόγραμμα.

Ένα συγκεκριμένο μπλοκ οργάνωσης (organization block OB) συναντάται σε κάθε γεγονός. Τα μπλοκ οργάνωσης αναπαριστούν τις τάξεις προτεραιότητας στο πρόγραμμα του χρήστη. Όταν ένα γεγονός λαμβάνει χώρα, η κεντρική μονάδα καλεί το ανάλογο μπλοκ οργάνωσης. Ένα τέτοιο μπλοκ αποτελεί ένα μέρος του προγράμματος του χρήστη.

Πριν ακόμα ξεκινήσει η CPU να εκτελεί το κυρίως πρόγραμμα, εκτελεί μια ρουτίνα εκκίνησης. Αυτή η ρουτίνα μπορεί να ενεργοποιηθεί με το άνοιγμα της κεντρικής τάσης τροφοδοσίας, από τον επιλογικό διακόπτη που υπάρχει στην CPU ή μέσω ενός προγραμματιστή.

Το κυρίως πρόγραμμα βρίσκεται μέσα στο μπλοκ οργάνωσης OB1, το οποίο η κεντρική μονάδα επεξεργασίας εκτελεί. Αφού έχει τελειώσει η εκτέλεση του OB1 (τέλος προγράμματος), η CPU επιστρέφει στο λειτουργικό σύστημα και μόλις καλέσει για εκτέλεση διάφορες λειτουργίες του συστήματος, όπως ενημέρωση του πίνακα διευθύνσεων των εξόδων, καλεί, ξανά, για άλλη μια φορά το OB1.

Γεγονότα που μπορούν να διακόψουν το πρόγραμμα είναι οι αιτήσεις διακοπών και τα σφάλματα. Οι διακοπές μπορούν να ζητηθούν από την διαδικασία (hardware interrupts) ή από την CPU (διακοπές εποπτείας). Όσον αφορά τα σφάλματα, υπάρχει ένας διαχωρισμός μεταξύ συγχρονισμένων και ασύγχρονων σφαλμάτων. Το ασύγχρονο σφάλμα είναι ανεξάρτητο από τον κύκλο του προγράμματος, για

παράδειγμα διακοπή τροφοδοσίας σε μια μονάδα επέκτασης ή διακοπή που έχει προκληθεί από την αντικατάσταση της μονάδας. Το σύγχρονο σφάλμα προκαλείται από την εκτέλεση του προγράμματος, όπως η αναφορά σε μη υπαρκτή διεύθυνση ή η δημιουργία σφάλματος μετατροπής τύπων δεδομένων. Ο τύπος και ο αριθμός των καταχωρημένων γεγονότων, καθώς και τα ανάλογα μπλοκ οργάνωσης εξαρτώνται από την κεντρική μονάδα επεξεργασίας.

1.2.7. Δομή προγράμματος

Μπορούμε να χωρίσουμε το πρόγραμμα σε όσα μέρη θέλουμε με σκοπό να το διαβάζουμε και να το αντιλαμβανόμαστε καλύτερα και ευκολότερα. Κάθε μέρος του προγράμματος πρέπει να έχει τεχνολογική και λειτουργική βάση. Αυτού του είδους τα μέρη ονομάζονται «Μπλοκ». Ένα μπλοκ αποτελεί ένα μέρος του προγράμματος του χρήστη που καθορίζεται από τις λειτουργίες του, τη δομή και τον σκοπό της ύπαρξης του.

1.2.7.1. Τύποι μπλοκ

Υπάρχουν διάφοροι τύποι μπλοκ για διάφορους σκοπούς:

- Μπλοκ χρήστη. Τα μπλοκ αυτά περιέχουν το πρόγραμμα και τα δεδομένα του χρήστη.
- Μπλοκ συστήματος. Τα μπλοκ αυτά περιέχουν το πρόγραμμα και τα δεδομένα του συστήματος.
- Στάνταρτ μπλοκ. Τα μπλοκ αυτά αποτελούν το κλειδί λειτουργίας των οδηγών (drivers) των ειδικών καρτών.

Μπλοκ χρήστη

Τα μεγάλα και περίπλοκα προγράμματα «δομούνται» (διαχωρίζονται) σε μπλοκ τα οποία εν μέρη είναι απαραίτητα. Μπορούμε να διαλέξουμε μεταξύ των διαφόρων τύπων των μπλοκ, ανάλογα με την εφαρμογή:

- Μπλοκ οργάνωσης (OB).

Τα προαναφερόμενα μπλοκ συμβάλουν στην επικοινωνία μεταξύ του λειτουργικού συστήματος και του προγράμματος του χρήστη. Οι κεντρικές μονάδες επεξεργασίας καλούν τα μπλοκ οργάνωσης όταν συγκεκριμένα γεγονότα λαμβάνουν χώρα. π.χ. στην περίπτωση διακοπής. Το κυρίως πρόγραμμα βρίσκεται στο μπλοκ οργάνωσης OB1. Τα άλλα μπλοκ οργάνωσης έχουν συγκεκριμένους αριθμούς βασισμένους στο είδος των γεγονότων, τα οποία καλούνται να χειριστούν.

- Μπλοκ λειτουργίας (FB).

Αποτελούν μέρος του προγράμματος του οποίου οι κλήσεις μπορούν να προγραμματιστούν μέσω παραμέτρων του μπλοκ. Οι μεταβλητές μνήμης που περιέχονται σε ένα μπλοκ δεδομένων το οποίο με την σειρά του περιλαμβάνεται στην κλήση του μπλοκ λειτουργίας. Επίσης είναι δυνατόν σε κάθε κλήση να περιέχεται και διαφορετικό μπλοκ δεδομένων (με την ίδια δομή δεδομένων άλλα διαφορετικές τιμές μεταβλητών).

- Μπλοκ δεδομένων (DB).

Αυτά τα μπλοκ περιέχουν τα δεδομένα του προγράμματος μας. Προγραμματίζοντας τα καθορίζουμε σε ποια μορφή θα σωθούν τα δεδομένα (σε ποιο μπλοκ, με ποια σειρά και με τι τύπο δεδομένων). Υπάρχουν δύο τρόποι χρήσης των μπλοκ δεδομένων: ως καθολικά και ως στιγμιαία μπλοκ. Ένα καθολικό μπλοκ δεδομένων είναι ένα «ελεύθερο» μπλοκ μέσα στο πρόγραμμα του χρήστη και δεν περιέχεται σε ένα κωδικοποιημένο μπλοκ. Ένα στιγμιαίο μπλοκ δεδομένων όμως, περιέχεται σε ένα μπλοκ λειτουργίας και αποθηκεύει μέρος των δεδομένων του μπλοκ λειτουργίας.

Ο αριθμός των μπλοκ ανά τύπο μπλοκ και το μήκος τους εξαρτάται από την CPU. Οι αριθμοί των μπλοκ οργάνωσης και το πλήθος τους είναι καθορισμένα. Αναθέτονται από το λειτουργικό σύστημα της κεντρικής μονάδας . Μπορούμε να ορίσουμε μόνοι μας τον αριθμό του μπλοκ των άλλων ειδών των μπλοκ, αρκεί αυτός να βρίσκεται μέσα σε καθορισμένα όρια. Επίσης έχουμε την επιλογή να ονομάσουμε κάθε μπλοκ μέσω του πίνακα συμβόλων και στη συνέχεια να αναφερόμαστε σ' αυτά με το όνομα τους.

Μπλοκ συστήματος

Τα μπλοκ συστήματος αποτελούν μέρος του λειτουργικού συστήματος. Μπορούν να περιέχουν προγράμματα (λειτουργίες συστήματος) ή μπλοκ λειτουργιών ή δεδομένα (μπλοκ δεδομένων συστήματος). Τα μπλοκ συστήματος πραγματοποιούν έναν αριθμό από σημαντικές λειτουργίες του συστήματος, προσβάσιμες στο χρήστη, όπως είναι ο χειρισμός του εσωτερικού ρολογιού της CPU, ή οι διάφορες λειτουργίες επικοινωνίας.

Μπορούμε να καλέσουμε τις λειτουργίες του συστήματος και τα μπλοκ λειτουργιών του συστήματος, αλλά δεν μπορούμε να τα διαμορφώσουμε ή να τα προγραμματίσουμε. Τα μπλοκ από μόνα τους δεν διατηρούν χώρο στην μνήμη. Μόνο οι κλήσεις των μπλοκ και τα στιγμιαία μπλοκ δεδομένων των μπλοκ λειτουργιών του συστήματος είναι στην μνήμη.

1.2.8. Δομή μπλοκ

Συνήθως τα μπλοκ αποτελούνται από τρία μέρη :

- Τον αριθμό του μπλοκ που περιέχει τις ιδιότητες του μπλοκ, όπως το όνομα του.
- Το μέρος των δηλώσεων όπου οι τοπικές μεταβλητές του μπλοκ δηλώνονται.
- Το μέρος του προγράμματος όπου περιέχονται οι εντολές του προγράμματος.

Ένα μπλοκ δεδομένων έχει παρόμοια μορφή:

- Την επικεφαλίδα του μπλοκ που περιέχει τις ιδιότητες του μπλοκ.
- Το μέρος των δηλώσεων που περιέχει τις δηλώσεις των τοπικών λειτουργιών του μπλοκ, στην περίπτωση αυτή τις διευθύνσεις των δεδομένων με τους τύπους των δεδομένων.
- Το μέρος της αρχικοποίησης, στο οποίο μπορούν να οριστούν αρχικές τιμές για μεμονωμένες διευθύνσεις δεδομένων.

1.2.9. Μορφές προγραμματισμού

Για να αναλύσουμε έναν περίπλοκο αυτοματισμό θα πρέπει να χωρίσουμε την εφαρμογή σε μικρότερα μέρη ανάλογα με την δομή της διαδικασίας που πρέπει να ελεγχθεί. Μετά μπορούμε να διαμορφώσουμε τα επιμέρους κομμάτια καθορίζοντας τις λειτουργίες και διοχετεύοντας τα εσωτερικά σήματα προς την διαδικασία ή άλλα μέρη. Αυτός ο διαχωρισμός μπορεί να εφαρμοστεί και στον προγραμματισμό μας. Μ' αυτόν τον τρόπο η δομή του προγράμματος μας ανταποκρίνεται στον διαχωρισμό της εφαρμογής.

Ένα τέτοιο πρόγραμμα μπορεί να διαμορφωθεί πιο εύκολα και να προγραμματιστεί σε μέρη, ακόμα και από διαφορετικά άτομα, στην περίπτωση που το πρόγραμμα είναι πολύ μεγάλο. Τέλος, χωρίζοντας το πρόγραμμα σε μέρη είναι πιο εύκολη η δοκιμή και η αποσφαλμάτωση του. Η δομή του προγράμματος του χρήστη εξαρτάται από το μέγεθος και τις λειτουργίες του.

Οι μορφές προγραμματισμού είναι οι εξής:

- Γραμμικός προγραμματισμός.

Εδώ όλο το κυρίως πρόγραμμα είναι το μπλοκ οργάνωσης OB1. Κάθε τρέχον μονοπάτι είναι σε ξεχωριστό network. Όταν διορθώνουμε και αποσφαλματώνουμε, μπορούμε να αναφέρουμε το κάθε network απευθείας από τον αριθμό του.

- Μερικός προγραμματισμός.

Ο μερικός προγραμματισμός βασίζεται στον γραμμικό προγραμματισμό μόνο που το πρόγραμμα χωρίζεται σε μπλοκ. Οι αιτίες για τον διαχωρισμό του προγράμματος σε μικρότερα μέρη είναι είτε το γεγονός ότι το πρόγραμμα είναι πολύ μεγάλο για το OB1, είτε επειδή θέλουμε να διαβάζεται πιο εύκολα. Τα μπλοκ τότε καλούνται με την σειρά. Μπορούμε επίσης να χωρίσουμε το πρόγραμμα ενός μπλοκ σε άλλα μπλοκ όπως κάναμε με το OB1. Αυτή η μέθοδος μας επιτρέπει να καλούμε συσχετισμένες λειτουργίες της διαδικασίας μέσα από ένα και το αυτό μπλοκ. Το πλεονέκτημα αυτής

της μορφής προγραμματισμού είναι ότι αν και το πρόγραμμα είναι γραμμικό μπορούμε να το αποσφαλματώσουμε σε μέρη (απλά μόνο καλώντας τα μπλοκ).

- Δομημένος προγραμματισμός.

Ο δομημένος προγραμματισμός χρησιμοποιείται όταν το επινοημένο σχέδιο είναι εξαιρετικά ακριβό, όταν θέλουμε να δημιουργήσουμε λειτουργίες προγράμματος και όταν πρέπει να λυθούν περίπλοκα προβλήματα. Μ' αυτήν την μέθοδο χωρίζουμε το πρόγραμμα σε κομμάτια (μπλοκ) με ενσωματωμένες λειτουργίες ή σε μπλοκ που εξυπηρετούν έναν συγκεκριμένο σκοπό λειτουργίας και τα οποία ανταλλάσσουν όσο το δυνατόν λιγότερα σήματα με τα άλλα μπλοκ. Αναθέτοντας σε κάθε κομμάτι μια συγκεκριμένη λειτουργία δημιουργούμε ευανάγνωστα μπλοκ με απλούστερη επικοινωνία με τα άλλα μπλοκ.

Τέλος, η οργάνωση του προγράμματος καθορίζει την σειρά με την οποία η κεντρική μονάδα επεξεργασίας θα εκτελέσει τα μπλοκ που έχουμε δημιουργήσει. Για να οργανώσουμε το πρόγραμμα μας, προγραμματίζουμε τις κλήσεις των μπλοκ με την σειρά που επιθυμούμε. Η σειρά αυτή θα πρέπει να είναι ανάλογη με την σειρά των επιμέρους λειτουργιών της διαδικασίας που θέλουμε να ελέγξουμε.

1.2.10. Γλώσσες προγραμματισμού

Οι ελεγκτές προγραμματίζονται συνήθως σε μια (ή περισσότερες) από τις παρακάτω πια διεθνώς τυποποιημένες μορφές γλωσσών:

- Λίστα εντολών (STATEMENTLIST = STL)
- Σχέδιο επαφών (LADDER DIAGRAM = LAD)
- Λογικό διάγραμμα (FUNCTION BLOCK DIAGRAM = FBD)

Οι μορφές αυτές έχουν τυποποιηθεί κατά DIN και IEC και αποτελούν κατά κάποιο τρόπο τη «γλώσσα» στον προγραμματιζόμενο αυτοματισμό. Μπορούμε να πούμε από την αρχή, ότι οι μορφές σχεδιασμού επαφών (LAD) και λογικού διαγράμματος (FBD) είναι γραφικές μορφές παράστασης, δηλ. το πρόγραμμα «ζωγραφίζεται» πάνω στην οθόνη μιας συσκευής προγραμματισμού.

Το σχέδιο επαφών χρησιμοποιεί λίγο πολύ σύμβολα του κλασσικού συνδεσμολογικού σχεδίου, π.χ. επαφές, πηνία, κτλ. Αντίθετα, το λογικό διάγραμμα χρησιμοποιεί σύμβολα λογικών πυλών, π.χ. πύλη AND, πύλη OR. κλπ.

1.2.11. Σύγκριση μορφών προγραμματισμού

Η «μητρική» γλώσσα κάθε ελεγκτή είναι αναμφίβολα η λίστα εντολών, η οποία έχει και τις μεγαλύτερες δυνατότητες και ευελιξία. Οπωσδήποτε, και οι δύο γραφικές μορφές (σχέδιο επαφών, λογικό διάγραμμα) έχουν το μεγάλο πλεονέκτημα της καλύτερης εποπτείας «με μία ματιά». Στη συνέχεια θα επιχειρήσουμε μια παρουσίαση των σημαντικότερων πλεονεκτημάτων και μειονεκτημάτων.

1.2.11.1. Πλεονεκτήματα λίστας εντολών (STL) σε σχέση με τις γραφικές μορφές (LAD, FBD)

- Έχει τις μεγαλύτερες δυνατότητες, γιατί υπάρχουν εντολές, οι οποίες δεν είναι δυνατόν να παρασταθούν γραφικά, αν και στο κοντινό μέλλον αυτό θα διορθωθεί.
- Γνωρίζουμε με απόλυτη ακρίβεια τη σειρά, με την οποία ο μικροεπεξεργαστής επεξεργάζεται το πρόγραμμα (τη μία εντολή ύστερα από την άλλη).
- Καταλαμβάνει μικρότερο χώρο στη μνήμη για την αποθήκευση του προγράμματος.
- Είναι πολύ προσιτή στην χρήση σε όποιον έχει ασχοληθεί ήδη με προγραμματισμό κάθε είδους.
- Μπορούν να χρησιμοποιηθούν μικροί, φτηνοί, φορητοί προγραμματιστές χειρός (ενώ αντίθετα για τη «σχεδίαση» μιας γραφικής μορφής απαιτείται οθόνη, αν θέλουμε να έχουμε εποπτεία).
- Ο χειρισμός κατά την πληκτρολόγηση του προγράμματος είναι πολύ απλούστερος. Αντίθετα, για την πληκτρολόγηση ενός στοιχείου στις γραφικές

μορφές, π.χ. μιας επαφής, πρέπει ο δείκτης (cursor) στην οθόνη να βρίσκεται στη σωστή θέση.

- Αν σαν βάση για τον προγραμματισμό χρησιμοποιηθεί ένα κλασσικό συνδεσμολογικό σχέδιο με ρελέ ή ένα λογικό διάγραμμα (flow-chart), τότε η «μετάφραση» τους σε λίστα εντολών είναι το ίδιο εύκολη με την «μετάφραση» τους σε σχέδιο επαφών ή λογικό διάγραμμα αντίστοιχα (αν όχι ευκολότερη πολλές φορές).
- Πρέπει να τονιστεί, ότι ένα ηλεκτρολογικό συνδεσμολογικό σχέδιο, πολύ σπάνια μπορεί να προγραμματιστεί όπως είναι, χωρίς μετατροπές, σε σχέδιο επαφών.

1.2.11.2. Μειονεκτήματα λίστας εντολών (STL) σε σχέση με τις γραφικές μορφές (LAD, FBD)

- Ένα πρόγραμμα γραμμένο σε λίστα εντολών δεν έχει την ίδια εποπτεία «με μια ματιά», την οποία έχουν οι γραφικές μορφές.

Με τις δυνατότητες όμως σχολιασμού προγράμματος, που παρέχουν οι σύγχρονες συσκευές προγραμματισμού, το μειονέκτημα αυτό παύει να είναι ιδιαίτερα σημαντικό.

- Η παρακολούθηση του αυτοματισμού σε λειτουργία (πάνω σε μια συσκευή προγραμματισμού οθόνης συνδεδεμένη στον ελεγκτή) είναι απλούστερη και πιο εποπτική, αν το πρόγραμμα είναι γραμμένο σε κάποια από τις δύο γραφικές μορφές.

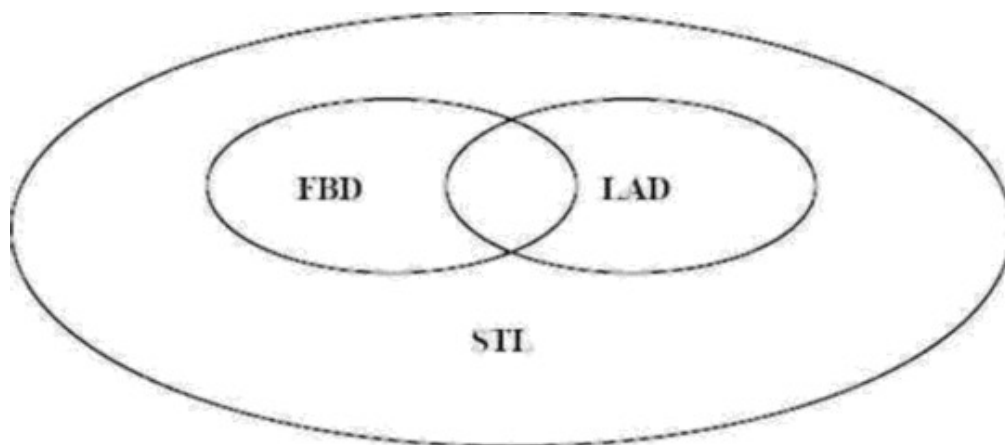
1.2.11.3. Συμπεράσματα.

Καλό είναι οι ελεγκτές να έχουν τη δυνατότητα να προγραμματιστούν και στις τρεις μορφές που προαναφέρθηκαν και να αφήνεται σ' αυτόν που θα φτιάξει το πρόγραμμα η επιλογή της μορφής προγραμματισμού. Θεωρείται αυτονόητο ότι οι τρεις μορφές είναι συμβατές μεταξύ τους, δηλ. σε όποια μορφή κι αν προγραμματίσουμε, έχουμε τη δυνατότητα να πάρουμε το πρόγραμμα και στις άλλες δύο, ζητώντας το από τη συσκευή.

Η χρήση περισσότερων από μία μορφή παράστασης ενός προγράμματος είναι πολλές φορές επιθυμητή και για άλλους λόγους:

Π.χ. σ' ένα μεγάλο εργοστάσιο, αυτός που θα φτιάξει το πρόγραμμα μπορεί να επιλέξει π.χ. τη λίστα εντολών, αλλά η ηλεκτρολογική συντήρηση πιθανόν να προτιμάει στο αρχείο της την παράσταση σχεδίου επαφών, για την ανεύρεση βλαβών.

Όσοι έχουν εμπειρία στον τομέα του αυτοματισμού επιλέγουν συνήθως τη μορφή προγραμματισμού που ταιριάζει καλύτερα στην εμπειρία τους, λαμβάνοντας υπ' όψη και τα πλεονεκτήματα -μειονεκτήματα που προαναφέραμε.



Εικ. 1.2 Συμβατότητα μεταξύ των γλωσσών.

1.2.12. Θέση σε λειτουργία Προγραμματιζόμενου λογικού ελεγκτή.

Η θέση σε λειτουργία χωρίζεται σε δύο τελείως ξεχωριστά μέρη:

- Θέση σε λειτουργία και έλεγχος όλων των μονάδων του ελεγκτή (HARDWARE).

Μπορεί να γίνει και από μη ειδικευμένο προσωπικό και προηγείται πάντοτε οποιουδήποτε ελέγχου.

- Θέση σε λειτουργία του προγράμματος (SOFTWARE). Γίνεται οπωσδήποτε από άτομο με γνώσεις προγραμματισμού.

1.2.12.1. Θέση σε λειτουργία και έλεγχος των μονάδων ενός ελεγκτή (HARDWARE)

Συνήθως, ο ελεγκτής παραδίνεται επί τόπου του έργου, τοποθετημένος και συρματωμένος μέσα σ' έναν πίνακα. Όλες οι εισοδοί και οι εξοδοί, που πρόκειται να συνδεθούν με την εξωτερική εγκατάσταση, πρέπει να είναι συρματωμένες σε κλέμες. Ο πίνακας περιέχει, συνήθως, και τα τροφοδοτικά για τις τάσεις εισόδων - εξόδων, με τους αντίστοιχους μικροαυτόματους κατανομής προς τις μονάδες.

Η θέση σε λειτουργία των μονάδων ενός ελεγκτή διακρίνεται σε 4 διαδοχικά στάδια:

- Προετοιμασία.
- Έλεγχος μονάδας τροφοδοσία και CPU.
- Έλεγχος των εισόδων και της συρμάτωσης τους.
- Έλεγχος των εξόδων και της συρμάτωσης τους.

Προετοιμασία

1. Βγάζουμε τις ασφάλειες από όλα τα κυκλώματα ισχύος (π.χ. κινητήρες, ώστε αν τυχαία οπλίσει ένα ρελέ να μην πάρει μπρος ο αντίστοιχος κινητήρας).
2. Βγάζουμε εκτός όλους τους μικροαυτόματους, οι οποίοι τροφοδοτούν τον ελεγκτή και τις εισόδους / εξόδους του .
3. Ξεσυρματώνουμε και τραβάμε έξω τις πρίζες καλωδίων από όλες τις μονάδες εισόδων - εξόδων.
4. Ελέγχουμε, βάσει του σχεδίου, αν σε κάθε θέση του πλαισίου υπάρχει ο σωστός τύπος μονάδας με τις σωστές διευθύνσεις.

Έλεγχος μονάδας τροφοδοσίας (PS) και κεντρικής μονάδος (CPU)

1. Ελέγχουμε αν η τάση του δικτύου έχει σωστή τιμή, αντίστοιχη με την ονομαστική τάση της μονάδας τροφοδοσίας. Αν υπάρχει επιλογικός διακόπτης τάσης, εξετάζουμε αν είναι στη σωστή θέση . Βάζουμε εντός τον μικροαυτόματο παροχής τροφοδοσίας και ελέγχουμε, αν τα LED ένδειξης των εσωτερικών τάσεων του ελεγκτή πάνω στη μονάδα τροφοδοσίας ανάβουν όλα.

2. Για την κεντρική μονάδα μπορούμε να κάνουμε ένα απλό τεστ, το οποίο λέει χονδρικά, αν η μονάδα είναι εντάξει.

Προσοχή! Το τεστ που ακολουθεί μπορεί να εφαρμοστεί μόνο αν είμαστε σίγουροι ότι δεν υπάρχει πρόγραμμα στη μνήμη. Σε αντίθετη περίπτωση το πρόγραμμα χάνεται.

Τεστ: Συνδέουμε μια συσκευή προγραμματισμού και προσπαθούμε να σβήσουμε όλη τη μνήμη, με τους χειρισμούς που λέει ο κατασκευαστής. Σε πολλούς τύπους ελεγκτών το σβήσιμο αυτό είναι δυνατό και χωρίς συσκευή προγραμματισμού, απλά και μόνο με χειρισμούς πάνω στους διακόπτες της μονάδας CPU.

Στη συνέχεια, πηγαίνουμε το διακόπτη της κεντρικής μονάδας στη θέση «RUN». Αν ανάψει το αντίστοιχο LED, σημαίνει ότι η μονάδα μπαίνει σε κύκλο λειτουργίας, άρα με μεγάλη πιθανότητα είναι εντάξει.

Έλεγχος των εισόδων και της συρμάτωσης τους

1. Βάζουμε εντός τους μικροαυτόματους τροφοδοσίας των μονάδων εισόδων.

2. Διεγείροντας έναν - έναν τερματικό, μπουτόν, διακόπτη κ.λπ. (π.χ. K1), ελέγχουμε με πολύμετρο, αν φτάνει η σωστή τάση στη σωστή κλέμα της τραβηγμένης πρίζας καλωδίων. Ελέγχουμε επίσης, αν φτάνει η τάση L- βάσει του σχεδίου. Έτσι βεβαιωνόμαστε για τον Έλεγχο των εισόδων και της συρμάτωσης τους.

3. Βάζουμε τις πρίζες καλωδίων των μονάδων εισόδων στη θέση τους και συνδέουμε μια συσκευή προγραμματισμού στην κεντρική μονάδα. Επιλέγουμε τη λειτουργία «Έλεγχος κατάστασης στοιχείου».

Γεφυρώνουμε διαδοχικά το μπαράκι κατανομής δυναμικού προς την εγκατάσταση με μία - μία κλέμα εισόδου πάνω στις πρίζες καλωδίων. Παρατηρούμε αν ανάβει το αντίστοιχο LED εισόδου και αν το αντίστοιχο σήμα της εισόδου στη συσκευή προγραμματισμού γίνεται «1». Αν δεν υπάρχει αντιστοιχία: «Τάση στην κλέμα» - «LED αναμμένο» = Σήμα εισόδου «1», η υπό έλεγχο είσοδος δεν λειτουργεί σωστά. Έτσι βεβαιωνόμαστε για τη σωστή λειτουργία των μονάδων εισόδου.

Σημείωση 1

Μελετώντας το ηλεκτρολογικό σχέδιο του συστήματος διαπιστώνουμε, ότι η κατάσταση «LED εισόδου αναμμένο» δεν σημαίνει τίποτε άλλο, παρά το ότι η τάση φτάνει στην αντίστοιχη κλέμα της μονάδας εισόδων.

Ότι η τάση αυτή μετατρέπεται σωστά και «περνάει» στο εσωτερικό του συστήματος ελέγχεται μόνο με τη συσκευή προγραμματισμού, αν επιλέξουμε τη λειτουργία που αναφέραμε.

Σημείωση 2

Από τα προηγούμενα διαπιστώσαμε ότι χρειάστηκε πρακτικά να γίνει η ίδια δουλειά δύο φορές, για τον έλεγχο της συρμάτωσης και της σωστής λειτουργίας εισόδων.

Πολλές φορές στην πράξη αυτό γίνεται σε μία φάση χωρίς τράβηγμα των πριζών καλωδίων και χωρίς χρήση του πολυμέτρου.

Σε περίπτωση που δεν υπάρχει, λοιπόν, διαθέσιμος χρόνος, κάνουμε μόνο έλεγχο της συρμάτωσης, όπως περιγράψαμε. Τυχόν λανθασμένη λειτουργία κάποιας εισόδου θα ανακαλυφθεί κατά τον έλεγχο του προγράμματος.

Έλεγχος των εξόδων και της συρμάτωσής τους

1. Βάζουμε «εντός» όλους τους μικροαυτόματους τροφοδοσίας των μονάδων εξόδων.
2. Ελέγχουμε, βάσει του σχεδίου, αν φτάνει η σωστή τάση τροφοδοσίας στην κλέμα τροφοδοσίας των μονάδων εξόδων, π.χ. κλέμα 1 της τραβηγμένης πρίζας καλωδίων.

Ελέγχουμε επίσης, αν φτάνει και η τάση L- (0V) στην αντίστοιχη κλέμα, π.χ. κλέμα 11.

3. Βάζουμε τις πρίζες καλωδίων των μονάδων εξόδων στη θέση τους και συνδέουμε τη συσκευή προγραμματισμού στην κεντρική μονάδα.

4. Χρησιμοποιώντας τη λειτουργία Εξαναγκασμός σήματος (Force), εξαναγκάζουμε διαδοχικά σήμα «1» σε κάθε έξοδο. Πρέπει να ανάβει το αντίστοιχο LED εξόδου και να διεγείρεται το αντίστοιχο στοιχείο που δέχεται την εντολή (π.χ. ρελέ C1).

Έτσι βεβαιωνόμαστε ταυτόχρονα και για τη σωστή συρμάτωση και για τη σωστή λειτουργία των εξόδων.

1.2.13. Σύγκριση μεταξύ κλασσικού αυτοματισμού και μοντέρνου (PLC)

Σε μια αυτοματοποιημένη παραγωγική διαδικασία (διακρίνουμε τα εξής βασικά μέρη, αν ο αυτοματισμός έχει γίνει με το κλασσικό τρόπο.

1. Ηλεκτρολογικός εξοπλισμός της διαδικασίας (τερματικοί διακόπτες, αναλογικά αισθητήρια, κινητήρες, βαλβίδες).
2. Πίνακας ισχύος για την τροφοδοσία των κινητήρων και των βαλβίδων.
3. Πίνακας αυτοματισμού με βοηθητικά ρελέ - χρονικά.
4. Πίνακας χειρισμών (μπουτόν, διακόπτες).
5. Μιμικό διάγραμμα (απεικόνιση, λυχνίες, αναλογικά όργανα).

Η «καρδιά» όλων των ανωτέρω είναι ο πίνακας αυτοματισμού ο οποίος:

- Δέχεται εντολές (σήματα) από τα αισθητήρια της εγκατάστασης, από το χειριστήριο και από τον πίνακα ισχύος (βοηθητικές επαφές των ρελέ ισχύος).
- Επεξεργάζεται τις εντολές αυτές με βάση κάποιο συνδεσμολογικό σχέδιο (λειτουργία αυτοματισμού).

- Στέλνει εντολές ενεργοποίησης προς τον πίνακα ισχύος, για την ζεύξη των κινητήρων και των βαλβίδων, και προς το μιμικό διάγραμμα, για ένδειξη στις λυχνίες και τα αναλογικά όργανα.

Το πρώτο βήμα βελτίωσης της «ποιότητας» είναι η αντικατάσταση του κλασικού πίνακα αυτοματισμού με τον προγραμματιζόμενο λογικό ελεγκτή. Η βασική διαφοροποίηση είναι ότι η λειτουργία του αυτοματισμού δεν «συρματώνεται» αλλά «προγραμματίζεται» με κάποια συσκευή προγραμματισμού, στην μνήμη της συσκευής αυτοματισμού, με όλα τα σχετικά πλεονεκτήματα, όπως:

- Αξιοπιστία λόγω μη κινούμενων μερών.
- Περισσότερες δυνατότητες, μη πραγματοποιήσιμες με ρελέ.
- Ευκολία αλλαγών χωρίς αλλαγή της συρμάτωσης.
- Δεν υπάρχουν «ανενημέρωτα» σχέδια, γιατί το τρέχον πρόγραμμα είναι μέσα στη μνήμη και μπορεί να διαβαστεί ή να τυπωθεί ανά πάσα στιγμή.
- Στη μελέτη δεν υπάρχει το πρόβλημα αν επαρκούν οι επαφές των ρελέ, των χρονικών ή των εξωτερικών τερματικών.
- Διευκόλυνση στον εντοπισμό εξωτερικών βλαβών λόγω υπάρξεως LED σε κάθε είσοδο, έξοδο.

Το επόμενο βήμα βελτίωσης, το οποίο μελετάται στη συνέχεια, είναι η αντικατάσταση του χειριστηρίου από ένα πληκτρολόγιο και του μιμικού διαγράμματος από έναν ηλεκτρονικό υπολογιστή. Έτσι η όλη διαδικασία της παρακολούθησης και του χειρισμού γίνεται αυτόνομα χωρίς «επιβάρυνση» της CPU του ελεγκτή (ταχύτερη επεξεργασία).

Ένα ακόμα βήμα βελτίωσης έγκειται στη προσθήκη κάποιων συσκευών εκτύπωσης, με σκοπό την δυνατότητα καταγραφής χρήσιμων στοιχείων που σχετίζονται άμεσα με την διαδικασία. Συνήθως απαιτείται η καταγραφή του χρόνου λειτουργίας κάποιων κινητήρων, φωτιστικών, κ.α. Παράλληλα, είναι πολύ θετικό να γίνεται ενημέρωση μέσω του εκτυπωτή για την ύπαρξη βλάβης σε κάποιο σημείο της διαδικασίας ή την

αποκατάσταση αυτής, με ταυτόχρονη εμφάνιση ώρας και ημερομηνίας για εκτίμηση της συχνότητας με την οποία επαναλαμβάνονται τα διάφορα σφάλματα.

Στις μέρες μας εξαιτίας της περιπλοκότητας των, βιομηχανικών και μη, διεργασιών που καλείται να ελέγξει ένας προγραμματιζόμενος λογικό ελεγκτής, έχουν αναπτυχθεί σε μεγάλο βαθμό τα Συστήματα Εποπτικού Ελέγχου.

1.3. Προγραμματισμός του PLC που εγκαταστήσαμε.

Τα PLC διαθέτουν πληθώρα χρησιμων εργαλείων εκ των οποίο πολύ σημαντικό και χρήσιμο είναι η δυνατότητα που παρέχει για on-line παρακολούθηση του προγράμματος όντας συνδεδεμένο με το PLC αλλά και η δυνατότητα πραγματοποίησης αλλαγών σε on-line κατάσταση.

Η βασική φιλοσοφία χρήσης του προγράμματος συνίσταται στα τρία παρακάτω βήματα:

1. Δημιουργία-γράψιμο του προγράμματος
2. Μεταφορά του προγράμματος στην μνήμη του PLC (download)
3. Τρέξιμο του προγράμματος, το PLC σε κατάσταση Run Mode.

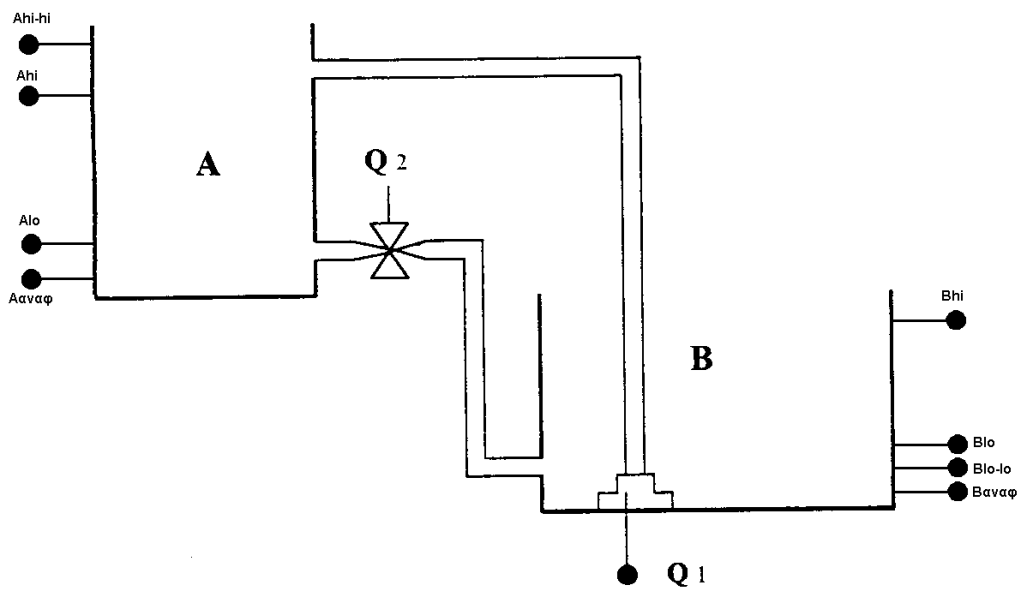
Ο κώδικας του προγραμματισμού που πραγματοποιήσαμε παρουσιάζεται σε επόμενο κεφάλαιο.

2. Περιγραφή της εφαρμογής

2.1. Εξήγηση λειτουργίας και κύκλος εργασίας

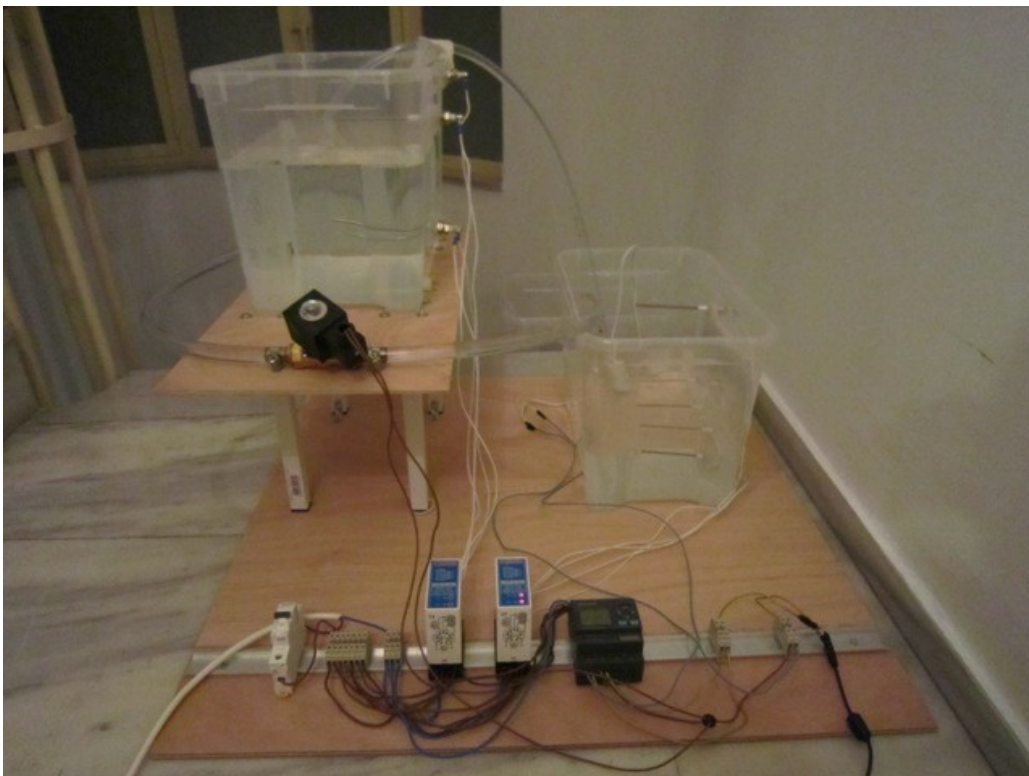
Πριν να εξηγηθεί το πως λειτουργεί το σύστημα δίνονται όλα τα σήματα που χρησιμοποιούνται σχεδιασμένα σ' ένα σκαρίφημα που απεικονίζει τις δεξαμενές και τα σημεία που τοποθετούνται τα αισθητήρια. Στην Α δεξαμενή (πάνω) τοποθετούνται τέσσερα αισθητήρια στάθμης εκ των οποίων το A_H και A_L είναι τα βασικά αισθητήρια (Hi/Low) μεταξύ των οποίων κινείται το υγρό ενώ υπάρχουν και άλλα δυο αισθητήρια, το ένα αναφοράς και το άλλο εφεδρικό (Hi-Hi) για να μην υπερχειλίσει η δεξαμενή Α σε περίπτωση που δε λειτουργήσουν τα δυο πρώτα αισθητήρια. Για τον έλεγχο της στάθμης της δεξαμενής Β υπάρχουν και πάλι τέσσερα αισθητήρια, εκ των οποίων το ένα είναι αναφοράς, τα άλλα δυο είναι για άνω και κάτω έλεγχο (B_H , B_L), ενώ το τέταρτο είναι εφεδρικό και είναι τοποθετημένο στο κάτω μέρος της δεξαμενής Β (Low-Low) για να μας προστατεύσει την αντλία σε περίπτωση που συνεχίσει να αδειάζει η δεξαμενή Β ακόμα και όταν το νερό έχει πέσει κάτω από το αισθητήριο B_L .

Ας υποθέσουμε ότι η δεξαμενή Β είναι γεμάτη χωρίς η στάθμη του να έχει προσεγγίσει το αισθητήριο B_H . Η δεξαμενή Α είναι εντελώς άδεια. Ενεργοποιούμε τον ελεγκτή με βάση το πρόγραμμα που δίνεται σε επόμενη παράγραφο, οπότε η επαφή Q_1 κλείνει και ενεργοποιεί την αντλία και αρχίζει το γέμισμα της πάνω δεξαμενής. Μόλις η στάθμη φτάσει στο A_H αλλά και το B_L απενεργοποιείται η αντλία (ανοίγει η επαφή Q_1) και με το κλείσιμο της επαφής Q_2 ανοίγει η ηλεκτροβαλβίδα και αδειάζει η δεξαμενή Α μέχρι τη στιγμή που το νερό πέσει κάτω από το αισθητήριο A_L αλλά και το B_H , οπότε και κλείνει η ηλεκτροβαλβίδα (ανοίγει η επαφή Q_2) και αρχίζει και πάλι να λειτουργεί η αντλία. Αυτός ο κύκλος επαναλαμβάνεται συνέχεια μέχρι τη στιγμή που θα σταματήσουμε το πρόγραμμα. Το αισθητήριο ασφαλείας A_{H-H} ενεργοποιείται στην περίπτωση που δεν λειτουργήσει το A_H ενώ το αισθητήριο B_{L-L} ενεργοποιείται στην περίπτωση που δε λειτουργήσει το B_L για να μην κινδυνεύσει η αντλία να μείνει χωρίς νερό.



Εικ. 2.1 : Σχηματική αναπαράσταση της κατασκευής ελέγχου στάθμης

Ακολουθεί αναλυτική περιγραφή των εξαρτημάτων που χρησιμοποιούνται στην κατασκευή.



Εικ. 2.2 : Σύστημα ελέγχου στάθμης υγρού με χρήση δεξαμενών

2.2.Αντλία

Γενικά μια αντλία αποτελείται από ένα ή περισσότερους δρομείς, τοποθετημένους σ' έναν άξονα που παίρνει κίνηση από κινητήρα και στρέφεται πάντοτε κατά την ίδια φορά μέσα σ' ένα περίβλημα. Το περίβλημα έχει ανοίγματα εισόδου και εξόδου του υγρού. Επάνω στον περιστρεφόμενο δρομέα είναι τοποθετημένα πτερύγια κατάλληλου σχήματος. Ο δρομέας μαζί με τα πτερύγια ονομάζονται πτερωτή (IMPELLER). Καθώς η πτερωτή περιστρέφεται, το υγρό αποκτά γωνιακή επιτάχυνση οπότε δημιουργείται φυγόκεντρη δύναμη που μεταφέρει το υγρό προς την περιφέρεια του περιβλήματος για να απομακρυνθεί από το άνοιγμα εξόδου.

Επειδή το υγρό απομακρύνεται από το κέντρο προς την περιφέρεια, η πίεση στο κέντρο ελαττώνεται και νέα ποσότητα υγρού κινείται προς το σημείο χαμηλής πίεσεως με αποτέλεσμα τη δημιουργία σταθερής ροής.

Ανάλογα με το σχήμα των πτερυγίων και τον τρόπο κινήσεως του υγρού οι δυναμικές αντλίες διακρίνονται ως εξής :

- Φυγόκεντρες αντλίες οι οποίες έχουν διάταξη πτερυγίων τέτοια ώστε το υγρό που εισέρχεται στο περίβλημα αξονικά, να κινείται και να εξέρχεται κάθετα προς τον άξονα, δηλαδή ακτινικά.
- Μικτής ροής, όπου τα πτερύγια είναι τοποθετημένα υπό γωνία ως προς τον άξονα της περιστροφής. Το υγρό εισέρχεται αξονικά και κινείται συγχρόνως αξονικά και ακτινικά.
- Αξονικής ροής όπου η πτερωτή έχει σχήμα έλικας, το υγρό εισέρχεται κινείται και εξέρχεται αξονικώς. Η αύξηση της πίεσης προέρχεται από την ανυψωτική δράση της έλικας που είναι πάντοτε βυθισμένη στο νερό.
- Στην κατηγορία αυτή ανήκει και η αντλία που χρησιμοποιούμε στην κατασκευή μας. Επειδή η αντλία είναι βυθισμένη στο νερό λέγεται και “βατραχάκι”. Δέχεται τροφοδοσία 12 V DC και ο σωλήνας εξόδου έχει διατομή 3/8”.
- Στροβιλαντλίες στις οποίες τα πτερύγια έχουν ειδική τοποθέτηση.



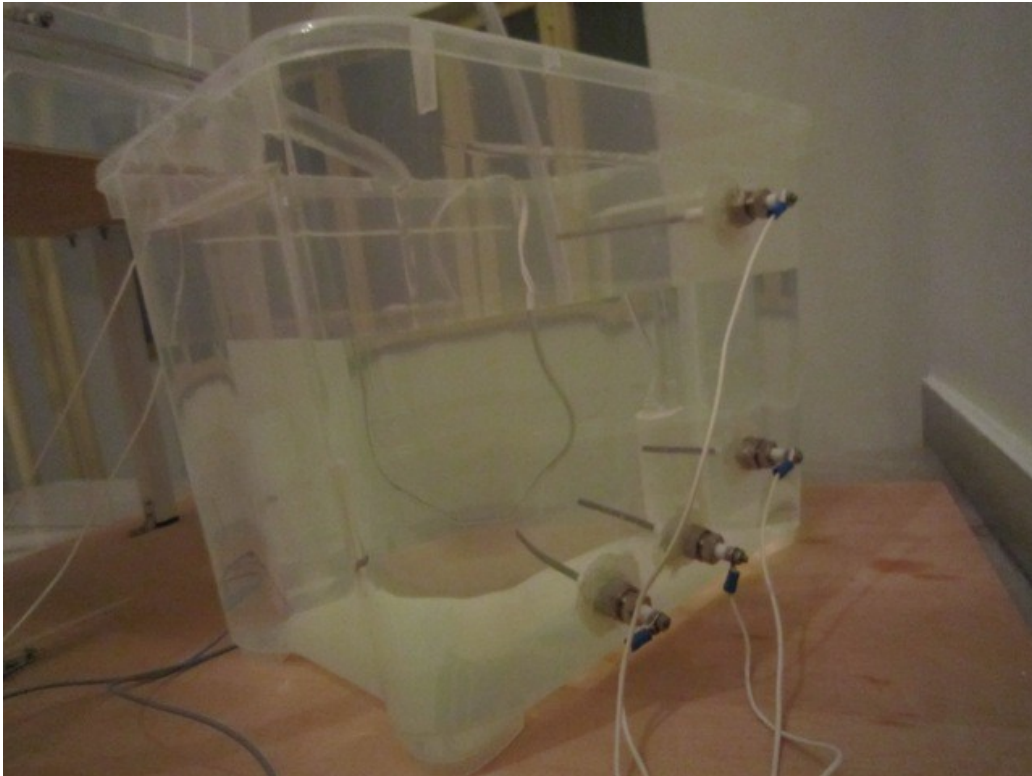
Εικ. 2.3 : Αντλία

2.3.Δεξαμενές-Σωλήνωση-Καλωδίωση

Οι δεξαμενές που χρησιμοποιήθηκαν είναι ίδιες και οι διαστάσεις τους είναι (ΥxΠxΜ) 25 x 21 x 31 cm.

Η σωλήνωση που χρησιμοποιήθηκε είναι με βάση την διατομή του αγωγού της ηλεκτροβαλβίδας δηλαδή 1/2 in. Αντίστοιχα στη σωλήνωση της αντλίας η διατομή είναι 3/8 in.

Για την καλωδίωση χρησιμοποιήσαμε τριών ειδών διαφορετικά καλώδια. Το πρώτο έχει διατομή 0,5 mm² και το χρησιμοποιήσαμε για τη σύνδεση των αισθητηρίων με τον επιτηρητή στάθμης. Το δεύτερο έχει διατομή 1 mm² και με αυτό κάναμε όλες τις συνδέσεις στην έξοδο του PLC καθώς και την αντλία και την ηλεκτροβαλβίδα. Το τρίτο καλώδιο έχει διατομή 1,5 mm² και με αυτό συνδέσαμε την τροφοδοσία μας με τους επιτηρητές τάσης και αυτούς αντίστοιχα με τις εισόδους του PLC.



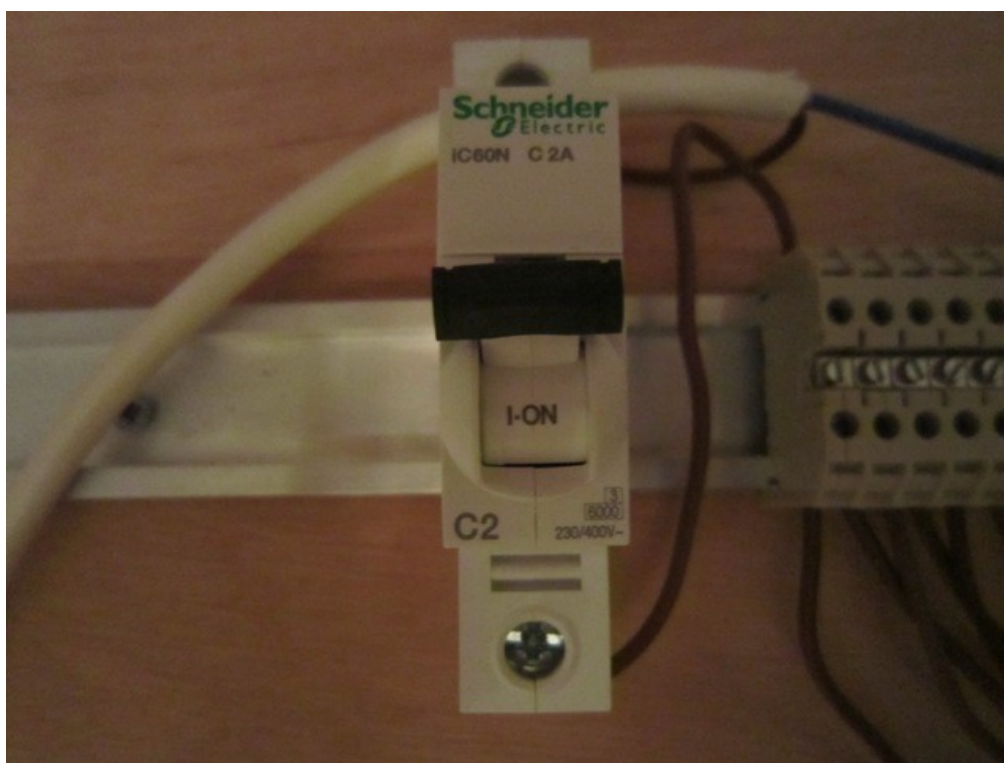
Εικ. 2.4 : Δεξαμενές



Εικ. 2.5 : Σωλήνωση ηλεκτροβαλβίδας

2.4.Τροφοδοσίες

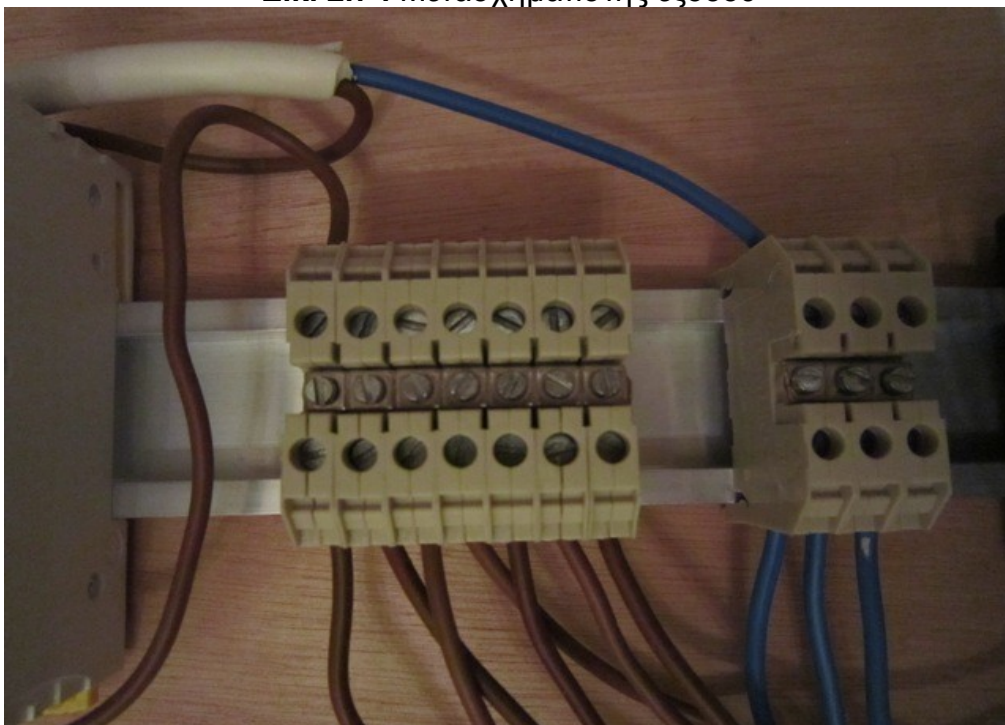
Το PLC καθώς και οι επιτηρητές μας δέχονται τροφοδοσία 230 V AC. Για να προστατεύσουμε το PLC από υπερφορτίσεις έχουμε χρησιμοποιήσει έναν μικροαυτόματο διακόπτη 2A της εταιρίας Schneider Electric. Επιπλέον οι εισοδοί του PLC για να λειτουργήσουν σωστά πρέπει να έχουν και αυτές 230 V AC. Αντίθετα η αντλία και η ηλεκτροβαλβίδα λειτουργούν με 12 V DC. Για το λόγο αυτό έχουμε χρησιμοποιήσει ένα μετασχηματιστή 230 V AC/12 V DC. Οι έξοδοι του PLC λειτουργούν απλώς ως επαφές οπότε δεν υπάρχει πρόβλημα για το τι τάση θα διαρρέει τις επαφές αυτές. Εμείς τις έχουμε συνδέσει με τάση 12 V DC μιας και αυτή είναι η τάση που χρειάζεται η αντλία και η ηλεκτροβαλβίδα που συνδέονται στις εξόδους. Επιπλέον χρησιμοποιήσαμε κλέμμες ράγας 3,5 cm της SIEMENS τις οποίες βραχυκυκλώσαμε με μπάρες παράλληλης σύνδεσης.



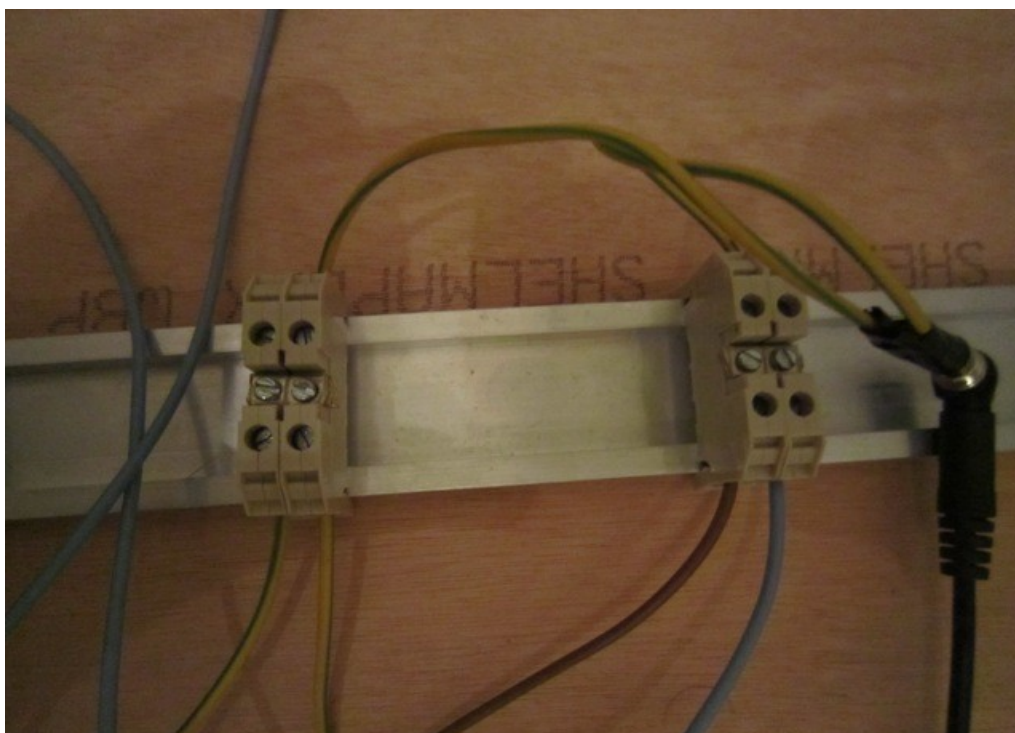
Εικ. 2.6 : Ασφάλεια Schneider 2A



Εικ. 2.7 : Μετασχηματιστής εξόδου



Εικ. 2.8 : Βραχυκυκλωμένες κλήμες εισόδου



Εικ. 2.9 : Βραχυκυκλωμένες κλέμες εξόδου

2.5. Έλεγχος στάθμης δεξαμενών με χρήση επιτηρητή

Για τον έλεγχο της στάθμης του υγρού μέσα στις δεξαμενές χρησιμοποιήθηκαν δυο επιτηρητές στάθμης της εταιρίας RINGEL ELECTRONICS και πιο συγκεκριμένα το μοντέλο RT 822, στον κάθε ένα από τους οποίους έχουν συνδεθεί τέσσερα ηλεκτρόδια στάθμης με υλικό κατασκευής INOX μήκους 10 cm. Πιο συγκεκριμένα ο επιτηρητής μας κάνει διπλό έλεγχο δηλαδή έχουμε ανεξάρτητη επιτήρηση ελάχιστης και μέγιστης στάθμης. Χρησιμοποιείται κυρίως σε αντλιακά συγκροτήματα, ατμολέβητες και στη βιομηχανία τροφίμων.

2.5.1. Περιγραφή Λειτουργίας

Η θέση του ηλεκτροδίου που συνδέεται στο pin 5 καθορίζει το μέγιστο όριο της στάθμης του υγρού, η θέση του ηλεκτροδίου που συνδέεται στο pin 6 την ελάχιστη και του ηλεκτροδίου 8 την στάθμη αναφοράς. Αν το υλικό του δοχείου είναι μεταλλικό (αγώγιμο) τότε αυτό μπορεί να χρησιμοποιηθεί ως αναφορά συνδέοντας το με την γείωση (pin 8) . Στο ηλεκτρόδιο αναφοράς εφαρμόζεται εναλλασσόμενη τάση 24Vac, 50Hz η οποία όταν η στάθμη του νερού φτάσει να καλύψει το ηλεκτρόδιο που

συνδέεται στο pin 5 κλείνει κύκλωμα με συνέπεια η επαφή 1-3 να κλείσει ενώ η 1-4 ανοίξει. Όσο η στάθμη παραμένει πάνω από το όριο που ορίζει το ηλεκτρόδιο που συνδέεται στο pin 5 η κατάσταση των εξόδων δεν αλλάζει. Όταν η στάθμη πέσει κάτω από το όριο που ορίζει το ηλεκτρόδιο που συνδέεται στο pin 6 η επαφή 1-3 ανοίγει και η 1-4 κλείνει.

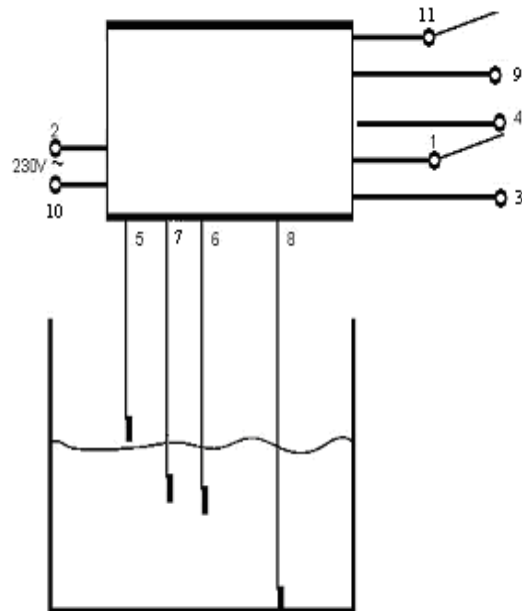
2.5.2.Τεχνικά Χαρακτηριστικά

Υστέρηση	Τυπική τιμή 10%
Καθυστέρηση ενεργοποίησης	-----
Ένδειξη σωστής διαδοχής	Πράσινο LED
Περιβάλλον Προστασία Θερμοκρασία λειτουργίας Θερμοκρασία αποθήκευσης	IP20 -20° to +50°C -50° to +85°C
Βάρος	200γραμ.(gr)
Σήμανση	CE, VDE

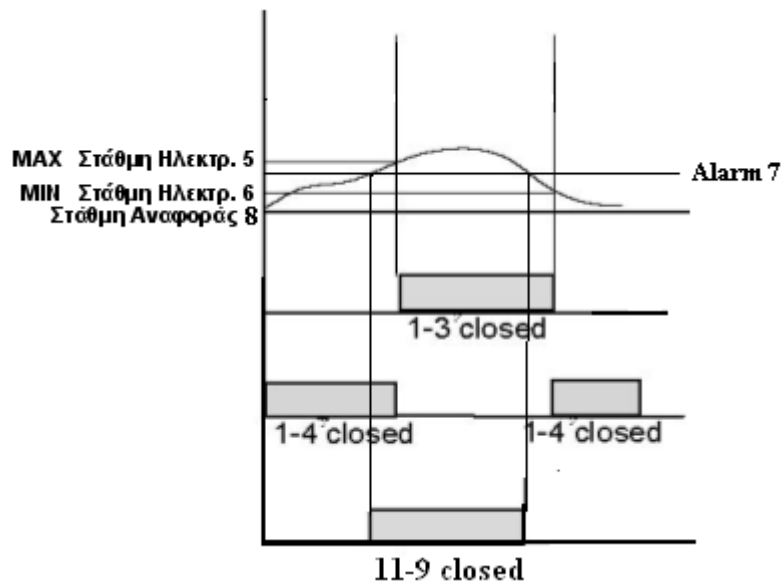
Έξοδος	Ρελέ SPDP
Τάση μόνωσης	250 Vac rms
Ρεύμα επαφών Ωμικό φορτίο AC1 DC1 Επαγ. Φορτίο AC15	10A/250V ac 5A/24V dc 2A/250 V ac 3A / 24V ac
Διάρκεια ζωής (μηχανική)	2x10 ⁶ ενεργ.
Διάρκεια ζωής (ηλεκτρική)	10 ⁵ ενεργ.

Εικ. 2.10 : Πίνακας τεχνικών χαρακτηριστικών επιτηρητή

2.5.3. Διάγραμμα Λειτουργίας



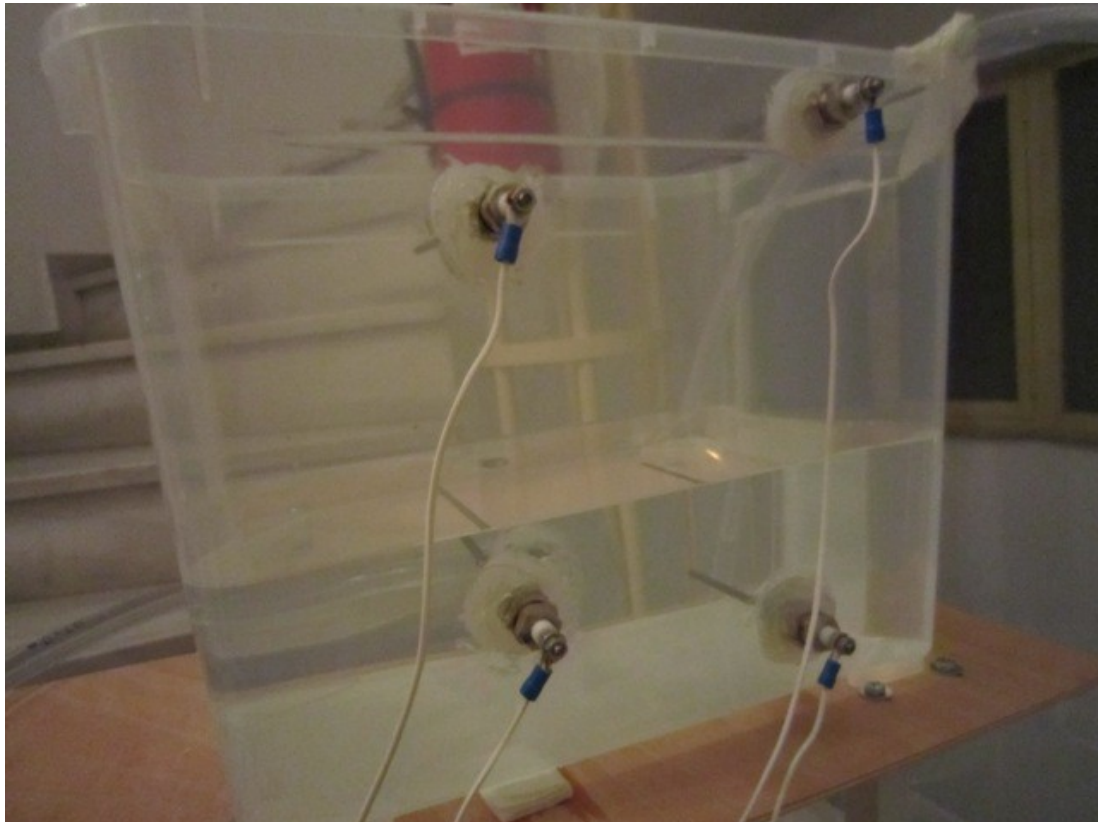
Εικ. 2.11 : Απεικόνιση αισθητηρίων – επαφών



Εικ. 2.12 : Διάγραμμα λειτουργίας



Εικ. 2.13 : Επιτηρητές στάθμης



Εικ. 2.14 : Αισθητήρια

2.6. Ηλεκτροβαλβίδα

2.6.1. Γενικά

Μια ηλεκτρομαγνητική βαλβίδα είναι μια ηλεκτρομηχανική βαλβίδα για χρήση σε υγρό ή αέριο. Η βαλβίδα αυτή ελέγχεται από ένα ηλεκτρικό ρεύμα μέσα από ένα σωληνοειδές πηνίο. Ηλεκτρομαγνητικές βαλβίδες μπορούν να έχουν δύο ή περισσότερους εισόδους- εξόδους. Στην περίπτωση των δύο εισόδων-εξόδων η μια είναι είσοδος και η άλλη έξοδος. Στην περίπτωση των τριών εισόδων-εξόδων της βαλβίδας η εισροή γίνεται από μία είσοδο και η εκροή από τις δύο εξόδους. Πολλαπλές ηλεκτρομαγνητικές βαλβίδες μπορούν να τοποθετηθούν μαζί σε ένα συλλέκτη.

Ηλεκτρομαγνητικές βαλβίδες είναι τα πιο συχνά χρησιμοποιούμενα στοιχεία ελέγχου ρευστών και χρησιμοποιούνται σε πολλές εφαρμογές. Τα καθήκοντά τους είναι να σταματούν ή να επιτρέπουν τη ροή υγρών. Τα σωληνοειδή προσφέρουν γρήγορη και ασφαλή μεταγωγή, υψηλή αξιοπιστία, μεγάλη διάρκεια ζωής, καλό μέσο συμβατότητας των υλικών που χρησιμοποιούνται, έχουν χαμηλή ισχύς ελέγχου και συμπαγή σχεδιασμό.

Οι ηλεκτροβαλβίδες αποτελούν νευραλγικό κομμάτι του αυτόματου αρδευτικού δικτύου και τα τεχνικά χαρακτηριστικά τους ποικίλουν ανάλογα με τις ανάγκες που καλούνται να εξυπηρετήσουν. Σαν μηχανισμοί υφίστανται διαρκώς καταπόνηση τόσο κατά τη φάση λειτουργίας τους, όσο και όταν διαδραματίζουν το ρόλο του διακόπτη παροχής ύδατος. Είναι συνεχώς εκτεθειμένες στις επιδράσεις του περιβάλλοντος χώρου και οποιαδήποτε δυσλειτουργία τους μπορεί να αποβεί έως μοιραία για την επιβίωση των φυτών που διακοσμούν τον κήπο μας. Για αυτό καλό θα ήταν να γνωρίζουμε κάποια απλά πράγματα για αυτές αφού στην ουσία διαχειρίζονται την προσφορά στα φυτά του σημαντικότερου αγαθού, δηλαδή του νερού.

2.6.2. Αρχή λειτουργίας

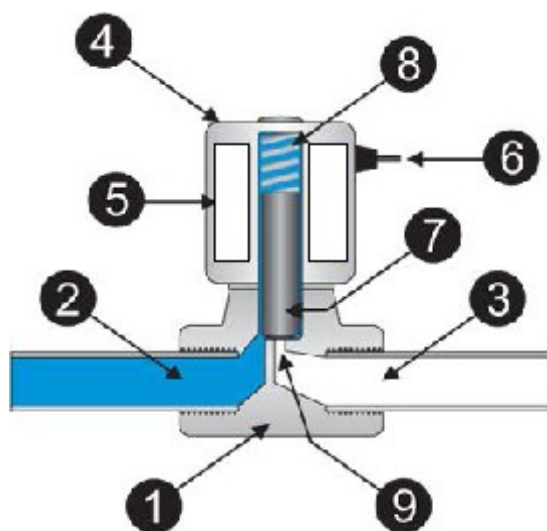
Μια ηλεκτρομαγνητική βαλβίδα έχει δύο κύρια μέρη: το σωληνοειδές και τη βαλβίδα. Το σωληνοειδές μετατρέπει ηλεκτρική ενέργεια σε μηχανική ενέργεια το οποίο, με τη

σειρά του, ανοίγει ή κλείνει τη βαλβίδα μηχανικά.

Το παρακάτω σχήμα απεικονίζει τα βασικά συστατικά μιας ηλεκτρομαγνητικής βαλβίδας. Η βαλβίδα που φαίνεται στην εικόνα είναι κλειστή σε ηρεμία (normally close), άμεσης ενεργοποίησης βαλβίδα. Αυτός ο τύπος ηλεκτρομαγνητικής βαλβίδας έχει τον πιο απλό και κατανοητό τρόπο λειτουργίας.

Η ηλεκτρομαγνητική βαλβίδα αποτελείται από τα μέρη:

1. Σώμα βαλβίδας
2. Στόμιο εισόδου
3. Στόμιο εξόδου
4. Σωληνοειδές(πηνίο)
5. Τυλίγματα
6. Καλώδια τροφοδοσίας
7. Έμβολο
8. Ελατήριο
9. Στόμιο



Εικ. 2.15 : Τεχνικά μέρη ηλεκτροβαλβίδας

Τα κύρια μέρη της παραπάνω ηλεκτροβαλβίδας είναι το σύστημα του πηνίου και του εμβόλου:

- Το σύστημα του πηνίου είναι αυτό που ενεργοποιείται από τον ηλεκτρονικό προγραμματιστή προκειμένου να ασφαλίσει ή να απασφαλίσει την ηλεκτροβαλβίδα. Η λειτουργία του βασίζεται σε απλούς φυσικούς νόμους του ηλεκτρισμού, αλλά δεν είναι καθόλου απλό να κατασκευαστεί ένα αξιόπιστο και ανθεκτικό πηνίο ώστε να ανταπεξέρχεται στο χρόνο και τις φθορές. Τα πηνία είναι η αχίλλειος πτέρνα των ηλεκτροβαλβίδων και η ποιότητα της κατασκευής τους χαρακτηρίζει καίρια την ποιότητα αυτών των βαλβίδων. Έχει τεράστια σημασία το σύστημα του πηνίου να είναι ένα συμπαγές τμήμα με πολύ καλή υδρομόνωση και δυνατότητα φιλτραρίσματος του νερού. Τα διάφορα πηνία συνήθως λειτουργούν σε τάση είκοσι τεσσάρων ή δώδεκα Volts αναλόγως του τύπου της εγκαταστάσεως που έχουμε επιλέξει. Στην περίπτωσή μας έχουμε επιλέξει τα 12 V DC. Πρέπει να διευκρινίσουμε ότι οι περισσότερες σύγχρονες ηλεκτροβαλβίδες μπορούν να ενεργοποιηθούν και χειροκίνητα με μερική στροφή των πηνίων τους.
- Το σύστημα του διαφράγματος βρίσκεται στην καρδιά του μηχανισμού της ηλεκτροβαλβίδας. Στην ουσία λειτουργεί σαν διαχωριστικό ανάμεσα σε δύο θαλάμους επηρεάζοντας τα χαρακτηριστικά ροής του νερού με βάση την κατάσταση λειτουργίας του πηνίου. Όλη η διαδικασία η οποία ενεργοποιεί ή διακόπτει τη ροή ύδατος οφείλεται στη δυνατότητα που παρέχεται στο νερό να ρέει μέσα από μικρής διαμέτρου οπές οι οποίες βρίσκονται στο διάφραγμα και τα υπόλοιπα τμήματα της βαλβίδας. Για αυτό έχει μεγάλη σημασία να υπάρχουν στρατηγικά τοποθετημένα φίλτρα μέσα τους προκειμένου να αποτρέπουν φερτά υλικά (πχ πετραδάκια) από το να φράσσουν αυτές τις οπές.

Η ηλεκτροβαλβίδα SIRAI τύπου L 133 που χρησιμοποιήσαμε στην κατασκευή ουσιαστικά λειτουργεί ως αυτόματη βάνα που ρυθμίζει τη ροή ή όχι του νερού. Λειτουργεί, δηλαδή, ως διακόπτης on-off στην παροχή του νερού και υποστηρίζει τη λειτουργία του PLC.

Ως προς τη συνδεσμολογία της έχει δυο ακροδέκτες χωρίς πολικότητα και μία γείωση. Εμείς συνδέουμε την έξοδο του ελεγκτή στους δυο ακροδέκτες ανεξάρτητα πολικότητας.



Εικ. 2.16 : Ηλεκτροβαλβίδα

3. Προγραμματιζόμενος Λογικός Ελεγκτής LOGO!

3.1.Γενικά

Οι προγραμματιζόμενοι λογικοί ελεγκτές (programmable logic controllers – PLC) γενικά και το LOGO! Ιδιαίτερα είναι υπολογιστές ειδικού σκοπού βασισμένοι σε μικροεπεξεργαστή, που ως σκοπό έχουν την επίτευξη βιομηχανικών και οικιακών αυτοματισμών. Συνδυάζει όλα τα πλεονεκτήματα της σύγχρονης τεχνολογίας, τόσο σε επίπεδο προγραμματισμού όσο και σε επίπεδο επικοινωνιών. Αυτό σημαίνει ότι τα μικρά PLC της SIEMENS έχουν εύκολο προγραμματισμό σε μορφή function block diagram (δηλαδή σε μια μορφή ψηφιακών πυλών με συνδυασμό ηλεκτρολογικών στοιχείων), αλλά είναι εφικτό να συνδεθούν και σε ευρύτερα δίκτυα ελέγχου τύπου AS-i.

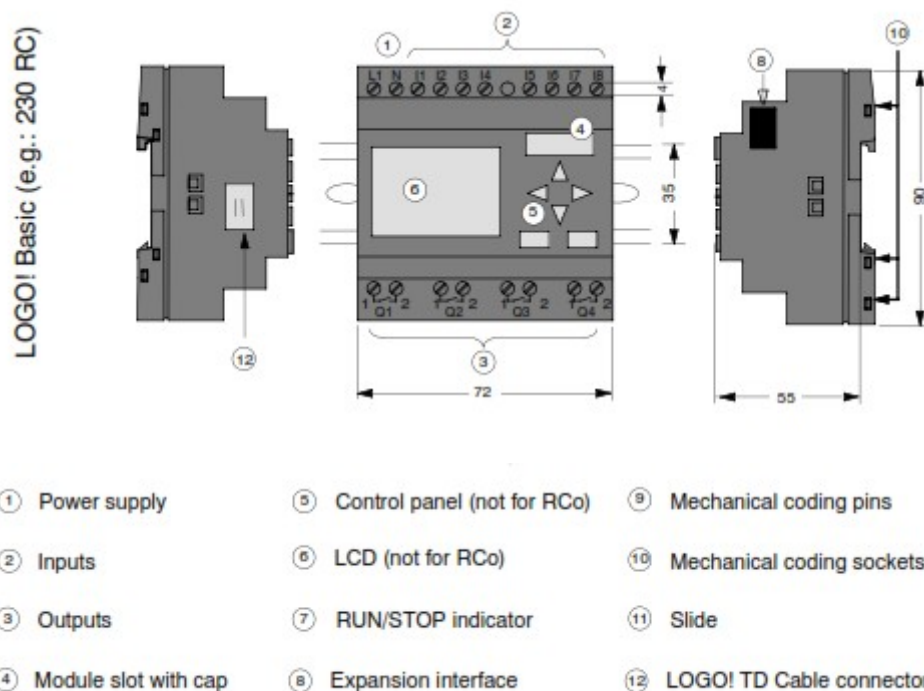
Τα LOGO! πληρούν τη βασική δομή ενός τυπικού PLC διαθέτοντας κεντρική μονάδα επεξεργασίας, μνήμη μόνιμη εσωτερική και προστιθέμενη εξωτερική, τμήμα I/O, ενσωματωμένη δυνατότητα προγραμματισμού με μορφή πλήκτρων και LCP.



Εικ. 3.1 : PLC LOGO! 230RC

Το LOGO! είναι μια παγκόσμια ενότητα λογικής της Siemens που ενσωματώνει:



- Έλεγχος
- Κουμπιά χειρισμού και φωτιζόμενη οθόνη
- Παροχή ρεύματος
- Διασύνδεση για μονάδες επέκτασης
- Διασύνδεση για την κάρτα μνήμης, μπαταρία, ή συνδυασμός κάρτας μνήμης με μπαταρία ή καλώδιο PC
- Διασύνδεση για ένα προαιρετικό τμήμα οθόνης κειμένου (TD)
- Προ-διαμορφωμένες στάνταρ συναρτήσεις λειτουργίες, όπως για παράδειγμα, χρονικά, ρελέ παλμού
- Χρονοδιακόπτες
- Ψηφιακά και αναλογικά σήματα
- Είσοδοι και έξοδοι, ανάλογα με τον τύπο της συσκευής



Εικ. 3.2 : Μορφή LOGO!

Στην εφαρμογή μας χρησιμοποιούμε το PLC LOGO! και μάλιστα του τύπου LOGO! 230 RC. Αυτός ο τύπος σημαίνει ότι λειτουργεί στα 230 V AC και έχει ως εξόδους ηλεκτρονόμους (R:Relay) και διαθέτει ρολόι (C:Clock). Γενικά στο LOGO! υπάρχουν

ΟΙ ΤΥΠΟΙ ΠΟΥ ΑΠΕΙΚΟΝΙΖΟΝΤΑΙ ΣΤΗΝ ΠΑΡΑΚΑΤΩ ΕΙΚΟΝΑ :

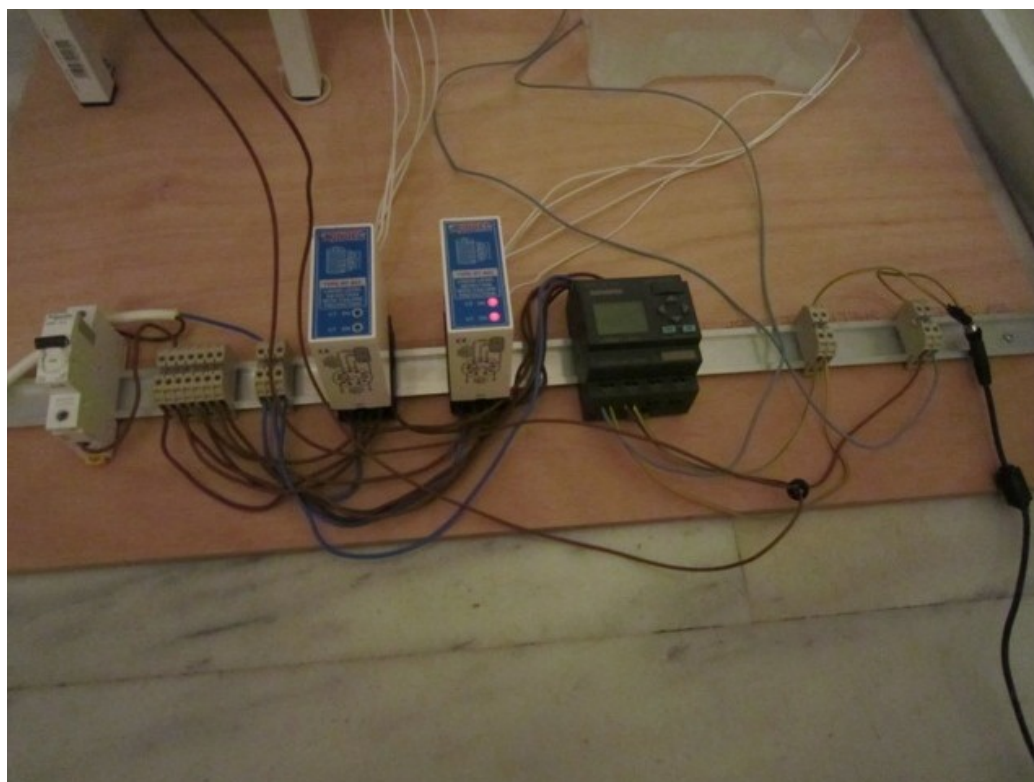
Symbol	Designation	Supply voltage	Inputs	Outputs	Properties
	LOGO! 12/24 RC	12/24 V DC	8 digital ⁽¹⁾	4 relays (10 A)	
	LOGO! 24	24 V DC	8 digital ⁽¹⁾	4 solid state 24V / 0.3A	no clock
	LOGO! 24RC ⁽³⁾	24 V AC/ 24 V DC	8 digital	4 relays (10A)	
	LOGO! 230RC ⁽²⁾	115...240 V AC/DC	8 digital	4 relays (10A)	
	LOGO! 12/24RCo	12/24 V DC	8 digital ⁽¹⁾	4 relays (10A)	no display unit no keyboard
	LOGO! 24o	24 V DC	8 digital ⁽¹⁾	4 solid state 24 V / 0.3A	no display unit no keyboard no clock
	LOGO! 24RCo ⁽³⁾	24 V AC / 24 V DC	8 digital	4 relays (10A)	no display unit no keyboard
	LOGO! 230RCo ⁽²⁾	115...240 V AC/DC	8 digital	4 relays (10A)	no display unit no keyboard

(1): Of those can be used alternatively: 4 analog inputs (0 ... 10V) and 4 fast digital inputs.

(2): 230 V AC versions: Two groups consisting of 4 inputs each. Each input within a group must be connected to the same phase. It is possible to interconnect groups with a different phase.

(3): The digital inputs can be operated with P or N action.

Εικ. 3.3 : Οι τύποι του LOGO!



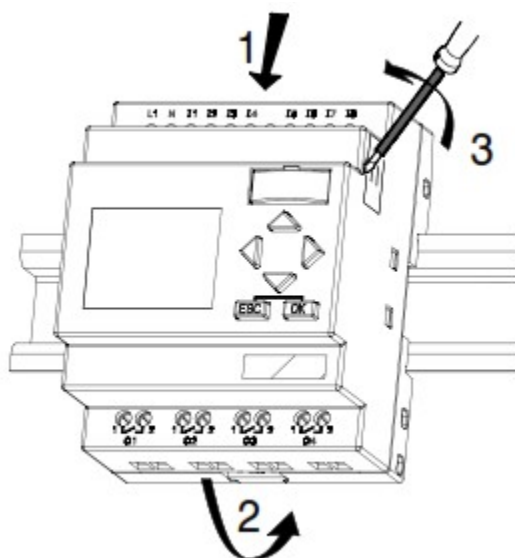
Εικ. 3.4 : Ράγα τοποθέτησης LOGO!

Τα σύμβολα που χρησιμοποιούνται για τις διάφορες εκδόσεις του LOGO! εξηγούνται ως εξής :

- 24 : 24 V DC
- 230 : 230 V AC
- R : Έξοδοι με ηλεκτρονόμους
- C : Ρολόι

3.2. Εγκατάσταση και καλωδίωση

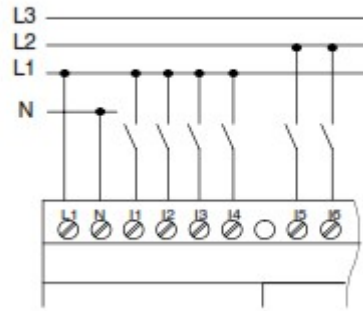
Το LOGO! Εγκαθίσταται πάντα σε distribution box εξασφαλίζοντας ότι οι γραμμές εισόδου και εξόδου δεν έχουν γυμνά σημεία ικανά να προκαλέσουν ηλεκτροπληξία. Εγκαθίσταται πάνω σε ράγα των 35 mm (DIN EN 50022) και σύμφωνα με την παρακάτω εικόνα :



Εικ. 3.5 : Εγκατάσταση LOGO!

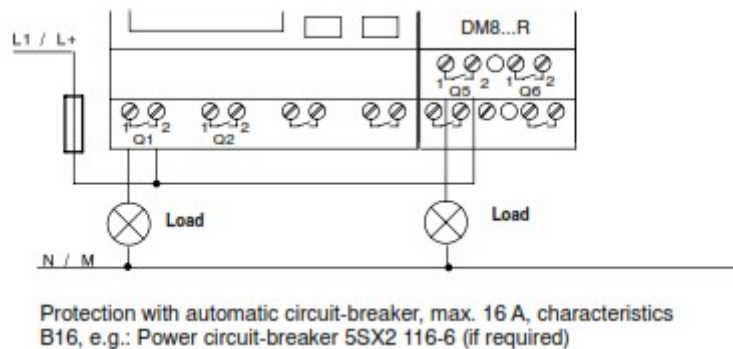
Η δε καλωδίωση μπορεί να γίνει με καλώδια των $1 \times 1,5 \text{ mm}^2$. Μ' αυτά τα καλώδια της τροφοδοσίας μπορούν να συνδεθούν 115 και 230 V για τον τύπο LOGO! 230 και 24 V για τον τύπο LOGO! 24. Ακόμα με τα ίδια καλώδια συνδέονται και οι επιτηρητές στις εισόδους του PLC σύμφωνα με την παρακάτω εικόνα :

LOGO! 230



Εικ. 3.6 : Καλωδίωση εισόδων και τροφοδοσίας PLC

Ότι αφορά τις εξόδους στο μοντέλο LOGO! BASIC που χρησιμοποιούμε μπορούν να αντέξουν μέγιστο ρεύμα 10 A για μη επαγωγικό φορτίο και 3 A για επαγωγικό φορτίο.



Εικ. 3.7 : Καλωδίωση εξόδων PLC

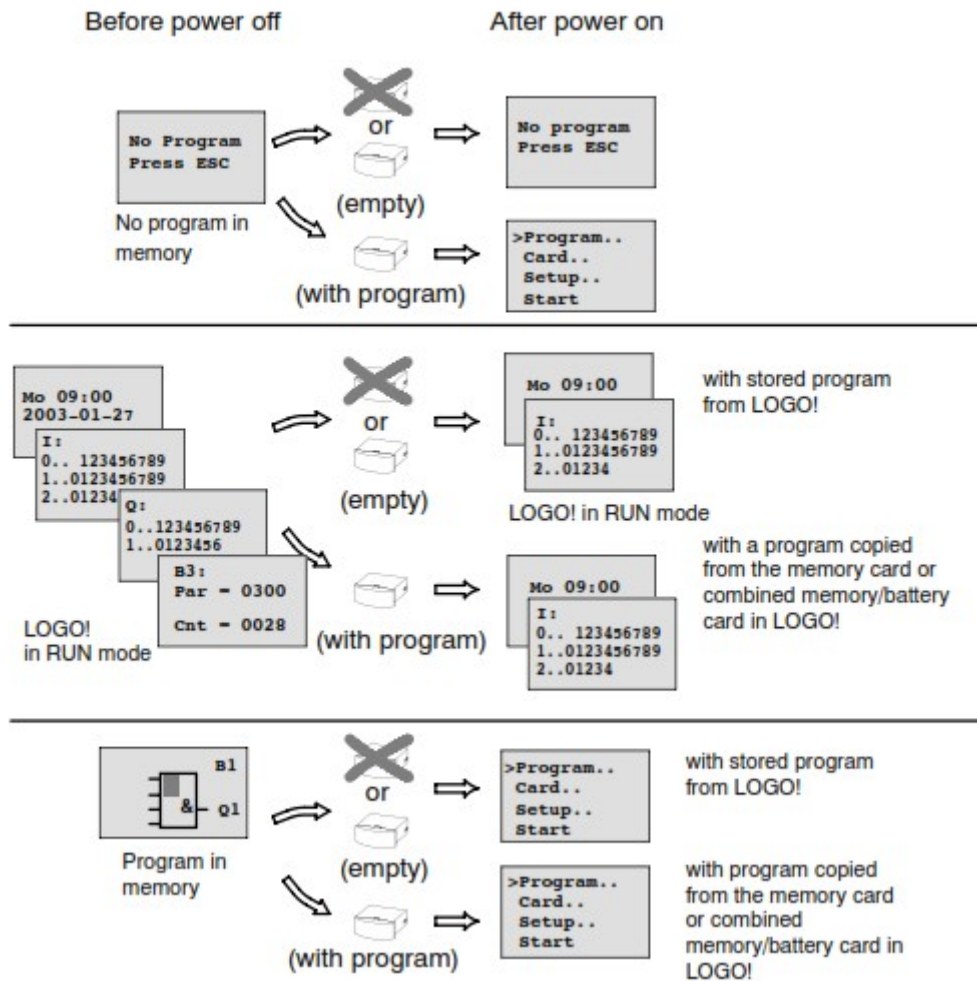
3.3. Καταστάσεις λειτουργίας (RUN) και διακόπτης (STOP)

Το LOGO! Έχει δύο καταστάσεις λειτουργίας την STOP και την RUN. Όταν είναι σε κατάσταση STOP, οι εισοδοί δεν μεταβιβάζονται, το πρόγραμμα δεν εκτελείται, οι επαφές των ηλεκτρονόμων εξόδου είναι πάντοτε ανοικτές. Όταν είναι σε κατάσταση RUN εμφανίζεται στην οθόνη του PLC η μάσκα με τις καταστάσεις των εξόδων και των εισόδων. Τότε το LOGO! Διαβάζει τις καταστάσεις των εισόδων, υπολογίζει την κατάσταση των εξόδων με βάση το πρόγραμμα και ανοιγοκλείνει τις επαφές των ηλεκτρονόμων.

Ένα λεπτό σημείο που πρέπει να προσεχθεί ιδιαίτερα είναι το τι θα συμβεί στο πρόγραμμα όταν διακοπεί η τροφοδοσία του PLC, είτε αυτό είναι σε κατάσταση RUN, είτε σε κατάσταση STOP, και μπορούν να συμβούν τα εξής :

1. Εάν δεν υπήρχε πρόγραμμα στο LOGO! Ή στη μικρή συσκευή μνήμης μετά τη διακοπή και επαναφορά της τροφοδοσίας δεν θα υπάρχει πάλι πρόγραμμα και θα εμφανίζεται το μήνυμα No program.
2. Υπάρχει πρόγραμμα στη συσκευή μνήμης, τότε αυτό μετά τη διακοπή και επαναφορά της τροφοδοσίας αντιγράφεται αυτόματα στο LOGO!.Εάν υπάρχει άλλο πρόγραμμα μέσα, τότε αυτό διαγράφεται από το νέο πρόγραμμα.
3. Εάν υπάρχει πρόγραμμα στο LOGO! ή σε συσκευή μνήμης, το LOGO! προσαρμόζει την κατάστασή του όπως ήταν πριν τη διακοπή της τροφοδοσίας.
4. Σε περίπτωση διακοπής και για τον τύπο LOGO! που χρησιμοποιούμε, μηδενίζονται όλοι οι χρονιστές και οι απαριθμητές.

Όλα αυτά απεικονίζονται στην παρακάτω εικόνα 3.8:



Εικ. 3.8 : Περιπτώσεις επαναφοράς προγράμματος μετά από διακοπή τροφοδοσίας

3.4. Προγραμματισμός LOGO!

Προγραμματισμός στο LOGO! Σημαίνει εισαγωγή ενός ηλεκτρολογικού κυκλώματος που αναπαρίσταται όχι σε συμβατική μορφή αλλά σε μορφή που μπορεί να εμφανίζεται στην οθόνη του PLC. Έτσι λοιπόν στη συνέχεια θα εξετάσουμε τα δύο βασικά στοιχεία με τα οποία προγραμματίζουμε το LOGO!, τα connector και τα blocks.

Τα connectors συμβολίζονται στην οθόνη του PLC με **CO** και αποτελούνται από τις εισόδους, τις εξόδους και τρεις ακόμη καταστάσεις.

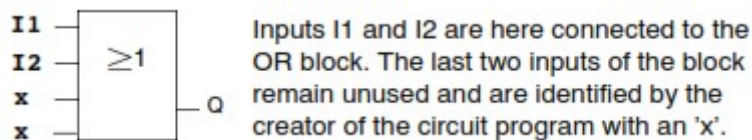
1. Οι εισόδοι (συμβολίζονται με **I**) και οι έξοδοι (που συμβολίζονται με **Q**) μπορούν να έχουν κατάσταση «0» και «1». Το «0» σημαίνει ότι δεν υπάρχει τάση στην

είσοδο και το «1» σημαίνει ότι υπάρχει.

2. Τα connector **hi** που είναι μια από τις τρεις καταστάσεις, ορίσθηκε ως «1» για να διευκολύνει το σχεδιασμό του κυκλώματος. Αντίστοιχα το **lo**, σημαίνει «0».
3. Τέλος η τρίτη κατάσταση αντιπροσωπεύεται από το **CO «x»** που χρησιμοποιείται όταν διαθέτουμε ένα block με πολλές εισόδους και κάποιες από τις εισόδους μας είναι αδιάφορες ή δε θέλουμε να τις χρησιμοποιήσουμε.

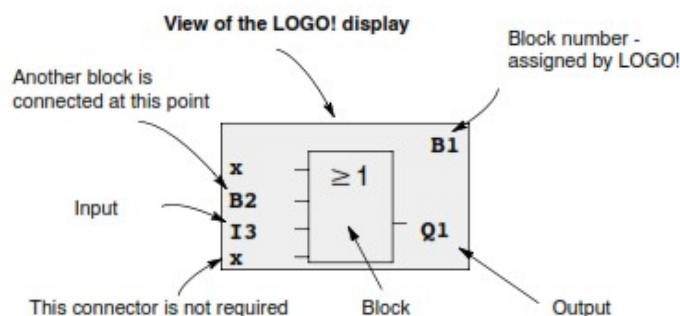
Γενικά θα μπορούσαμε να ισχυριστούμε ότι στη λέξη connector περιλαμβάνονται τόσο όλες οι συνδέσεις εισόδου–εξόδου, όσο και καταστάσεις που χρησιμοποιούνται στα LOGO!.

Από την άλλη πλευρά τα block στο LOGO! Είναι η συνάρτηση η οποία μετατρέπει την πληροφορία εισόδου σε πληροφορία εξόδου. Ουσιαστικά ο προγραμματισμός του PLC συνίσταται στη σύνδεση του connector με block, δηλαδή, στη σύνδεση της πληροφορίας εισόδου ή εξόδου ενός PLC με τη συνάρτηση η οποία επεξεργάζεται κατάλληλα το σήμα που λαμβάνει στην είσοδό της. Παρακάτω φαίνεται ένα απλό block :



Εικ. 3.9 : Πύλη OR

Οι εισόδοι I_1 και I_2 συνδέονται στο block OR. Οι δύο τελευταίες εισόδοι του block δεν χρησιμοποιούνται και γι' αυτό σημειώνονται με «x». Μερικές πιο σύνθετες μορφές block, καθώς και το πώς απεικονίζονται στην οθόνη του LOGO! Φαίνονται στη παρακάτω εικόνα 3.10:



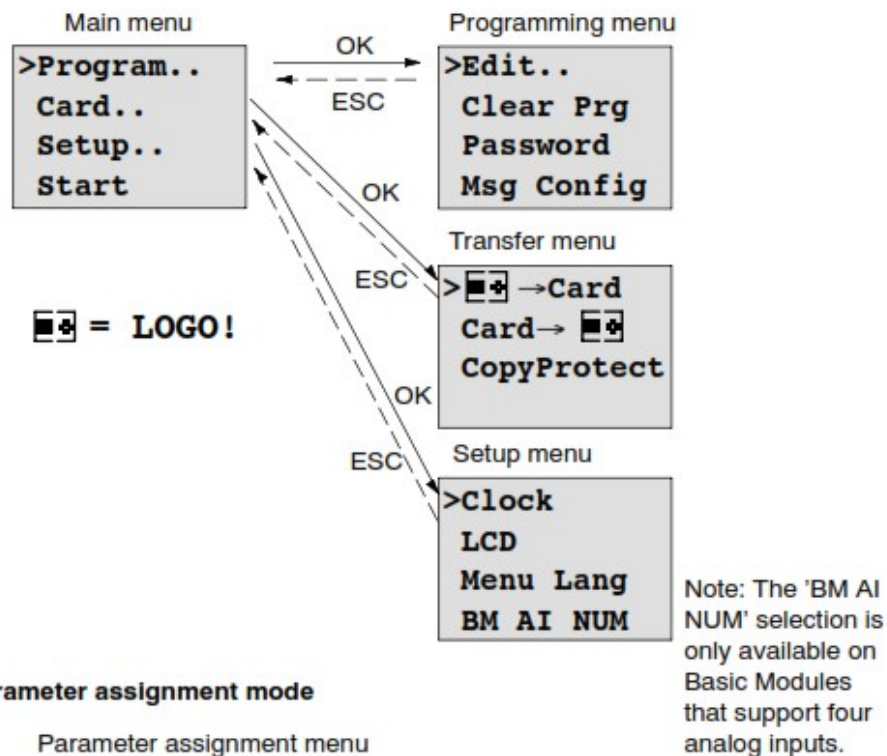
Εικ. 3.10 : Απεικόνιση block στο LOGO!

Το LOGO! Έχει τη δυνατότητα της παράστασης ενός και μόνο block τη φορά. Γι' αυτό εισάγεται αυτόματα από το πρόγραμμα αρίθμησης των block (B_) που βοηθά στον έλεγχο της δομής της κυκλωματολογίας. Κάθε φορά που εισάγεται ένα block, αριθμείται αυτόματα και ο αριθμός αυτός χρησιμοποιείται για τη διασύνδεση μεταξύ των block. Αυτό σημαίνει ότι η είσοδος B2 του block B1, αντιπροσωπεύει την έξοδο του block B2 (όπως φαίνεται στην παραπάνω εικόνα). Με αυτόν τον τρόπο δίνεται δυνατότητα της χρήσης εσωτερικών αποτελεσμάτων, του συνολικού κυκλώματος, καθώς και μεταβλητών για περισσότερο από μια φορά. Τελικά η χρήση της μνήμης ελαχιστοποιείται και ο απαιτούμενος χρόνος εργασίας μειώνεται.

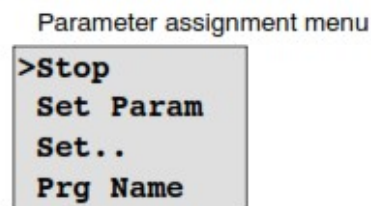
3.5. Εισαγωγή προγράμματος στο LOGO!

Στο LOGO! Υπάρχουν τέσσερα βασικά μενού, μέσα από τα οποία μπορεί κάποιος να γράψει πρόγραμμα, να το τρέξει, να το σβήσει, να το στείλει από τον υπολογιστή στο PLC και αντίθετα.

Programming mode



Parameter assignment mode



Εικ. 3.11 : LOGO!'s menus

Υπάρχουν επίσης τέσσερις βασικοί κανόνες για τον προγραμματισμό των LOGO! :

Κανόνας 1 : Για να εισάγεις το πρόγραμμα πρέπει πρώτα να θέσεις το PLC σε κατάσταση προγραμματισμού (programic mode) πατώντας ταυτόχρονα τα πλήκτρα δεξί βέλος, αριστερό βέλος και OK. Επίσης για να θέσεις τους χρόνους και τις παραμέτρους πρέπει να θέσεις το PLC σε κατάσταση παραμετροποίησης (parameterization mode), πατώντας ταυτόχρονα τα πλήκτρα escape και OK.

Κανόνας 2 : Η εισαγωγή του κυκλώματος γίνεται από την έξοδο προς την είσοδο.

Κανόνας 3 : Όταν ο κέρσορας εμφανίζεται με τη μορφή μιας παλόμενης γραμμής, μπορείς να μετακινήσεις τον κέρσορα χρησιμοποιώντας τα πλήκτρα που αντιπροσωπεύουν τα βέλη, μπορείς να επιλέξεις ένα connector ή block πατώντας το OK, μπορείς να πατήσεις το escape για να βγεις από το κύκλωμα.

Όταν ο κέρσορας εμφανίζεται σε μορφή block μπορείς να επιλέξεις ένα connector ή block, χρησιμοποιώντας τα βέλη άνω και κάτω κατεύθυνσης, μπορείς να αποδεχθείς μία επιλογή πατώντας το OK ή πατώντας το escape να μετακινηθείς στη προηγούμενη επιλογή.

Κανόνας 4 : Πριν εισάγεις την κυκλωματολογία να έχεις σχεδιάσει ολόκληρο το κύκλωμα γιατί το LOGO! δέχεται μόνο ολοκληρωμένα κυκλώματα , διαφορετικά δεν εξέρχεται από την κατάσταση προγραμματισμού.

Η εισαγωγή και η διόρθωση θα περιγραφεί στην παράγραφο που δίνεται το πρόγραμμα. Ως αναφορά το Functions που αντιπροσωπεύουν τα blocks χωρίζονται σε τέσσερις ομάδες. Αυτές είναι :

1. Co : Ομάδα με connector για
 - Εισόδους : I_1, \dots
 - Εξόδους : Q_1, \dots
 - Επίπεδο τάσης : l_0, h_i
 - Μη συνδεδεμένα : x
2. GF : Ομάδα με τις βασικές συναρτήσεις AND, OR, NOT, ...
3. SF : Ομάδα των ειδικών δυνάμεων
4. BN : Ομάδα των block που ήδη έχουν αριθμηθεί στο κύκλωμα

Τα GF και SF δίνονται σε μορφή πίνακα.

Υπάρχει μία περίπτωση το LOGO! Να μην απεικονίζει τίποτα στην οθόνη του.

Γεγονός που μπορεί να συμβεί μόνο όταν :

1. Κανένα block δε μπορεί να εισαχθεί επειδή ξεπεράστηκε ο μέγιστος αριθμός των τριάντα block που μπορούν να εισαχθούν.
2. Ένα special block απαιτεί περισσότερη μνήμη από τη μνήμη που έχει απομείνει στο LOGO!.
3. Ο αριθμός των blocks που συνδέονται σε σειρά να έχει περάσει το μέγιστο αριθμό των επτά blocks.

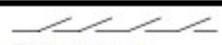
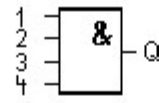
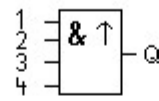

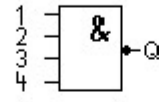
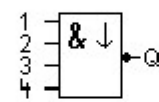

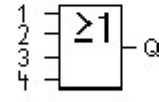
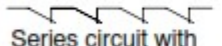
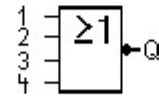
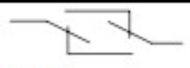
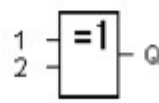
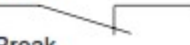
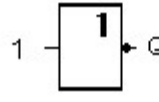


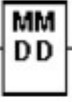
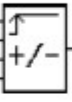
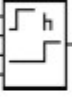






View in the circuit diagram	View in LOGO!	Name of the basic function
 <p>Series circuit make contact</p>		AND
		AND with edge evaluation
 <p>Parallel circuit with break contacts</p>		NAND (not AND)
		NAND with edge evaluation
 <p>Parallel circuit with make contacts</p>		OR
 <p>Series circuit with break contacts</p>		NOR (not OR)
 <p>Double changeover contact</p>		XOR (exclusive OR)
 <p>Break contact</p>		NOT (negation, inverter)

Fig. 3.12 : Basic functions

View in LOGO!	Name of the special function	Rem
Times		
	On-delay	REM
	Off-delay	REM
	On-/Off-delay	REM
	Retentive on-delay	REM
	Wiping relay (pulse output)	REM
	Edge triggered wiping relay	REM
	Asynchronous pulse generator	REM
	Random generator	
	Stairway lighting switch	REM

View in LOGO!	Name of the special function	Rem
Trg R Par  Q	Multiple function switch	REM
No1 No2 No3 Par  Q	Weekly timer	
No  Q	Yearly timer	
Counter		
R Cnt Dir Par  Q	Up/down counter	REM
R n Ral Par  Q	Hours counter	REM
Fre Par  Q	Threshold trigger	
Analog		
Ax Par  Q	Analog threshold trigger	
Ax Par  Q	Analog differential trigger	
Ax Ay Par  Q	Analog comparator	
En Ax Par  Q	Analog value monitoring	
Ax Par  AQ	Analog amplifier	

View in LOGO!	Name of the special function	Rem
	Analog multiplexer	
	Pulse Width Modulator (PWM)	
	Analog math	
	Analog ramp	
	PI controller	REM
Miscellaneous		
	Latching relay	REM
	Pulse relay	REM
	Message texts	
	Softkey	REM
	Shift register	REM
	Analog math error detection	

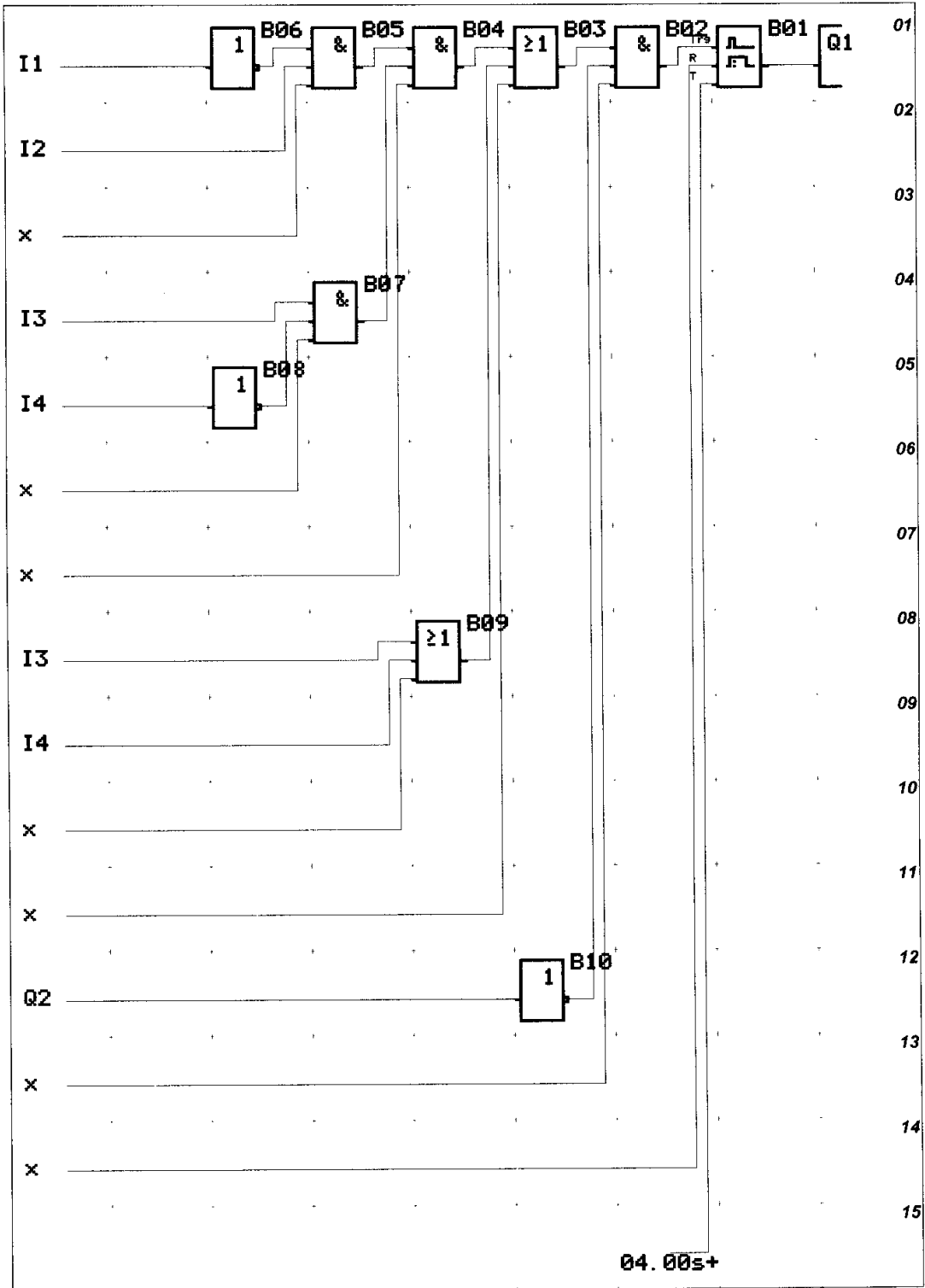
EIK. 3.13 : Special functions

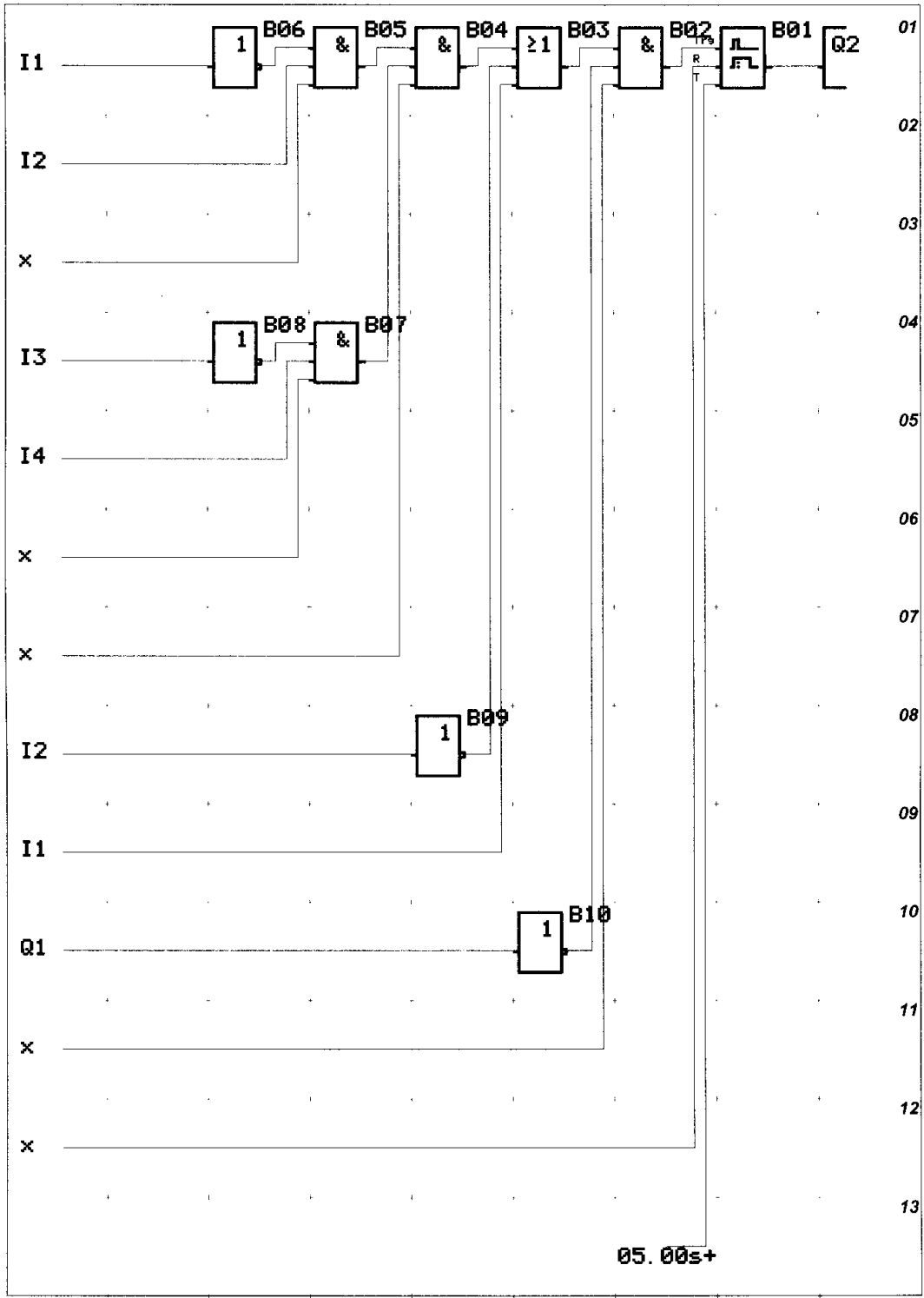
3.6. Πρόγραμμα ελέγχου στάθμης υγρού δεξαμενών

Το πρόγραμμα του PLC στηρίζεται στη λογική των ψηφιακών πυλών σε συνδυασμό με ορισμένα ηλεκτρολογικά στοιχεία. Για τον έλεγχο της στάθμης χρησιμοποιήσαμε πύλες AND, OR, NOT, timer. Ως έξοδοι στα Q_1 και Q_2 αντιστοιχήσαμε την αντλία και την ηλεκτροβαλβίδα αντίστοιχα. Υπάρχουν δύο επιπλέον έξοδοι στις οποίες θα μπορούσαμε να συνδέσουμε ηχητικό ή φωτεινό σήμα για την περίπτωση που υπάρξει κίνδυνος για υπερχείλιση των δεξαμενών μας.

Στην έξοδο Q_1 συνδέεται η έξοδος Q_2 με αναστροφή, ώστε όταν η τελευταία έξοδος είναι ενεργοποιημένη να μην επιτρέπεται η ενεργοποίηση της Q_1 . Είναι δηλαδή μανδαλωμένες μεταξύ τους. Αν λοιπόν όλες οι έξοδοι είναι απενεργοποιημένες για να εκκινήσει η Q_1 πρέπει τα αισθητήρια I_2 και I_3 να δέχονται σήμα ενώ τα αισθητήρια I_1 και I_4 να μην έχουν σήμα. Δηλαδή πρέπει η δεξαμενή A να μην είναι γεμάτη και η αντλία της B να έχει επαρκή ποσότητα νερού.

Όμοια και στην έξοδο Q_2 υπάρχει μανδάλωση ενώ ενεργοποιείται με τα σήματα I_2 και I_4 και απενεργοποιείται με τα σήματα I_1 και I_3 . Στην επόμενη παράγραφο φαίνεται το πρόγραμμα του PLC που χρησιμοποιήσαμε στην εφαρμογή μας.





ΣΥΜΠΕΡΑΣΜΑΤΑ

Στόχος αυτής της εργασίας είναι η χρήση αυτοματισμών και ο τρόπος που αυτοί επενεργούν στον έλεγχο στάθμης δεξαμενών. Για να το πετύχουμε αυτό κατασκευάσαμε ένα πειραματικό μοντέλο το οποίο ελέγχει τη στάθμη σε δύο μοντέλα δεξαμενών. Μπορεί να φροντίσει αυτόματα ώστε να μην πέσει η στάθμη του υγρού κάτω από ένα κατώτερο όριο και αν συμβεί αυτό να μπορεί να το εφοδιάζει με υγρό από την άλλη δεξαμενή. Επίσης αν το υγρό για οποιοδήποτε λόγο υπερβεί κάποιο ανώτερο όριο τότε ενεργοποιείται πάλι αυτόματα μια αντλία η οποία αδειάζει τη δεξαμενή και με αυτόν τον τρόπο αποφεύγεται η υπερχειλίση της. Όλοι αυτοί οι έλεγχοι καθώς και οι ενέργειες που απαιτούνται γίνονται αυτόματα με χρήση των κατάλληλων εξαρτημάτων χωρίς την ανθρώπινη παρουσία.

Οι εφαρμογές αυτής της κατασκευής είναι πολλές στην καθημερινή ζωή και ακόμη περισσότερες σε μέρη που υπάρχει έλλειψη νερού και απαιτείται η αποθήκευση του σε δεξαμενές αλλά και σε μέρη που η ποιότητα του νερού είναι υποβαθμισμένη και δεν είναι κατάλληλο για να χρησιμοποιηθεί σε όλες τις εφαρμογές. Μερικές από τις εφαρμογές στις οποίες χρησιμοποιούνται αυτοματισμοί για τη διαχείριση κάποιου υγρού είναι σε γεωτρήσεις, αντλιοστάσια, μονάδες επεξεργασίας νερού, υδροφράγματα, αλλά ακόμα περισσότερο εφαρμογή έχει στα δίκτυα νερού που συναντάμε σε χωριά και πόλεις.

Για την κατασκευή του μοντέλου απαιτήθηκε να αντιμετωπίσουμε αρκετά προβλήματα καθώς οι συνθήκες λειτουργίας του δεν ήταν πραγματικές και η ανεύρεση όλων των εξαρτημάτων έγινε έτσι ώστε να λειτουργούν σε μικρές παροχές νερού. Επιπλέον κάποια από τα εξαρτήματα λειτουργούσαν σε διαφορετικές τάσεις και χρειάστηκε να μετασχηματίσουμε την τροφοδοσία μας ώστε να λειτουργήσουν σωστά.

Τέλος, ένα άλλο σημαντικό πρόβλημα ήταν να καταφέρουμε να συνδέσουμε τις επαφές εξόδου του επιτηρητή στάθμης με τις εισόδους του PLC και να συντονίσουμε ανάλογα με αυτές το πρόγραμμά μας ώστε να μας δώσει τα επιθυμητά αποτελέσματα. Παρά τις δυσκολίες η κατασκευή μας λειτούργησε και τα αποτελέσματά της επιβεβαίωσαν τα αναμενόμενα συμπεράσματα που υποθέσαμε πριν την πραγματοποίησή της.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. W. Bolton, "Programmable Logic Controllers", Newnes (ELSEVIER), fifth edition, 2009, p. 400
2. Jay F. Hooper, "Introduction to PLCs", Carolina Academic Press, second edition, 2006, p. 108
3. Jon Stenerson, "Programmable Logic Controllers with ControlLogix", Delmar Cengage Learning, 2009, p. 514
4. E. A. Parr, "Programmable Controllers : an engineer's guide", Newnes, third edition, 2003, p. 429
5. Stephen, Philip Tubbs, "Programmable Logic Controller (PLC) Tutorial, Siemens Simatic S7-200", Stephan P. Tubbs, 2007, p. 131
6. Kelvin T. Erickson, "Programmable Logic Controllers : an emphasis on design and application", Dogwood Valley Press, 2005, p. 1460
7. Keith Clements – Jewery, W. Jeffcoat, "The PLC workbook : programmable logic controllers made easy", Prentice Hall, 1996, p. 197
8. Frank D. Petruzella, "Programmable Logic Controllers", Mc Graw-Hill Companies, fourth edition, 2010, p. 512
9. Kevin Collins, "PLC Programming for industrial automation", Meadow Books, 2007, p. 140
10. Stephen J. Ball, "Education PLC : understanding private sector participation in public sector education", Taylor & Francis Group, 2007, p. 216
11. Peter Rohner, "Automation with Programmable logic Controllers", KNSW Press, 1996, p. 226
12. Srinivas Medida, "Pocket guide on industrial automation for engineers and technicians", IDC Technologies, p. 171
13. John R. Hackworth and Frederick D. Hackworth, Jr., "Programmable logic Controllers: Programming methods and Applications", Pearson/Prentice Hall, 2004, p. 303

14. Myke Predko, "Programming and customizing the PLC microcontroller", The Mc Graw Hill Companies, third edition, 2008, p. 1293
15. Henry Ward Beecher, "Introduction to PLC Programming and Implementation – from relay logic to PLC logic", Industrial Text and Video Company, 1999, p. 76
16. C.T. Jones, "Programmable logic Controllers : the complete guide to the technology", Patrik – turner, 1996, p. 464
17. James A. Rehg, Glenn J. Sartori, "Programmable logic Controllers", Prentice Hall, 2007, p. 606
18. Max Rabiee, "Programmable logic Controllers : Hardware and programming", Goodheart – Willcox Publ, 2009, p. 368
19. Rex Miller, Mark Miller, "Industrial Electricity and Motor Controls", The Mc Graw – Hill Companies, 2008, p. 448
20. William C. Dunn, "Fundamentals of industrial instrumentation and process control", Mc Graw – Hill, 2005, p. 337
21. Ned Mohan, Tore M. Undeland, William P. Robbins, "Power Electronics, converters, application and design" John Wiley & sons, second edition, 1995, p. 820
22. Steven F. Barrelet and Daniel J. Pack, "Microcontrollers Fundamentals for engineers and scientists", Morgan & Claypool, 2006, p. 126
23. Muhammad H. Rashid, "Power Electronics Handbook", Elsevier Inc., third edition, 2011, p. 1229
24. Andrej M. Pawlak, "Sensors and actuators in mechatronics", Taylor & Francis Group, 2007, p. 409
25. Syedkamrul Islam, Mohammad Rafiqua Haider, "Sensors and low power signal processing", Springer, 2010, p. 125

LOGO! functions

Organization

LOGO! provides you with various elements in programming mode. In order to maintain the overview, we have organized the elements in 'Lists'. These lists are:

- ↓**Co**: Connector list (**C**onnector)
(see Chapter 4.1)
- ↓**GF**: List of the basic functions AND, OR, ...
(refer to Chapter 4.2)
- ↓**SF**: List of the special functions
(refer to Chapter 4.4)
- ↓**BN**: List of reusable blocks configured in the circuit program

List contents

All lists show the elements available in LOGO! Usually, this includes *all* connectors, basic functions, and special functions. The ↓**BN** list shows all the blocks that you have created in LOGO!.

If not all is shown

LOGO! does *not* show all elements if:

- No further blocks may be added.

This is either an indication of insufficient memory space or that the maximum number of blocks has been reached.
- A specific block's memory space requirement would exceed the space available in LOGO!

4.1 Constants and connectors - Co

Constants and connectors (= Co) represent inputs, outputs, flags and constant voltage levels (constants).

Inputs:

1) Digital inputs

Digital inputs are identified with an **I**. The number of the digital inputs (I1, I2, ...) corresponds to the number of the input connectors of the LOGO! Basic and of the connected digital modules, in the order of their installation. The fast digital inputs I3, I4, I5, and I6 of the LOGO! versions LOGO! 24, LOGO! 24o, LOGO! 12/24RC and LOGO! 12/24RCo can be used as fast counters.

2) Analog inputs

The LOGO! versions LOGO! 24, LOGO! 24o, LOGO! 12/24RC and LOGO! 12/24RCo are equipped with the inputs I1, I2, I7 and I8, which can also be programmed for use as **AI3, AI4, AI1** and **AI2** inputs. As described in section 5.2.4, you can configure these modules to use either two analog inputs (AI1 and AI2), or all four. Signals at the I1, I2, I7 and I8 inputs are interpreted as digital values, and those at the AI3, AI4, AI1 and AI2 inputs are interpreted as analog values. Note that AI3 corresponds to I1 and AI4 corresponds to I2. This numbering preserves the previous correspondence of AI1 to I7 and AI2 to I8 that was available with the 0BA5 series. The inputs of a connected analog module are numbered according to the already existing analog inputs. See section 2.1.1 for example setups. In programming mode, when you select the input signal of a special function that takes an analog input, LOGO! offers the analog inputs AI1...AI8, analog flags AM1...AM6, analog outputs AQ1 and AQ2, and the block numbers of functions with analog outputs.

Outputs:

1) Digital outputs

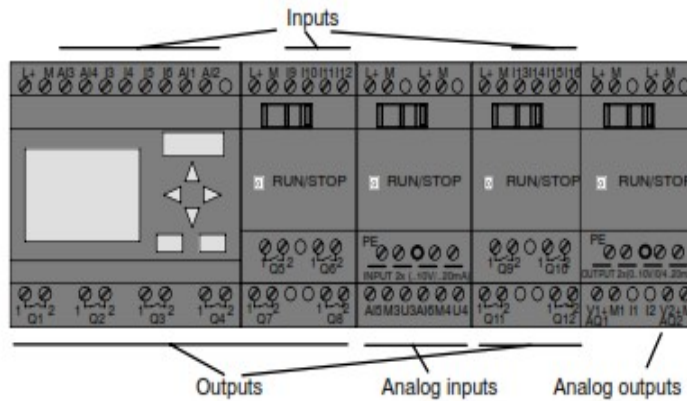
Digital outputs are identified by the character **Q**. The output numbers (Q1, Q2, ... Q16) correspond with the numbers of the output connectors at the LOGO! Basic and with those of the expansion modules, in their order of installation. See the following figure.

There are also 16 blank outputs available. These are identified with an **x** and cannot be reused in a circuit program (in contrast to flags, for example). The list shows all programmed blank outputs, and one blank output which is not yet configured. A blank output, for example, is useful for the special function "Message texts" (see Chapter 4.4.23), if only the message text is of significance to a circuit program.

2) Analog outputs

Analog outputs are identified by the letters **AQ**. Two analog outputs are available, namely AQ1 and AQ2. An analog output can only be connected with the analog input of a function, an analog flag AM or an analog output connector.

The following figure shows an example LOGO! configuration and the numbering of the inputs and outputs for the circuit program.



Flag blocks

Flag blocks are identified by the letters **M** or **AM**. These are virtual outputs, which output the value of their inputs. LOGO! provides 27 digital flags M1 ... M27 and 6 analog flags AM1 ... AM6.

Startup flag

Flag M8 is set in the first cycle of the user program and can thus be used in your circuit program as a startup flag. This signal is automatically reset after the circuit program has completed its first cycle.

The M8 flag can be used in all further cycles for setting, deletion and evaluation procedures in the same way as other flags.

Backlight flags M25 and M26

The M25 flag controls the backlight of the LOGO! Display. The M26 flag controls the backlight of the LOGO! TD.

Note: The backlight lifetime of the LOGO! TD is 20,000 hours.

Message text character set flag M27

The M27 flag selects between the two character sets that LOGO! uses to display message texts. State 0 corresponds to Character Set 1, and state 1 corresponds to Character Set 2. If M27=0 (low), only message texts configured for Character Set 1 will display; if M27=1 (high), only message texts configured for Character Set 2 will display. If you do not include M27 in the circuit program, message texts display in the character set that you selected from either the Msg Config menu or from LOGO!Soft Comfort.

Note

The output of the flag always carries the signal of the previous program cycle. This value does not change within the same program cycle.

Shift register bits

LOGO! provides the shift register bits S1 to S8, which are assigned the read-only attribute in the circuit program. The content of shift register bits can only be modified by means of the "Shift register" special function (see Chapter 4.4.25).

Cursor keys

Up to four cursor keys are available to you, namely C ▲, C ►, C ▼ and C ◀ ("C" = "Cursor"). Cursor keys are programmed for the circuit program in the same way as other inputs. You can set the cursor keys in the corresponding display while the system is in RUN (see Chapter 3.7.6), and in an active message text (ESC + Key). Cursor keys can save switches and inputs, and allow operator control of the circuit program. Cursor key inputs from the LOGO! TD are identical to cursor key inputs from the LOGO! module.

LOGO! TD function keys

The LOGO! TD has four function keys that you can use in your circuit program. You program these keys in the same way as other inputs. Like the cursor keys, you can press these keys when LOGO! is in RUN mode to affect the behavior of the circuit program, and to save switches and inputs. They are identified as F1, F2, F3, and F4.

Levels

Voltage levels are designated **hi** and **lo**. A constant "1" = hi or "0" = lo status at the block can be set by means of a permanent voltage level or constant value hi or lo.

Open connectors

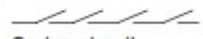
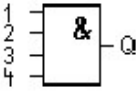
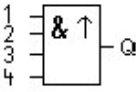
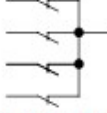
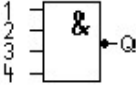
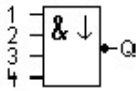

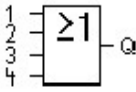
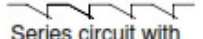
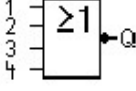

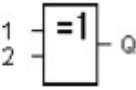

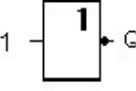
Unused block connectors can be identified with an **x**.

4.2 Basic functions list - GF

Basic functions represent simple logical elements of Boolean algebra.

You can invert the inputs of individual basic functions, that is, the circuit program inverts a logical "1" at a relevant input to a logical "0"; if "0" is set at the input, the program sets a logical "1". A programming example is found in Chapter 3.7.3.

The GF list contains the basic function blocks you can use for your circuit program. The following basic functions are available:

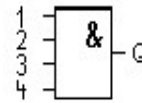
View in the circuit diagram	View in LOGO!	Name of the basic function
 <p>Series circuit make contact</p>		AND (see page 106)
		AND with edge evaluation (see page 106)
 <p>Parallel circuit with break contacts</p>		NAND (not AND) (see page 107)
		NAND with edge evaluation (see page 108)
 <p>Parallel circuit with make contacts</p>		OR (see page 109)
 <p>Series circuit with break contacts</p>		NOR (not OR) (see page 109)
 <p>Double changeover contact</p>		XOR (exclusive OR) (see page 110)
 <p>Break contact</p>		NOT (negation, inverter) (see page 111)

4.2.1 AND

Circuit diagram of a series circuit with several make contacts:



Symbol in LOGO:



The output of the AND is only 1 if **all** inputs are 1, i.e. all contacts are closed.

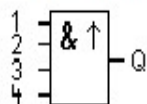
At an unused block input (x): x = 1.

AND function logic table

1	2	3	4	Q
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

4.2.2 AND with edge evaluation

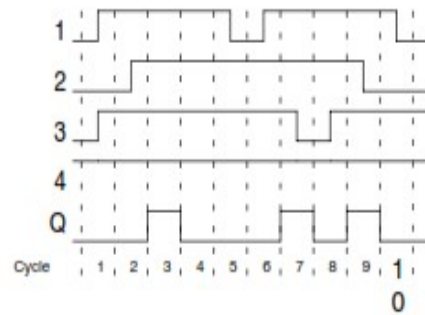
Symbol in LOGO:



The output of an edge triggered AND is only 1 if **all** inputs are 1 and if **at least one** input was low in the previous cycle.

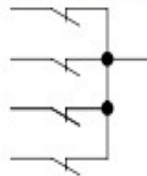
At an unused block input (x): x = 1.

Timing diagram for the AND with edge evaluation

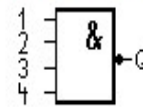


4.2.3 NAND (not AND)

Parallel circuit with multiple break contacts in the circuit diagram:



Symbol in LOGO!:



The output of the NAND is only 0 if the status at **all** inputs is 1, i.e. the contacts are closed.

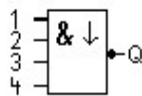
At an unused block input (x): $x = 1$.

NAND function logic table

1	2	3	4	Q
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

4.2.4 NAND with edge evaluation

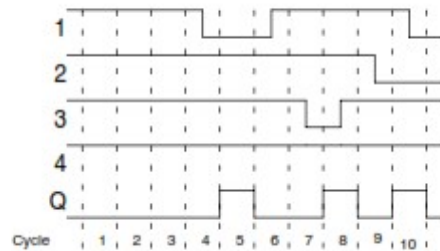
Symbol in LOGO!



The output status of the NAND with edge evaluation is only 1 if **at least one** input is 0 and if **all** inputs were 1 in the previous cycle.

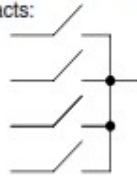
At an unused block input (x): $x = 1$.

Timing diagram for the NAND with edge evaluation

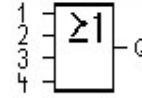


4.2.5 OR

Circuit diagram of a parallel circuit with several make contacts:



Symbol in LOGO!:



The output status of the OR element is only 1 if **at least one** input is 1, i.e. at least one of the contacts is closed.

At an unused block input (x): $x = 0$.

OR function logic table

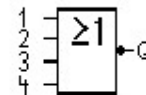
1	2	3	4	Q
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

4.2.6 NOR (not OR)

Circuit diagram of a series circuit with several break contacts:



Symbol in LOGO!:



The output status of the NOR is only 1 if **all** inputs are 0, i.e. if switched off. The NOR output is set to 0 when one of the inputs is switched on (logical 1 status).

At an unused block input (x): $x = 0$.

NOR function logic table

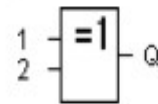
1	2	3	4	Q
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

4.2.7 XOR (exclusive OR)

The XOR in a circuit diagram, shown as series circuit with 2 changeover contacts:



Symbol in LOGO!



The output status of the XOR is 1 if the inputs are **not equivalent**.

At an unused block input (x): x = 0.

XOR function logic table

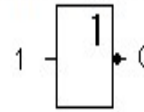
1	2	Q
0	0	0
0	1	1
1	0	1
1	1	0

4.2.8 NOT (Negation, Inverter)

A break contact in the circuit diagram:



Symbol in LOGO!:



The output status is 1 if the input is 0. The NOT block inverts the input status.

Advantage of the NOT block, for example: LOGOI does not require break contacts. You simply use a make contact and the NOT block to convert these into a break contact.

NOT function logic table

1	Q
0	1
1	0

4.3 Special functions

Because of their different input designation, you can see right away that there is a difference between the special functions and basic functions. SFs contain timer functions, retentive functions and various parameter assignment options, which allow you to adapt the circuit program to suit your own requirements.

This section provides you with a brief overview of input designations and with some particular background information on SFs. The SFs in particular are described in Chapter 4.4.

4.3.1 Designation of the inputs

Logical inputs

Here, you will find the description of the connectors you can use to create a logical link to other blocks or to the inputs of the LOGO! unit.

- **S (Set):**
A signal at input S sets the output to logical "1".
- **R (Reset):**
The reset input R takes priority over all other inputs and resets the outputs.
- **Trg (Trigger):**
This input is used to trigger the start of a function.
- **Cnt (Count):**
This input is used for counting pulses.
- **Fre (Frequency):**
Frequency signals to be evaluated are applied to this input.
- **Dir (Direction):**
This input determines the direction of count, for example.
- **En (Enable):**
This input enables a block function. When this input is "0", other signals to the block will be ignored.
- **Inv (Invert):**
A signal at this input inverts the output signal of the block.
- **Ral (Reset all):**
All internal values are reset.

Connector X at SF inputs

SF inputs connected to connector x are set low. That is, the inputs carry a 'lo' signal.

Parameter inputs

At some of the inputs you do not apply any signals. You configure the relevant block values instead. Examples:

- **Par (Parameter):**
This input will not be connected. Here, you set the relevant block parameters (times, on/off thresholds etc.).
- **No (Cam):**
This input will not be connected. Here, you configure the time patterns.
- **P (Priority):**
This is an open input. Here, you define priorities and specify whether a message is to be acknowledged in RUN.

4.3.2 Time response

Parameter T

At some of the SFs it is possible to configure a time value T. When you preset this time, note that your input values are based on the timebase set:

Timebase	-- : --
s (seconds)	seconds : $\frac{1}{100}$ seconds
m (minutes)	minutes : seconds
h (hours)	hours : minutes

B1	+	
T	=04:10h	

Setting a time T of 250 minutes:
 Unit in hours h:
 04:00 hours 240 minutes
 00:10 hours +10 minutes
 = 250 minutes

Note

Always specify a time T in ≥ 0.02 s. The time T is not defined for $T < 0.02$ s.

Accuracy of T

Because of slight tolerances in the characteristics of electronic components, the set time T may deviate. LOGO! has a maximum tolerance of ± 0.02 %.

When 0.02 % of the time T is less than 0.02 seconds, the maximum deviation is 0.02 seconds.

Example:

The maximum tolerance per hour (3600 seconds) is ± 0.02 %, which is proportional to ± 0.72 seconds.

The maximum tolerance per minute (60 seconds) is ± 0.02 seconds.

Accuracy of the timer (weekly/yearly timer)

To prevent timing inaccuracy of the real-time clock in C versions caused by this deviation, the timer value is continuously compared with a high-precision timebase and corrected. The resultant maximum timing inaccuracy is ± 5 s/day.

4.3.3 Backup of the real-time clock

Because the internal real-time clock of a LOGO! is backed up, it continues operation after a power failure. The ambient temperature influences the backup time. At an ambient temperature of 25 °C , the typical backup time is 80 hours.

If there is a power outage of a LOGO! for more than 80 hours, the internal clock responds, dependent on the equipment series, as shown below:

- Equipment series 0BA0:
On restarting, the clock is set to "Sunday 00:00 1 January". The time starts to run. Due to this, the system processes the time switches, which trigger actions if necessary.
- Equipment series 0BA1 and later:
On restarting, the clock is set to "Sunday 00:00 1 January". The time is stopped and flashes. LOGO! is back in the status that it was in before the power outage. In the RUN status, the system processes the counters that were parameterized with the time above. The clock is, however, still stopped.
- Equipment series 0BA6:
If you are using the optional LOGO! Battery card, or combined LOGO! Memory/Battery card, LOGO! can retain the clock time for up to two years. These cards are available for equipment series 0BA6.

4.3.4 Retentivity

The switching states and counter values of SFs can be set retentive. This means that current data are retained after a power failure, and that the block resumes operation at the break point. The timer is not reset, but resumes operation until the time-to-go has expired, for example.

To enable this response, however, the relevant functions must be set retentive. Two options are available:

R: The data is retained.

/: Current data is not retained (default). See the example on Page 84.

SFs hours counter, weekly timer, yearly timer and PI controller are always retentive.

4.3.5 Parameter protection

In the parameter protection settings, you can determine whether or not the parameters can be displayed and edited in LOGO! parameter assignment mode. Two options are available:

+: The parameter attribute permits read/write access in parameter assignment mode (default).

-: The parameter settings are read-/write-protected in parameter assignment mode, and can be edited only in programming mode. See the example on Page 84.

4.3.6 Calculating the gain and offset of analog values

A sensor is connected to the analog input and converts a process variable into an electrical signal. This value of signal lies within the typical range of this sensor.

LOGO! always converts the electrical signals at the analog input into digital values from 0 to 1000.

A voltage of 0 to 10 V at input AI is transformed internally into a range of values from 0 to 1000. An input voltage exceeding 10 V is shown as internal value 1000.

Because you cannot always process the range of values from 0 to 1000 as predetermined by LOGO!, you can multiply the digital values by a gain factor and then shift the zero of the range of values (offset). This allows you to output an analog value to the LOGO! display, which is proportional to the actual process variable.

Parameter	Minimum	Maximum
Input voltage (in V)	0	≥ 10
Internal value	0	1000
Gain	-10.00	+10.00
Offset	-10000	+10000

Mathematical rule

Actual value Ax =
(internal value at input Ax · gain) + offset

Gain and offset calculation

The gain and offset is calculated based on the relevant high and low values of the function.

Example 1:

The available thermocouples have the following technical data: -30 to +70°C, 0 to 10 V DC (i.e. 0 to 1000 in LOGO!).

Actual value = (internal value · gain) + offset, thus

$$-30 = (0 \cdot A) + B, \text{ i.e. offset } B = -30$$

$$+70 = (1000 \cdot A) - 30, \text{ i.e. gain } A = 0.1$$

Example 2:

A pressure sensor converts a pressure of 1000 mbar into a voltage of 0 V, and a pressure of 5000 mbar into a voltage of 10 V.

Actual value = (internal value · gain) + offset, thus

$$1000 = (0 \cdot A) + B, \text{ i.e. offset } B = 1000$$

$$5000 = (1000 \cdot A) + 1000, \text{ i.e. gain } A = 4$$

Example of analog values

Process variable	Voltage (V)	Internal value	Gain	Offset	Value shown (Ax)
-30° C	0	0	0.1	-30	-30
0° C	3	300	0.1	-30	0
+70° C	10	1000	0.1	-30	70
1000 mbar	0	0	4	1000	1000
3700 mbar	6.75	675	4	1000	3700
5000 mbar	10	1000	4	1000	5000
	0	0	0.01	0	0
	5	500	0.01	0	5
	10	1000	0.01	0	10
	0	0	1	0	0
	5	500	1	0	500
	10	1000	1	0	1000
	0	0	10	0	0
	5	500	10	0	5000
	10	1000	10	0	10000
	0	0	0.01	5	5
	5	500	0.01	5	10
	10	1000	0.01	5	15
	0	0	1	500	500
	5	500	1	500	1000
	10	1000	1	500	1500
	0	0	1	-200	-200
	5	500	1	-200	300
	10	1000	1	-200	800
	0	0	10	-10000	-10000
	10	1000	10	-10000	0
	0.02	2	0.01	0	0
	0.02	2	0.1	0	0
	0.02	2	1	0	2
	0.02	2	10	0	20

A sample application is found in the description of the "Analog comparator" SF on Page 165.

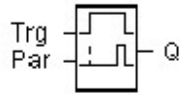
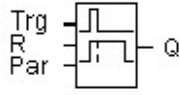
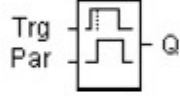
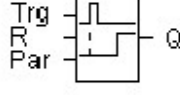
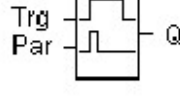
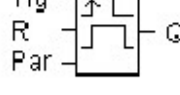
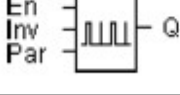
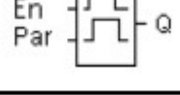
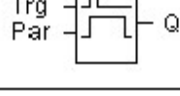
For further information on analog inputs, refer to Chapter 4.1.

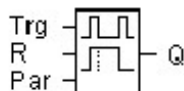
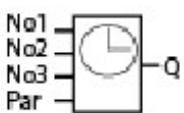
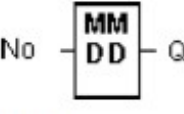
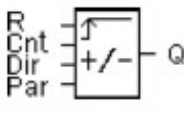
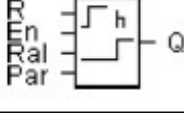
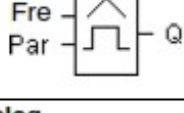
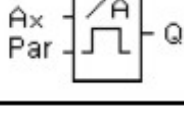


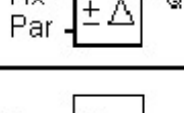
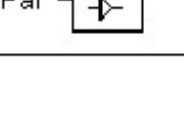
4.4 Special functions list - SF

When you create your circuit program in LOGO!, you find the special function blocks in the SF list.

You can invert the inputs of SFs individually, that is, the circuit program converts a logical "1" at the input into a logical "0"; a logical "0" it converts into a logical "1". An example of the program code is found in Chapter 3.7.3.

The table also specifies whether the relevant function can be set retentive (Rem). The following SFs are available:

View in LOGO!	Name of the special function	Rem
Times		
	On-delay (see page 120)	REM
	Off-delay (see page 123)	REM
	On-/Off-delay (see Page 124)	REM
	Retentive on-delay (see page 126)	REM
	Wiping relay (pulse output) (see page 127)	REM
	Edge triggered wiping relay (see page 128)	REM
	Asynchronous pulse generator (see Page 130)	REM
	Random generator (see page 132)	
	Stairway lighting switch (see page 134)	REM

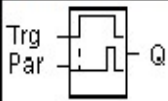
View in LOGO!	Name of the special function	Rem
	Multiple function switch (see page 136)	REM
	Weekly timer (see Page 139)	
	Yearly timer (see Page 143)	
Counter		
	Up/down counter (see Page 149)	REM
	Hours counter (see page 152)	REM
	Threshold trigger (see Page 156)	
Analog		
	Analog threshold trigger (see page 159)	
	Analog differential trigger (see page 162)	
	Analog comparator (see Page 165)	
	Analog value monitoring (see Page 170)	
	Analog amplifier (see Page 173)	

View in LOGO!	Name of the special function	Rem
	Analog multiplexer (see Page 194)	
	Pulse Width Modulator (PWM) (see Page 206)	
	Analog math (see Page 209)	
	Analog ramp (see Page 196)	
	PI controller (see Page 200)	REM
Miscellaneous		
	Latching relay (see Page 174)	REM
	Pulse relay (see Page 175)	REM
	Message texts (see Page 177)	
	Softkey (see Page 189)	REM
	Shift register (see Page 192)	REM
	Analog math error detection (see Page 212)	

4.4.1 On-delay

Short description

The output is only set after a configurable on-delay time has expired.

Symbol in LOGO!	Wiring	Description
	Input Trg	A signal at input Trg (Trigger) triggers the on-delay timer.
	Parameter	T represents the time after which the output is switched on (0 to 1 transition of the output signal). Retentivity: / = no retentivity R = the status is retentive.
	Output Q	Q is switched on when the set time T has expired, provided Trg is still set.

Parameter T

Note the defaults for parameter T in Chapter 4.3.2.

The time for parameter T can also be preset based on the actual value of another, already-configured function. You can use the actual values of the following functions:

- Analog comparator (actual value Ax - Ay, see Chapter 4.4.18)
- Analog threshold trigger (actual value Ax, see Chapter 4.4.16)
- Analog amplifier (actual value Ax, see Chapter 4.4.20)
- Analog multiplexer (actual value AQ, see Chapter 4.4.26)
- Analog ramp (actual value AQ, see Chapter 4.4.27)
- Analog math (actual value AQ, see Chapter 4.4.30)
- PI controller (actual value AQ, see Chapter 4.4.28)
- Counter (actual value Cnt, see Chapter 4.4.13)

Select the required function by the block number. The timebase is adjustable.

Please note the following listing:

Valid ranges of the timebase, if T = parameter

Timebase	max. value	min. resolution	Accuracy
s (seconds)	99:99	10 ms	+ 10 ms
m (minutes)	99:59	1s	+ 1 s
h (hours)	99:59	1 min	+ 1 min

The display in programming mode (example):

```

B12      +R
T  =04:10h
    
```

Valid ranges of the timebase, if T = Actual value of an already-programmed function

Timebase	max. value	Meaning	Accuracy
ms	99990	Number of ms	+ 10 ms
s	5999	Number of s	+ 1 s
m	5999	Number of min	+ 1 min

The display in programming mode (example):

```

B12      +R
T  →B006s
    
```

If the referenced block (B6, in the example) returns a value that lies out of the valid range, the value is rounded up or down to the next valid value.

Parameter preset = Actual value of an already-programmed function

How to include the actual value of an already-programmed function:

1. Press ► to move the cursor to the equal sign of parameter T.

```

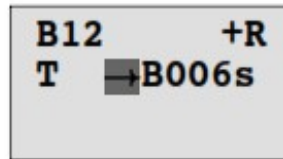
B12      +R
T  =04:10h
    
```

Press ► twice

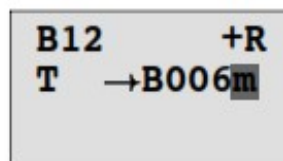
```

B12      +R
T  ▣=04:10h
    
```

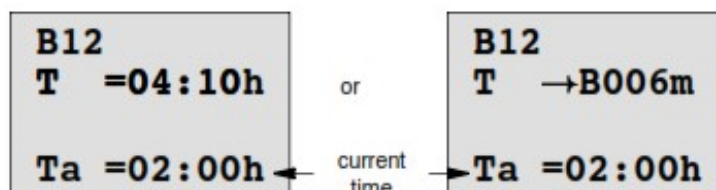

- Press ▼ to change the equal sign into an arrow. If it exists, the last referenced block and its timebase is shown.



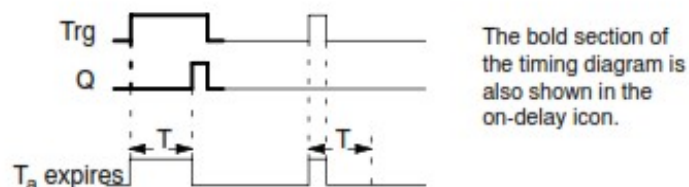
- Press ► to move the cursor to the "B" of the shown block, and then press ▼ to select the required block number.
- Press ► to move the cursor to the block's timebase and press ▼ to select the required timebase.



The view in parameter assignment mode (example):



Timing diagram



Functional description

The time T_a is triggered with a 0 to 1 transition at input Trg (T_a is the current LOGO! time).

If the status of input Trg is 1 at least for the duration of the configured time T, the output is set to 1 on expiration of this time (the output follows the input with on-delay).

The time is reset when the status at input Trg returns to 0 before the time T has expired.

The output is reset to 0 when the signal at input Trg is 0.

If retentivity is not set, output Q and the expired time are reset after a power failure.

4.4.2 Off-delay

Short description

When an on-delay is set, the output is reset when the configured time has expired.

Symbol in LOGO!	Wiring	Description
	Input Trg	The off-delay timer starts with a negative edge (1 to 0 transition) at input Trg (Trigger)
	Input R	A signal at input R resets the on-delay time and the output.
	Parameter	The output switches off (transitions from 1 to 0) when the delay time T expires. Retentivity: / = No retentivity R = The status is retentive.
	Output Q	Q is set with a signal at input Trg. It holds this state until T has expired.

Parameter T

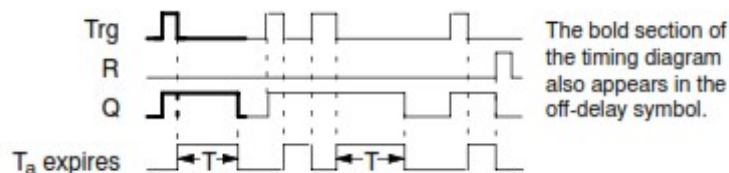
Note the parameter T defaults specified in Chapter 4.3.2.

The time for parameter T can be based on the actual value of another, already-configured function. You can use the actual value of the following functions:

- Analog comparator (actual value Ax - Ay, see Chapter 4.4.18)
- Analog threshold trigger (actual value Ax, see Chapter 4.4.16)
- Analog amplifier (actual value Ax, see Chapter 4.4.20)
- Analog multiplexer (actual value AQ, see Chapter 4.4.26)
- Analog ramp (actual value AQ, see Chapter 4.4.27)
- Analog math (actual value AQ, see Chapter 4.4.30)
- PI controller (actual value AQ, see Chapter 4.4.28)
- Up/down counter (actual value Cnt, see Chapter 4.4.13)

Select the required function by the block number. The timebase is configurable. For information on valid timebase ranges and parameter preset, refer to Chapter 4.4.1.

Timing diagram



Functional description

Output Q is set to hi immediately when the input Trg changes to hi.

The actual time T_a in LOGO! is retriggered at the 1 to 0 transition of Trg. The output remains set. Output Q is reset to 0 with off-delay when T_a reaches the value configured at T ($T_a=T$).

The time T_a is retriggered with a one-shot at input Trg.

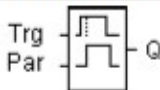
You can set input R (Reset) to reset the time T_a and the output before T_a has expired.

If retentivity is not set, output Q and the expired time are reset after a power failure.

4.4.3 On-/Off-delay

Short description

The on-/off-delay function sets the output after the set on-delay time has expired, and resets it upon expiration of the off-delay time.

Symbol in LOGO!	Wiring	Description
	Input Trg	A positive edge (0 to 1 transition) at input Trg (Trigger) triggers the on-delay time T_H . A negative edge (1 to 0 transition) at input Trg (Trigger) triggers the off-delay time T_L .
	Parameter	T_H is the time after which the output is set hi (output signal transition 0 to 1). T_L is the time after which the output is reset (output signal transition 1 to 0). Retentivity: / = No retentivity R = The status is retentive.
	Output Q	Q is set when the configured time T_H has expired and Trg is still set. It is reset on expiration of the time T_L , if the trigger Trg has not been set again.

Parameters T_H and T_L

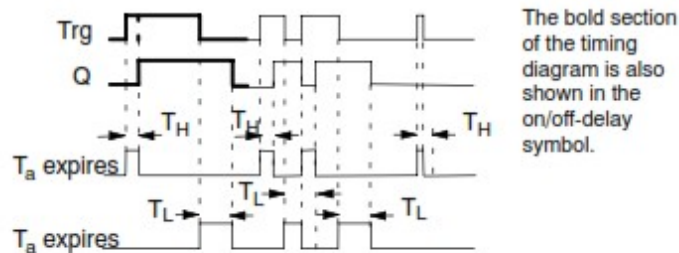
Note the preset values for the parameters T_H and T_L in Chapter 4.3.2.

The on-delay and off-delay times for parameters T_H and T_L can be based on the actual value of another, already-configured function. You can use the actual values of the following functions:

- Analog comparator (actual value $A_x - A_y$, see Chapter 4.4.18)
- Analog threshold trigger (actual value A_x , see Chapter 4.4.16)
- Analog amplifier (actual value A_x , see Chapter 4.4.20)
- Analog multiplexer (actual value AQ , see Chapter 4.4.26)
- Analog ramp (actual value AQ , see Chapter 4.4.27)
- Analog math (actual value AQ , see Chapter 4.4.30)
- PI controller (actual value AQ , see Chapter 4.4.28)
- Up/down counter (actual value Cnt , see Chapter 4.4.13)

Select the required function by the block number. The timebase is configurable. For information on valid timebase ranges and parameter preset, refer to Chapter 4.4.1.

Timing diagram



Functional description

The time T_H is triggered with a 0 to 1 transition at input Trg .

If the status at input Trg is 1 at least for the duration of the time T_H , the output is set to 1 on expiration of the time T_H (the output follows the input with on-delay).

The time is reset when the signal at input Trg is reset to 0 before the time T_H has expired..

A 1 to 0 transition at input Trg triggers the time T_L .

If the status at input Trg is 0 at least for the duration of the signal T_L , the output is set to 0 on expiration of the time T_L (the output follows the input with off-delay).

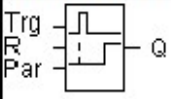
The time is reset when the signal at input Trg changes to 1 again before the time T_L has expired.

If retentivity is not set, output Q and the expired time are reset after a power failure.

4.4.4 Retentive on-delay

Short description

A one-shot at the input triggers a configurable on-delay time. The output is set when this time has expired.

Symbol in LOGO!	Wiring	Description
	Input Trg	A signal at input Trg (Trigger) triggers the on-delay timer.
	Input R	A signal at input R resets the on-delay time and the output.
	Parameter	T represents the on-delay time for the output (output status transition 0 to 1). Retentivity: / = no retentivity R = The status is retentive.
	Output Q	Q is set after the time T has expired.

Parameter T

Note the defaults specified in Chapter 4.3.2.

The time for parameter T can be provided by the actual value of another already-programmed function. You can use the actual values of the following functions:

- Analog comparator (actual value Ax - Ay, see Chapter 4.4.18)
- Analog threshold trigger (actual value Ax, see Chapter 4.4.16)
- Analog amplifier (actual value Ax, see Chapter 4.4.20)
- Analog multiplexer (actual value AQ, see Chapter 4.4.26)
- Analog ramp (actual value AQ, see Chapter 4.4.27)
- Analog math (actual value AQ, see Chapter 4.4.30)
- PI controller (actual value AQ, see Chapter 4.4.28)
- Up/down counter (actual value Cnt, see Chapter 4.4.13)

Select the required function by the block number. The timebase is configurable. For information on valid ranges and parameter defaults, refer to Chapter 4.4.1.

Timing diagram



The bold section of the timing diagram is also shown in the symbol of the retentive on-delay.

Functional description

The 0 to 1 signal transition at input Trg triggers the current time T_a . Output Q is set when $T_a = T$. A further signal at input Trg does not influence the time T_a .

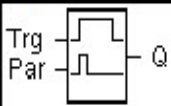
The output and the time T_a are reset with the next 1 signal at input R.

If retentivity is not set, output Q and the expired time are reset after a power failure.

4.4.5 Wiping relay (pulse output)

Short description

An input pulse generates a signal with a configurable period at the output.

Symbol in LOGO!	Wiring	Description
	Input Trg	A signal at input Trg (Trigger) triggers the time for the wiping relay function.
	Parameter	The output is switched off after the time T has expired (output signal transition 1 to 0). Retentivity: / = No retentivity R = The status is retentive.
	Output Q	A signal at input Trg sets Q. If the input signal = 1, output Q remains set for the time T_a .

Parameter T

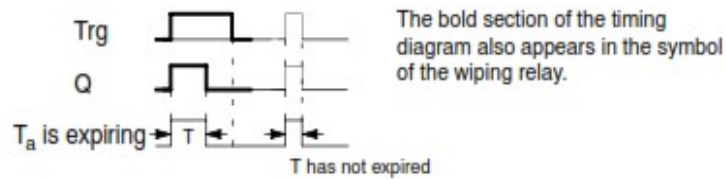
Note the information on parameter T in Chapter 4.3.2.

The time for parameter T can be provided by the actual value of another already-programmed function. You can use the actual values of the following functions:

- Analog comparator (actual value $A_x - A_y$, see Chapter 4.4.18)
- Analog threshold trigger (actual value A_x , see Chapter 4.4.16)
- Analog amplifier (actual value A_x , see Chapter 4.4.20)
- Analog multiplexer (actual value AQ , see Chapter 4.4.26)
- Analog ramp (actual value AQ , see Chapter 4.4.27)
- Analog math (actual value AQ , see Chapter 4.4.30)
- PI controller (actual value AQ , see Chapter 4.4.28)
- Up/down counter (actual value Cnt , see Chapter 4.4.13)

Select the required function by the block number. The timebase is configurable. For information on valid ranges and parameter defaults, refer to Chapter 4.4.1.

Timing diagram



Functional description

A 0 to 1 transition at input Trg sets the output, and triggers a time T_a during which the output remains set.

Output Q is reset to 0 (pulse output) when T_a reaches the value preset at T ($T_a = T$).

The output is immediately reset if there is a 1 to 0 transition at input Trg before the specified time has expired.

If retentivity is not set, output Q and the expired time are reset after a power failure.

4.4.6 Edge triggered wiping relay

Short description

An input pulse generates a preset number of output pulses with a defined pulse/pause ratio (retriggerable), after a configured delay time has expired.

Symbol in LOGO!	Wiring	Description
	Input Trg	A signal at input Trg (Trigger) triggers the times for the edge triggered wiping relay.
	Input R	A signal at input R resets the current time (T_a) and the output.
	Parameter	<p>The interpulse width T_L and the pulse width T_H are configurable.</p> <p>N determines the number of pulse/pause cycles T_L/T_H: Range of values: 1...9</p> <p>Retentivity: / = No retentivity R = The status is retentive.</p>
	Output Q	Q is set after T_L has expired, and reset after T_H has expired.

Parameters TH and TL

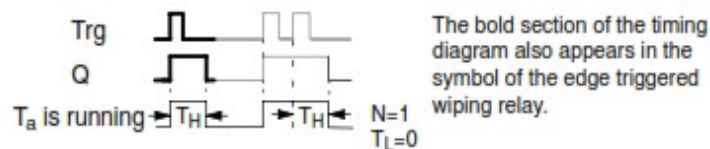
Note the information on parameter T in Chapter 4.3.2.

The pulse width TH and the interpulse width TL can be provided by the actual value of another already-programmed function. You can use the actual values of the following functions:

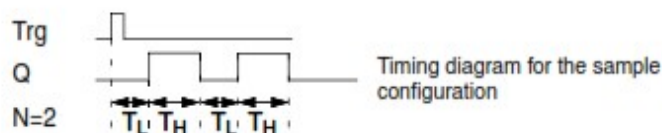
- Analog comparator (actual value Ax - Ay, see Chapter 4.4.18)
- Analog threshold trigger (actual value Ax, see Chapter 4.4.16)
- Analog amplifier (actual value Ax, see Chapter 4.4.20)
- Analog multiplexer (actual value AQ, see Chapter 4.4.26)
- Analog ramp (actual value AQ, see Chapter 4.4.27)
- Analog math (actual value AQ, see Chapter 4.4.30)
- PI controller (actual value AQ, see Chapter 4.4.28)
- Up/down counter (actual value Cnt, see Chapter 4.4.13)

Select the required function by the block number. The timebase is configurable. For information on valid ranges and parameter defaults, refer to Chapter 4.4.1.

Timing diagram A



Timing diagram B



Functional description

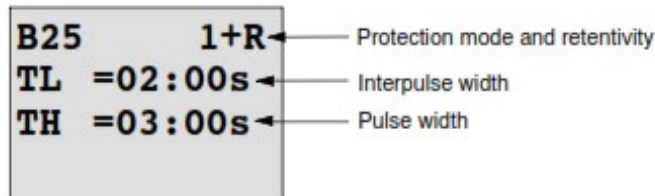
A 0 to 1 transition at input Trg triggers the time T_L (Time Low). After the time T_L has expired, output Q is set for the duration of T_H (Time High).

If there is a further 0 to 1 transition (retriggering pulse) at input Trg before the preset time ($T_L + T_H$) has expired, T_a is reset and the pulse/pause cycle is restarted.

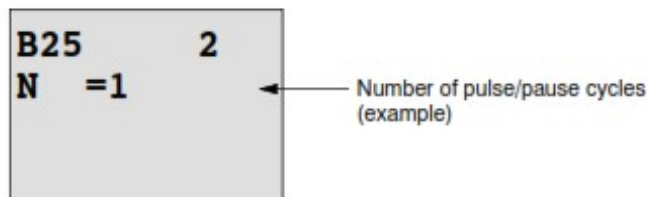
If retentivity is not set, output Q and the time are reset after a power failure.

Setting the Par parameter

View in programming mode (example):



Press ►



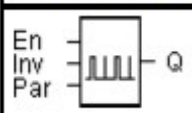
View in parameter assignment mode (example):



4.4.7 Asynchronous pulse generator

Short description

The output pulse shape can be modified by reconfiguring the pulse/pause ratio.

Symbol in LOGO!	Wiring	Description
	Input En	You can use input EN to set and reset the asynchronous pulse generator.
	Input INV	Input INV can be used to invert the output signal of the active asynchronous pulse generator.
	Parameter	You can configure the pulse width T_H and the interpulse width T_L . Retentivity: / = No retentivity R = The status is retentive.
	Output Q	Q is set and reset cyclically according to the pulse/pause ratio T_H and T_L .

Parameters TH and TL

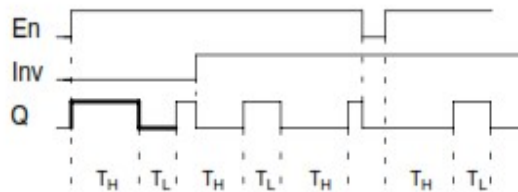
Note the information on parameter T in Chapter 4.3.2.

The pulse width TH and the interpulse width TL can be provided by the actual value of another already-programmed function. You can use the actual values of the following functions:

- Analog comparator (actual value Ax - Ay, see Chapter 4.4.18)
- Analog threshold trigger (actual value Ax, see Chapter 4.4.16)
- Analog amplifier (actual value Ax, see Chapter 4.4.20)
- Analog multiplexer (actual value AQ, see Chapter 4.4.26)
- Analog ramp (actual value AQ, see Chapter 4.4.27)
- Analog math (actual value AQ, see Chapter 4.4.30)
- PI controller (actual value AQ, see Chapter 4.4.28)
- Up/down counter (actual value Cnt, see Chapter 4.4.13)

Select the required function by the block number. The timebase is configurable. For information on valid ranges and parameter defaults, refer to Chapter 4.4.1.

Timing diagram



Functional description

You can configure the pulse/interpulse width at the TH (Time High) and TL (Time Low) parameters.

Input Inv can be used to invert the output signal, provided the block is enabled with a signal at input EN.

If retentivity is not set, output Q and the expired time are reset after a power failure.

4.4.8 Random generator

Short description

The output of the random generator is set or reset within a configured time.

Symbol in LOGO!	Wiring	Description
	Input En	<p>A positive edge (0 to 1 transition) at input En (Enable) triggers the on-delay time of the random generator.</p> <p>A negative edge (1 to 0 transition) at input En (Enable) triggers the off-delay time of the random generator.</p>
	Parameter	<p>The on-delay is set at random to a value between 0 s and T_H.</p> <p>The off-delay is set at random to a value between 0 s and T_L.</p>
	Output Q	Output Q is set when the on-delay has expired and if En is still set. It is reset when the off-delay has expired, provided En was not set again meanwhile.

Parameter T_H and T_L

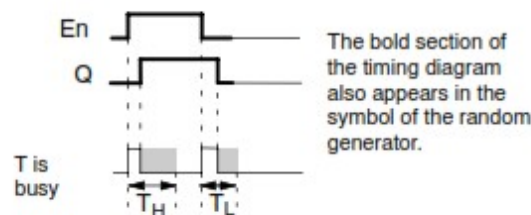
Note the defaults of the T_H and T_L parameters listed in Chapter 4.3.2.

The on-delay time T_H and the off-delay time T_L can be provided by the actual value of another already-programmed function. You can use the actual values of the following functions:

- Analog comparator (actual value $A_x - A_y$, see Chapter 4.4.18)
- Analog threshold trigger (actual value A_x , see Chapter 4.4.16)
- Analog amplifier (actual value A_x , see Chapter 4.4.20)
- Analog multiplexer (actual value A_Q , see Chapter 4.4.26)
- Analog ramp (actual value A_Q , see Chapter 4.4.27)
- Analog math (actual value A_Q , see Chapter 4.4.30)
- PI controller (actual value A_Q , see Chapter 4.4.28)
- Up/down counter (actual value Cnt , see Chapter 4.4.13)

Select the required function by the block number. The timebase is configurable. For information on valid ranges and parameter defaults, refer to Chapter 4.4.1.

Timing diagram



Functional description

The 0 to 1 transition at input En triggers a random on-delay time between 0 s and T_H . The output is set when the on-delay time has expired and if the signal at input En remains hi at least for the duration of this time.

The time is reset if input En is reset before the on-delay time has expired.

A 1 to 0 transition at input EN triggers a random off-delay time between 0 s and T_L .

The output is reset after the off-delay time has expired, provided input En remains lo at least for the duration of this time.

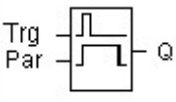
The time is reset if the signal at input En changes to 1 again before the off-delay time has expired.

The time expired is reset after a power failure.

4.4.9 Stairway lighting switch

Short description

An input edge triggers a configurable and retriggerable time. The output is reset after this time has expired. A warning signal can be output before this time has expired to warn of the impending shutdown.

Symbol in LOGO!	Wiring	Description
	Input Trg	A signal at input Trg (Trigger) triggers the off-delay time for the stairway lighting switch.
	Parameter	<p>T represents the off-delay time of the output (output signal transition 1 to 0).</p> <p>T_I determines the triggering time for the pre-warning.</p> <p>T_{IL} determines the length of the pre-warning signal.</p> <p>Retentivity: / = No retentivity R = The status is retentive.</p>
	Output Q	Q is reset after the time T has expired. A warning signal can be output before this time has expired.

Parameter T, T_I and T_{IL}

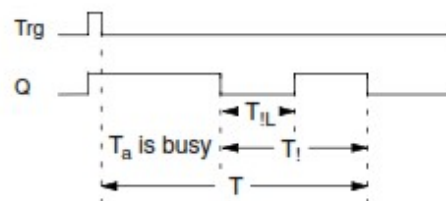
Note the defaults of the T parameters listed in Chapter 4.3.2.

The off-delay time T, the pre-warning time T_I and the pre-warning period T_{IL} can be provided by the actual value of another already-programmed function. You can use the actual values of the following functions:

- Analog comparator (actual value Ax - Ay, see Chapter 4.4.18)
- Analog threshold trigger (actual value Ax, see Chapter 4.4.16)
- Analog amplifier (actual value Ax, see Chapter 4.4.20)
- Analog multiplexer (actual value AQ, see Chapter 4.4.26)
- Analog ramp (actual value AQ, see Chapter 4.4.27)
- Analog math (actual value AQ, see Chapter 4.4.30)
- PI controller (actual value AQ, see Chapter 4.4.28)
- Up/down counter (actual value Cnt, see Chapter 4.4.13)

Select the required function by the block number. The timebase is configurable. For information on valid ranges and parameter defaults, refer to Chapter 4.4.1.

Timing diagram



Functional description

A 0 to 1 signal transition at input Trg sets output Q. The next 1 to 0 transition at Trg retriggers the current time T_a , and output Q remains set.

Output Q is reset when $T_a = T$. You can output a warning signal before the off-delay time $(T - T_l)$ has expired to reset Q for the time of the pre-warning period T_{lL} .

A further one-shot at input Trg during T_a retriggers the time T_a .

If retentivity is not set, output Q and the expired time are reset after a power failure.

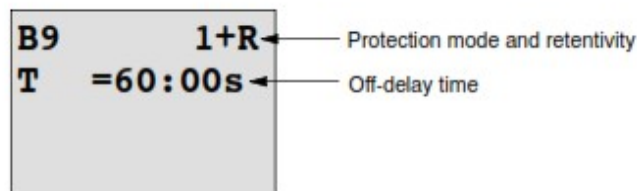
Setting the Par parameter

Note the defaults specified in Chapter 4.3.2.

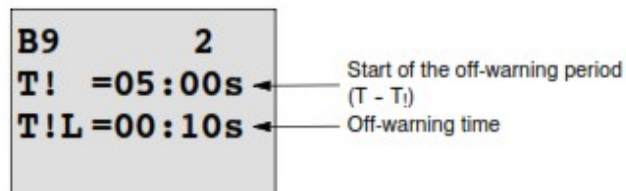
Note

All times must have the same timebase.

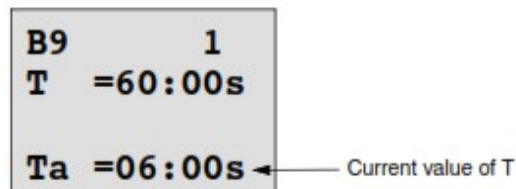
View in programming mode (example):



Press ►



View in parameter assignment mode (example):

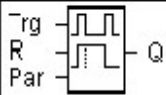


4.4.10 Multiple function switch

Short description

Switch with two different functions:

- Pulse switch with off-delay
- Switch (permanent lighting)

Symbol in LOGO!	Wiring	Description
	Input Trg	A signal at input Trg (Trigger) sets output Q (permanent light) or resets Q with an off-delay. When active, output Q can be reset with a signal at input Trg.
	Input R	A signal at input R resets the current time T_a and resets the output.
	Parameter	<p>T represents the off-delay time. The output is reset (1 to 0 transition) when time T expires.</p> <p>T_L represents the time during which the output must be set to enable the permanent light function.</p> <p>T_I represents the on-delay for the prewarning time.</p> <p>T_{IL} represents the length of the prewarning time period.</p> <p>Retentivity: / = No retentivity R = The status is retentive.</p>
	Output Q	A signal at Trg switches on output Q. Depending on the length of the input at Trg, the output is switched off again or switched on permanently, or it is reset with a further signal at Trg.

Parameters T, T_L , T_I and T_{IL}

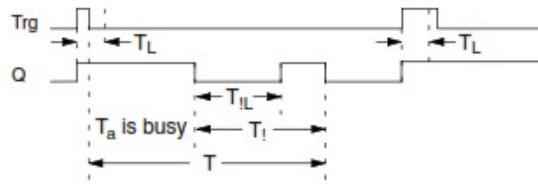
Note the defaults of the T parameters listed in Chapter 4.3.2.

The off-delay time T, the permanent light time T_L , the on-delay prewarning time T_I and the prewarning time period T_{IL} can be provided by the actual value of another already-programmed function. You can use the actual values of the following functions:

- Analog comparator (actual value Ax - Ay, see Chapter 4.4.18)
- Analog threshold trigger (actual value Ax, see Chapter 4.4.16)
- Analog amplifier (actual value Ax, see Chapter 4.4.20)
- Analog multiplexer (actual value AQ, see Chapter 4.4.26)
- Analog ramp (actual value AQ, see Chapter 4.4.27)
- Analog math (actual value AQ, see Chapter 4.4.30)
- PI controller (actual value AQ, see Chapter 4.4.28)
- Up/down counter (actual value Cnt, see Chapter 4.4.13)

Select the required function by the block number. The timebase is configurable. For information on valid ranges and parameter defaults, refer to Chapter 4.4.1.

Timing diagram



Functional description

A 0 to 1 transition at input Trg sets output Q.

If output $Q = 0$, and input Trg is set hi at least for the duration of T_L , the permanent lighting function is enabled and output Q is set accordingly.

The off-delay T is triggered when input Trg returns to 0 before T_L has expired.

Output Q is reset when $T_a = T$.

You can output an off-warning signal prior to the expiration of the off-delay time ($T - T_I$) that resets Q for the duration of the prewarning time period T_{IL} . A subsequent signal at Trg always resets T and the output Q.

If retentivity is not set, output Q and the expired time are reset after a power failure.

Setting the Par parameter

Note the defaults specified in Chapter 4.3.2.

Note

T, T_I and T_{IL} must all have the same timebase.

View in programming mode (example):

B5	1+R	← Protection mode and retentivity
T	=60:00s	← Off-delay
TL	=10:00s	← Permanent light on-time

Press ►

B5	2	
T!	=30:00s	← Start of the off-warning period (T - T _I)
T!L	=20:00s	← Off-warning time

View in parameter assignment mode (example):

B5	1	
T	=60:00s	
TL	=10:00s	
Ta	=06:00s	← Current value of the time T _L or T

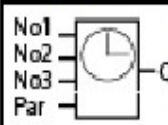
4.4.11 Weekly timer

Short description

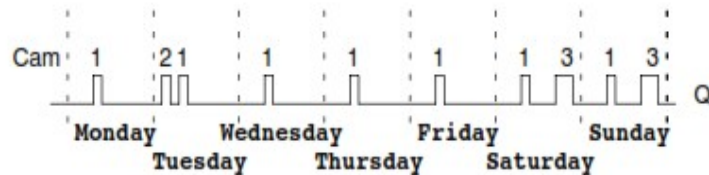
The output is controlled by means of a configurable on/off date. The function supports any combination of weekdays. You select the active weekdays by hiding the inactive days.

Note

Because the LOGO! 24/24o does not have a real-time clock, the weekly timer function is not available for this version.

Symbol in LOGO!	Wiring	Description
	Cam parameters 1, 2, and 3	At the Cam parameters, you set the on- and off-times of the weekly timer for each Cam switch. Here you also configure the days and the time-of-day.
	Par	You specify whether the timer pulses on for one cycle when activated and then reset. The pulse setting applies to all three cams.
	Output Q	Q is set when the configured cam is actuated.

Timing diagram (three examples)



Cam 1:	Daily:	06:30 h to 8:00 h
Cam 2:	Tuesday:	03:10 h to 04:15 h
Cam 3:	Saturday and Sunday:	16:30 h to 23:10 h

Functional description

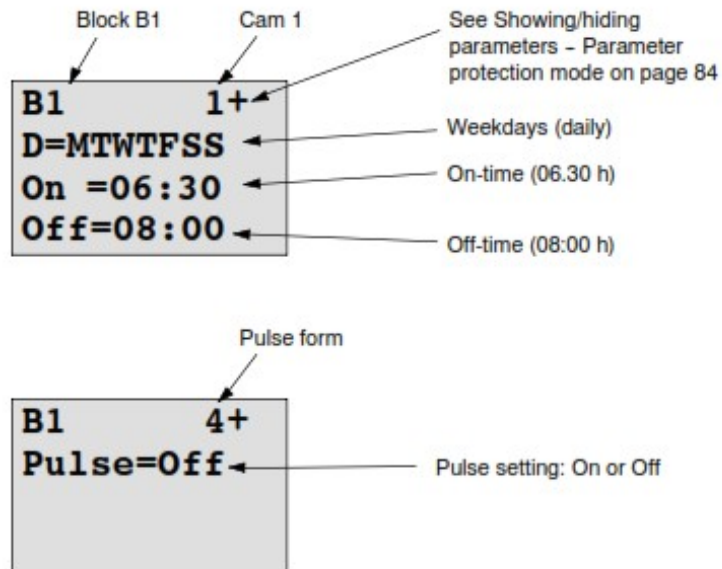
Each weekly timer has three cams you can use to configure a time hysteresis. You specify the on- and off-times at the Cam parameters. The weekly timer sets the output at a certain on-time, if this is not already set.

The weekly timer resets the output at a certain off-time if you configured an off-time, or at the end of the cycle if you specified a pulse output. You will cause a conflict if the on- and off-times you set for the weekly timer are identical, though set on different cams. In this case, cam3 takes priority over cam2, whereas cam2 takes priority over cam1.

The switching state of the weekly timer is determined by the status of all three cams.

Parameter assignment screen form

View of the parameter assignment screen form, for example for Cam1 and the Pulse setting:



Weekday

The suffix of "D=" (Day) has the following meaning:

- M : Monday
- T : Tuesday
- W : Wednesday
- T : Thursday
- F : Friday
- S : Saturday
- S : Sunday

The uppercase letter indicates: A weekday is selected. A "-" means: A weekday has not been selected.

On-/Off-times

Any time between 00:00 h and 23:59 h is possible. You can also configure the on time to be a pulse signal. The timer block will be activated at the specified time for one cycle and then the output is reset.

--:-- means: No on-/off-times set.

Setting the weekly timer

To set the on-/off-times:

1. Move the cursor to one of the Cam parameters of the timer (e.g. No1).
2. Press **OK**. LOGO! opens the Cam parameter assignment screen form. The cursor is positioned on the weekday.
3. Press **▲** and **▼** to select one or several weekdays.
4. Press **▶** to move the cursor to the first position of the on-time.
5. Set the on-time.
Modify the value at the respective position, using the keys **▲** and **▼**. Move to the cursor to the various positions, using the keys **◀** and **▶**. At the first position, you can only select the value --:--
(--:-- means: No on-/off-times set).
6. Press **▶** to move the cursor to the first position of the off-time.
7. Set the off-time (in same way as in step 5).
8. Confirm your entries with **OK**.

The cursor is now positioned on the No2 parameter (Cam2) and you can configure a further cam.

Note

For information on timer accuracy, refer to the technical data and to Chapter 4.3.2.

Weekly timer: Example

The output of the weekly timer switch is to be set daily from 06:30 h to 08:00 h. The output should also be set every Tuesday from 03:10 h to 04:15 h, and on the weekends from 16:30 h to 23:10 h.

This requires three cams.

Here are the parameter assignment screen forms of the cams No 1, 2 and 3, based on the timing diagram shown earlier.

Cam1

Cam No1 must set the output of the weekly timer daily from 06:30 h to 08:00 h.

```
B1      1+
D=MTWTFSS
On =06:30
Off=08:00
```

Cam2

Cam No2 must set the output of the weekly timer every Tuesday from 03:10 h to 04:15 h.

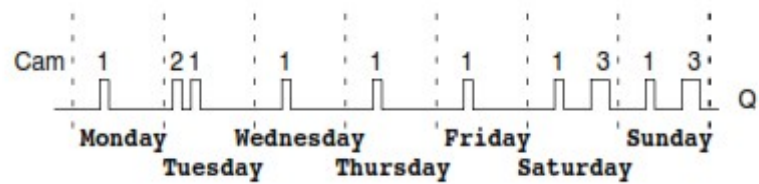
```
B1      2
D=-T-----
On =03:10
Off=04:15
```

Cam3

Cam No3 must set the output of the weekly timer switch every Saturday and Sunday from 16:30 h to 23:10 h.

```
B1      3
D=-----SS
On =16:30
Off=23:10
```

Result




4.4.12 Yearly timer

Short description

The output is controlled by means of a configurable on/off date. You can configure the timer to activate on a yearly, monthly, or user-defined time basis. With any mode, you can also configure the timer to pulse the output during the defined time period. The time period is configurable within the date range of January 1, 2000 to December 31, 2099

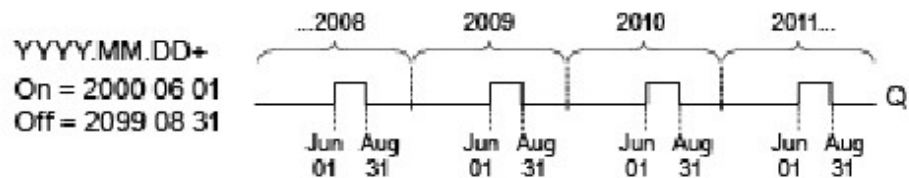
Note

Because LOGO! 24/24o does not have a real-time clock, the yearly timer is not available for this version.

Symbol in LOGO!	Wiring	Description
No 	Cam parameter	At the Cam parameter, you configure the timer mode, the on-/off-times for the timer, and whether the output is a pulse output.
	Output Q	Q is set when the configured cam is switched on.

Timing diagrams

Example 1: Yearly mode on, Monthly mode off, Pulse Off, On Time = 2000-06-01, Off Time = 2099-08-31: Every year on June 1 the timer output switches on and remains on until August 31.

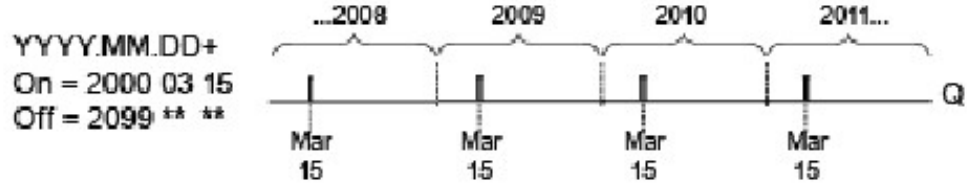


B6	1+
Yearly = On	
Monthly=Off	
Pulse =Off	

B6	2+
ON :	
YYYY-MM-DD	
2000-06-01	

B6	3+
OFF :	
YYYY-MM-DD	
2099-08-31	

Example 2: Yearly mode on, Monthly mode off, Pulse on, On Time = 2000-03-15, Off Time = 2099-**-**: Every year on March 15, the timer switches on for one cycle.

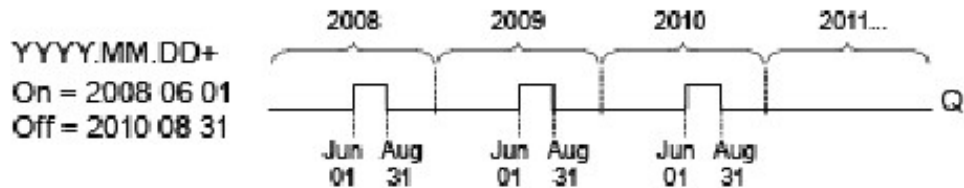


B6 1+
 Yearly = On
 Monthly=Off
 Pulse = On

B6 2+
 ON :
 YYYY-MM-DD
 2000-03-15

B6 3+
 OFF :
 YYYY-MM-DD
 2099-**-**

Example 3: Yearly mode on, Monthly mode off, Pulse off, On Time = 2008-06-01, Off Time = 2010-08-31: On June 1 of 2008, 2009, and 2010 the timer output switches on and remains on until August 31.

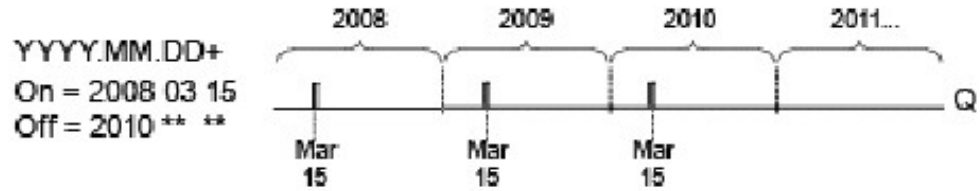


B6 1+
 Yearly = On
 Monthly=Off
 Pulse =Off

B6 2+
 ON :
 YYYY-MM-DD
 2008-06-01

B6 3+
 OFF :
 YYYY-MM-DD
 2010-08-31

Example 4: Yearly mode on, Monthly mode off, Pulse on, On Time = 2008-03-15, Off Time = 2010-**-**-**: On March 15 of 2008, 2009, and 2010, the timer output switches on for one cycle.

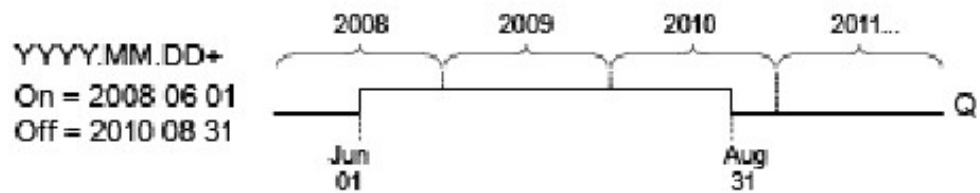


B6 1+
 Yearly = On
 Monthly=Off
 Pulse = On

B6 2+
 ON :
 YYYY-MM-DD
 2008-03-15

B6 3+
 OFF :
 YYYY-MM-DD
 2010-**-**-

Example 5: Yearly mode off, Monthly mode off, Pulse off, On Time = 2008-06-01, Off Time = 2008-08-31: On June 1, 2008 the timer output switches on and remains on until August 31, 2010.

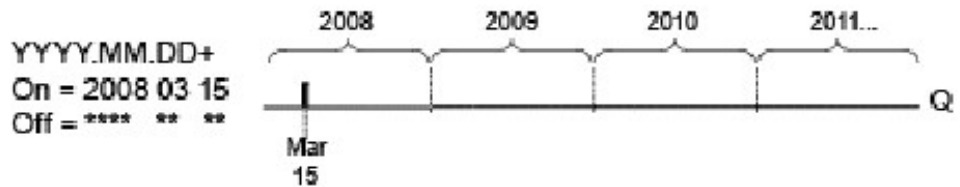


B6 1+
 Yearly =Off
 Monthly=Off
 Pulse =Off

B6 2+
 ON :
 YYYY-MM-DD
 2008-06-01

B6 3+
 OFF :
 YYYY-MM-DD
 2010-08-31

Example 6: Yearly mode off, Monthly mode off, Pulse selected, On Time = 2008-03-15, Off Time = ****-**-**: On March 15, 2008 the timer output switches on for one cycle. Because the timer does not have a monthly action or yearly action, the timer output pulses only one time at the specified On Time.

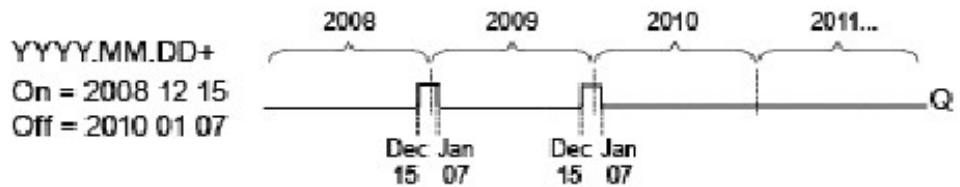


B6	1+
Yearly =Off	
Monthly=Off	
Pulse =On	

B6	2+
ON :	
YYYY-MM-DD	
2008-03-15	

B6	3+
OFF :	
YYYY-MM-DD	
****-**-**	

Example 7: Yearly mode on, Monthly mode off, Pulse off, On Time = 2008-12-15, Off Time = 2010-01-07: On December 15 of 2008 and 2009, the timer output switches on and remains on until January 7 of the following year. When the timer output turns off on January 7, 2010 it does NOT turn on again the following December 15.

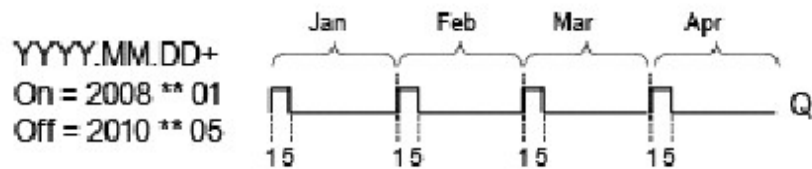


B6	1+
Yearly = On	
Monthly=Off	
Pulse =Off	

B6	2+
ON :	
YYYY-MM-DD	
2008-12-15	

B6	3+
OFF :	
YYYY-MM-DD	
2010-01-07	

Example 8: Yearly mode on, Monthly mode on, On Time = 2008-**-01, Off Time = 2010-**-05: Starting in 2008, on the first day of each month the timer output switches on and switches off on the fifth day of the month. The timer continues in this pattern through the last month of 2010.



B6	1+
Yearly =On	
Monthly=On	
Pulse =Off	

B6	2+
ON :	
YYYY-MM-DD	
2008-**-01	

B6	3+
OFF :	
YYYY-MM-DD	
2010-**-05	

Functional description

The yearly timer sets and resets the output at specific on and off dates. Sets and resets are executed at 00:00. If your application requires a different time, use a weekly timer together with a yearly timer in your circuit program.

The on time specifies when the timer is activated. The off time specifies when the output is reset again. For the on and off times, note the order of the fields: The first field defines the year, the second the month and the third the day.

If you set the Monthly mode on, the timer output switches on each month at the specified day of the on time and remains on until the specified day of the off time. The on time specifies the initial year in which the timer is activated. The off time defines the last year in which the timer turns off. The maximum year is 2099.

If you set the Yearly mode on, the timer output switches on each year at the specified month and day of the on time and remains on until the specified month and day of the off time. The on time specifies the initial year in which the timer is activated. The off time defines the last year in which the timer turns off. The maximum year is 2099.

If you set Pulse output, the timer output switches on at the specified on time for one cycle and then the timer output is reset. You can choose to pulse a timer on a monthly or yearly basis, or just a single time.

If you set none of the Monthly, Yearly, or Pulse modes on, you can define a specific time period with the on time and off time. It can span any time period that you choose.

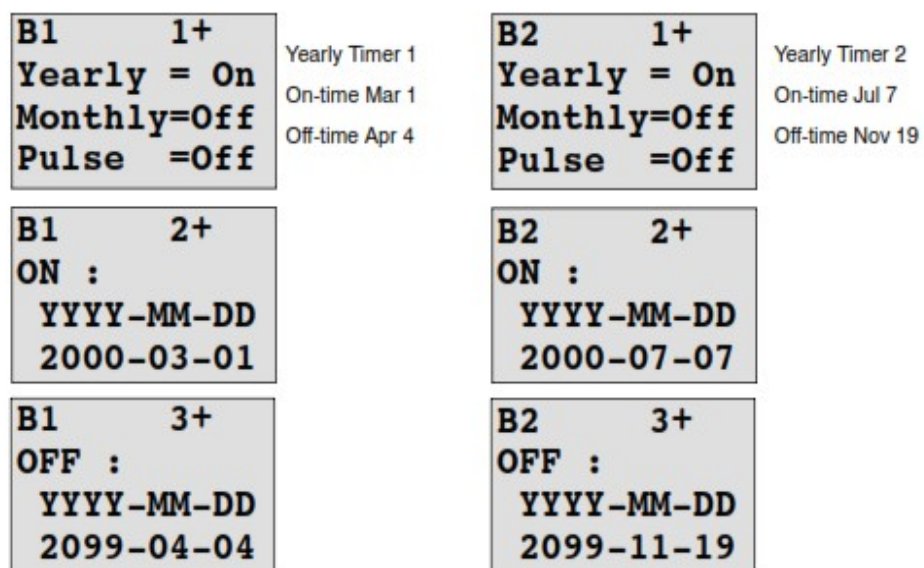
For a process action that is to be switched on and off at multiple but irregular times during the year, you can define multiple yearly timers with the outputs connected by an OR function block.

Backup of the real-time clock

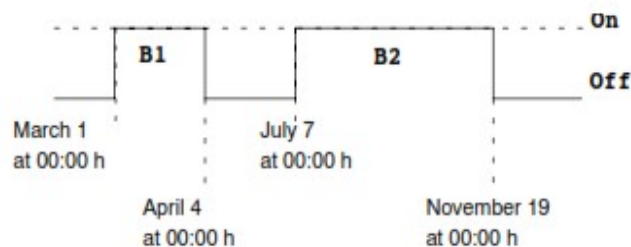
The internal real-time clock of LOGO! is buffered against power failure. The buffering time is influenced by the ambient temperature, and is typically 80 hours at an ambient temperature of 25 °C. If you are using the optional LOGO! Battery card, or combined LOGO! Memory/Battery card, LOGO! can retain the clock time for up to two years.

Sample configuration

The output of a LOGO! is to be set annually on March 1, reset on April 4, set again on July 7, and reset again on November 19. You need to configure two yearly timers with corresponding on-times. Then logically link the outputs by means of an OR block.



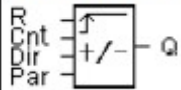
Result



4.4.13 Up/down counter

Short description

An input pulse increments or decrements an internal value, depending on the parameter setting. The output is set or reset when a configured threshold is reached. The direction of count can be changed with a signal at input Dir.

Symbol in LOGO!	Wiring	Description
	Input R	A signal at input R resets the internal count value to zero.
	Input Cnt	The function counts the 0 to 1 transitions at input Cnt. 1 to 0 transitions are not counted. Use <ul style="list-style-type: none"> inputs I3, I4, I5, and I6 for fast counting (only LOGO! 12/24 RC/RCo and LOGO! 24/24o): max. 5 kHz. any other input or circuit component for counting low frequency signals (typ. 4 Hz).
	Input Dir	You set the direction of count at input Dir: Dir = 0: Up count Dir = 1: Down count
	Parameter	On: On threshold Range of values: 0...999999 Off: Off threshold Range of values: 0...999999 StartVal: Initial value from which to begin counting either down or up. Retentivity for internal counter value Cnt: / = No retentivity R = The status is retentive.
	Output Q	Q is set or reset, depending on the current value at Cnt and the set thresholds.

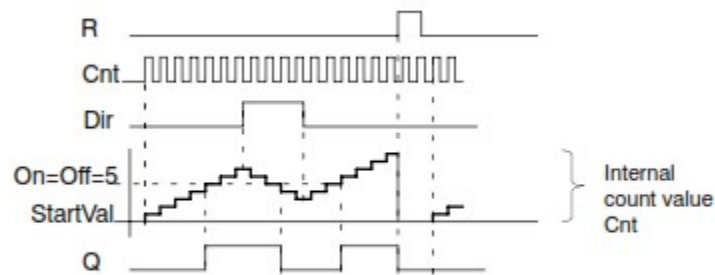
Parameters On and Off

The on threshold On and the off threshold Off can be provided by the actual value of another already-programmed function. You can use the actual values of the following functions:

- Analog comparator (actual value Ax - Ay, see Chapter 4.4.18)
- Analog threshold trigger (actual value Ax, see Chapter 4.4.16)
- Analog amplifier (actual value Ax, see Chapter 4.4.20)
- Analog multiplexer (actual value AQ, see Chapter 4.4.26)
- Analog ramp (actual value AQ, see Chapter 4.4.27)
- Analog math (actual value AQ, see Chapter 4.4.30)
- PI controller (actual value AQ, see Chapter 4.4.28)
- Up/down counter (actual value Cnt, see Chapter 4.4.13)

Select the required function by the block number.

Timing diagram



Functional description

The internal counter increments (Dir = 0) or decrements (Dir = 1) by one count with every positive edge at input Cnt.

You can use input R to reset the internal count value to the start value. As long as R = 1, the output is also 0 and the pulses at input Cnt are not counted.

If retentivity is not set, output Q and the expired time are reset after a power failure.

Q is set or reset depending on the current value at Cnt and the set thresholds. See the calculation rule below.

Calculation rule

- If the On threshold \geq Off threshold, then:
Q = 1, if Cnt \geq On
Q = 0, if Cnt < Off.
- If the On threshold < Off threshold, then Q = 1,
if On \leq Cnt < Off.

Note

The system scans the counter limit value cyclically.

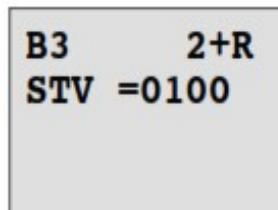
Thus, if the pulse frequency at the fast digital inputs I3, I4, I5 or I6 is faster than the cycle time, the special function might not switch until after the specified limit value is exceeded

Example: Up to 100 pulses per cycle can be counted; 900 pulses have been counted so far. On = 950; Off = 10000. The output is set in the next cycle, after the value has reached 1000. (The output would not be set at all if the value Off = 980.)

View in programming mode (example):

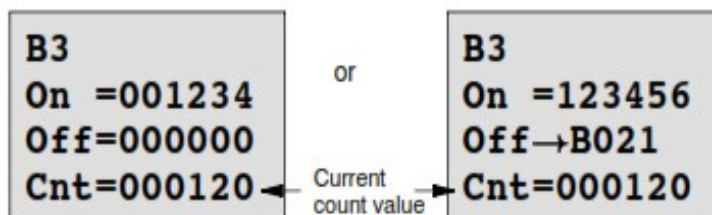


To set the start value, press ▲ or ▼ to access the following screen form:



If the referenced block (B021, in the example) returns a value that lies out of the valid range, the value is rounded to the next valid value.

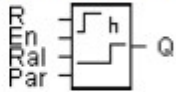
The view in parameter assignment mode (example):



4.4.14 Hours counter

Short description

A configured time is triggered with a signal at the monitoring input. The output is set when this time has expired.

Symbol in LOGO!	Wiring	Description
	Input R	A positive edge (0 to 1 transition) at input R resets output Q and sets a configured value MI at the counter for the duration of the time-to-go (MN).
	Input En	En is the monitoring input. LOGO! scans the on-time of this input.
	Input Ral	A positive edge at input Ral (Reset all) resets the hours counter (OT) and the output, and sets the time-to-go value (MN) to the maintenance interval MI: <ul style="list-style-type: none"> • output Q = 0, • measured operating time OT = 0 and • the time-to-go of the maintenance interval MN = MI.
	Parameter	MI: Maintenance interval to be preset in units of hours and minutes Range of values: 0000...9999 h, 0...59 m OT: The accumulated total operating time; you can specify an offset in hours and minutes Range of values: 00000...99999 h, 0...59 m Q→0: <ul style="list-style-type: none"> • When "R" is selected: Q = 1, if MN = 0; Q = 0, if R = 1 or Ral = 1 • When "R+En" is selected: Q = 1, if MN = 0; Q = 0, if R = 1 or Ral = 1 or En = 0.
	Output Q	The output is set when the time-to-go MN = 0 (see timing diagram). The output is reset: <ul style="list-style-type: none"> • When "Q→0:R+En", if R = 1 or Ral = 1 or En = 0 • When "Q→0:R", if R = 1 or Ral = 1.

MI = Configured time interval

MN= Time-to-go

OT= Total time expired since the last hi signal at input Ral

These values are always retentive!

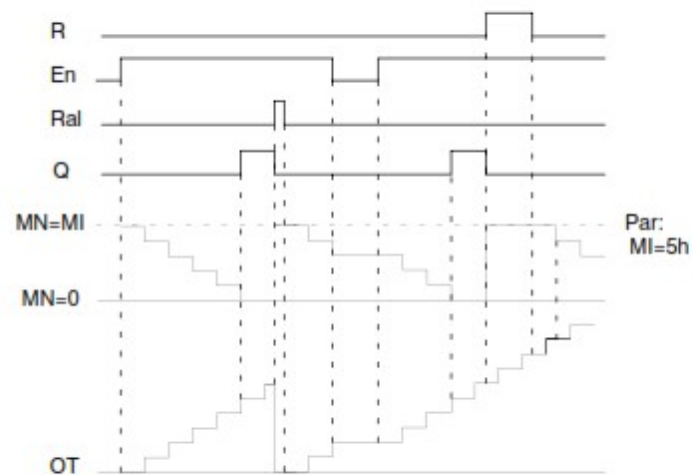
Parameter MI

The maintenance interval MI can be provided by the actual value of another already-programmed function. You can use the actual values of the following functions:

- Analog comparator (actual value Ax - Ay, see Chapter 4.4.18)
- Analog threshold trigger (actual value Ax, see Chapter 4.4.16)
- Analog amplifier (actual value Ax, see Chapter 4.4.20)
- Analog multiplexer (actual value AQ, see Chapter 4.4.26)
- Analog ramp (actual value AQ, see Chapter 4.4.27)
- Analog math (actual value AQ, see Chapter 4.4.30)
- PI controller (actual value AQ, see Chapter 4.4.28)
- Up/down counter (actual value Cnt, see Chapter 4.4.13)

Select the required function by the block number.

Timing diagram



MI = Configured time interval

MN = Time-to-go

OT = Total time expired since the last hi signal at input Ral

Functional description

The hours counter monitors input En. When En = 1, LOGO! computes the time expired and the time-to-go MN. LOGO! shows these times in parameter assignment mode. Output Q is set when the time-to-go MN = 0.

A signal at reset input R resets output Q and sets the preset value of MI at the counter for the duration of MN. The hours counter OT is not affected.

With a signal at the reset input Ral, you reset output Q and set the preset value of MI at the counter for the duration of MN. The hours counter OT is reset to zero.

Depending on your configuration of parameter Q, the output is either reset with a signal at input

R or Ral ("Q→0:R"), or when a reset signal is set hi, or the En signal is set lo ("Q→0:R+En").

Viewing the MI, MN and OT values

- LOGO! Basic with display unit: You can open the parameter assignment mode when the system is in RUN to view the actual values of MI, MN and OT.
- LOGO! Basic without display unit: In LOGO!Soft Comfort, you can use the Online Test to read these values (for further information, see Chapter 7).

Limit value of OT

The value of the operating hours in OT are retained when you reset the hours counter with a signal at input R. The hours counter OT continues the count as long as En = 1, irrespective of the status at the reset input R.

The counter limit of OT is 99999 h.

The hours counter stops when it reaches this value.

In programming mode, you can set the initial value of OT. The counter starts operation at any value other than zero. MN is automatically calculated at the START, based on the MI and OT values

(Example: MI is a reference parameter to the actual value of Block 1, which is 100. OT = 30, the result is MN = 70).

Setting the Par parameter

View in programming mode:

```
B16      1+R
MI = 0100h
      00 m
```

```
B16      1+R
MI-> B001h
```

```
B16      2+R
OT =00030h
      00 m
```

```
B16      3+R
Q->0:R+En
```

MI is the configurable time interval. The permissible range of values is 0 to 9999 hours.

For information on how to assign the actual value of another already-programmed function to a parameter, see section 4.4.1

View in parameter assignment mode:

```
B16      1
MI = 0100h ← Time interval
      00 m
```


```
B16      2
OT =00083h ← Total operating hours
      15 m
```

```
B16      3
MN = 0016h ← Time-to-go
      45 m
```


4.4.15 Threshold trigger

Short description

The output is set and reset with two configurable threshold triggers.

Symbol in LOGO!	Wiring	Description
	Input Fre	<p>The function counts the 0 to 1 transitions at input Fre. 1 to 0 transitions are not counted.</p> <p>Use</p> <ul style="list-style-type: none"> inputs I3, I4, I5, I6 for fast counting (only LOGO! 12/24 RC/RCo and LOGO! 24/24o): max. 5 kHz. any other input or circuit component for counting low frequency signals (typ. 4 Hz).
	Parameter	<p>On: On threshold Range of values: 0000...9999</p> <p>Off: Off threshold Range of values: 0000...9999</p> <p>G_T: Time interval or gate time during which the input pulses are measured. Range of values: 00:05 s...99:99 s</p>
	Output Q	Q is set and reset at the thresholds.

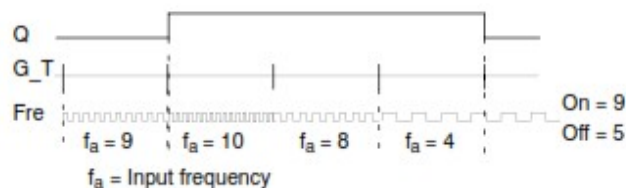
Parameter G_T

The gate time G_T can be provided by the actual value of another already-programmed function. You can use the actual values of the following functions:

- Analog comparator (actual value Ax - Ay, see Chapter 4.4.18)
- Analog threshold trigger (actual value Ax, see Chapter 4.4.16)
- Analog amplifier (actual value Ax, see Chapter 4.4.20)
- Analog multiplexer (actual value AQ, see Chapter 4.4.26)
- Analog ramp (actual value AQ, see Chapter 4.4.27)
- Analog math (actual value AQ, see Chapter 4.4.30)
- PI controller (actual value AQ, see Chapter 4.4.28)
- Up/down counter (actual value Cnt, see Chapter 4.4.13)

Select the required function by the block number.

Timing diagram



Functional description

The threshold trigger measures the signals at input Fre. The pulses are recorded across a configurable time G_T.

Output Q is set and reset in accordance with the set thresholds. See the calculation rule below.

Calculation rule

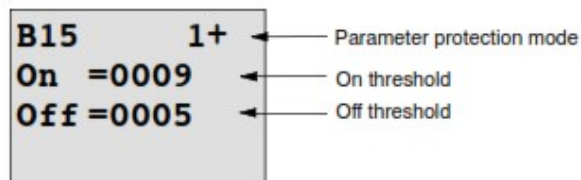
- If the On threshold \geq Off threshold, then:
Q = 1, if $f_a > \text{On}$
Q = 0, if $f_a \leq \text{Off}$.
- If the On threshold < Off threshold, then Q = 1 if
 $\text{On} \leq f_a < \text{Off}$.

Setting the Par parameter

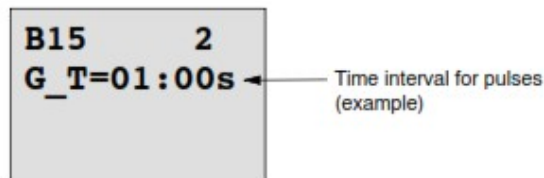
Note

The system scans the counter limit value once per interval G_T.

View in programming mode (example):



Press ►

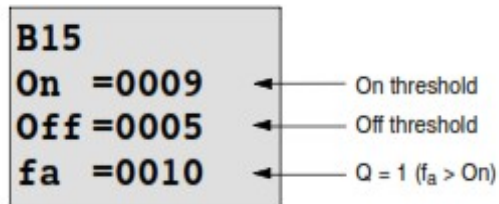


Note

The "seconds" timebase is here set as permanent default.

When you preset a time G_T of 1 s, LOGO! returns the current frequency in parameter f_a in Hz.

View in parameter assignment mode (example):




Note

f_a always represents the total pulses measured per time unit G_T .

4.4.16 Analog threshold trigger

Short description

The output is set and reset at two configurable thresholds.

Symbol in LOGO!	Wiring	Description
	Input Ax	<p>You apply the analog signal to be analyzed at input Ax.</p> <p>Use the analog inputs AI1...AI8 ^(*), the analog flags AM1...AM6, the block number of a function with analog output, or the analog outputs AQ1 and AQ2.</p>
	Parameter	<p>A: Gain Range of values: ±10.00</p> <p>B: Zero offset Range of values: ±10,000</p> <p>On: On threshold Range of values: ±20,000</p> <p>Off: Off threshold Range of values: ±20,000</p> <p>p: Number of decimals Range of values: 0, 1, 2, 3</p>
	Output Q	Q is set or reset by the threshold triggers.

* AI1...AI8: 0...10 V corresponds with 0...1000 (internal value).

Gain and offset parameters

Please note the information on gain and offset parameters in Chapter 4.3.6.

Parameters On and Off

The On and Off parameters can be provided by the actual value of another already-programmed function. You can use the actual values of the following functions:

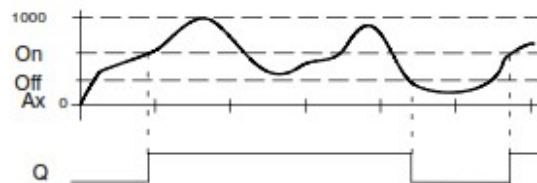
- Analog comparator (actual value Ax - Ay, see Chapter 4.4.18)
- Analog threshold trigger (actual value Ax, see Chapter 4.4.16)
- Analog amplifier (actual value Ax, see Chapter 4.4.20)
- Analog multiplexer (actual value AQ, see Chapter 4.4.26)
- Analog ramp (actual value AQ, see Chapter 4.4.27)
- Analog math (actual value AQ, see Chapter 4.4.30)
- PI controller (actual value AQ, see Chapter 4.4.28)
- Up/down counter (actual value Cnt, see Chapter 4.4.13)

Select the required function by the block number.

Parameter p (number of decimals)

Does not apply to the display of On, Off and Ax values in a message text.
Does not apply to the comparison of On and Off values! (The compare function ignores the decimal point.)

Timing diagram



Functional description

The function fetches the analog signal at input A_x .

A_x is multiplied by the value of the A (gain) parameter, and the value at parameter B (offset) is added to product, i.e. $(A_x \cdot \text{gain}) + \text{offset} = \text{actual value of } A_x$.

Output Q is set or reset, depending on the set thresholds. See the calculation rule below.

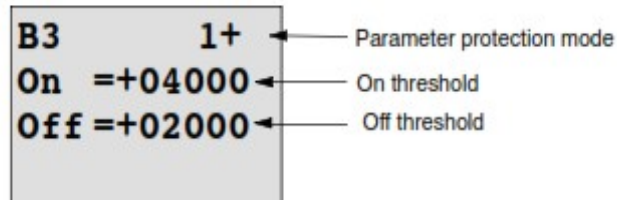
Calculation rule

- If the On threshold \geq Off threshold, then:
 $Q = 1$, if the actual value $A_x > \text{On}$
 $Q = 0$, if the actual value $A_x \leq \text{Off}$.
- If the On threshold $<$ Off threshold, then $Q = 1$ if
 $\text{On} \leq \text{actual value } A_x < \text{Off}$.

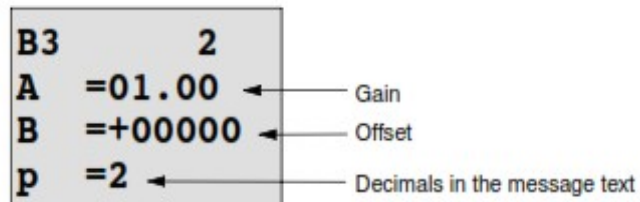
Setting the Par parameter

The gain and offset parameters are used to adapt the sensors to the relevant application.

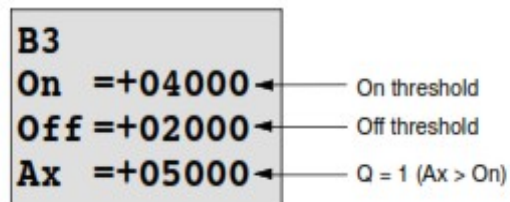
View in programming mode (example):



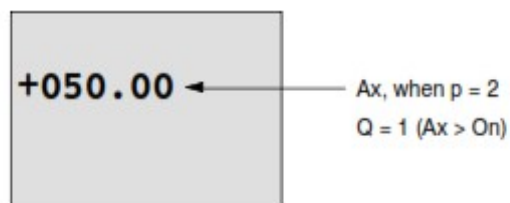
Press ►



View in parameter assignment mode (example):



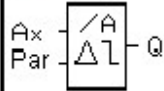
View in the message text (example):



4.4.17 Analog differential trigger

Short description

The output is set and reset depending on a configurable threshold and a differential value.

Symbol in LOGO!	Wiring	Description
	Input Ax	<p>You apply the analog signal to be analyzed at input Ax.</p> <p>Use the analog inputs AI1...AI8 (*), the analog flags AM1...AM6, the block number of a function with analog output, or the analog outputs AQ1 and AQ2.</p>
	Parameter	<p>A: Gain Range of values: $\pm 10,00$</p> <p>B: Zero offset Range of values: $\pm 10,000$</p> <p>On: On/Off threshold Range of values: $\pm 20,000$</p> <p>Δ: Differential value for calculating the off parameter Range of values: $\pm 20,000$</p> <p>p: Number of decimals Range of values: 0, 1, 2, 3</p>
	Output Q	Q is set or reset, depending on the threshold and difference values.

* AI1...AI8: 0...10 V corresponds with 0...1000 (internal value).

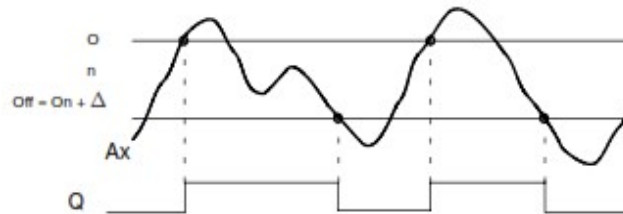
Gain and offset parameters

Please note the information on gain and offset parameters in Chapter 4.3.6.

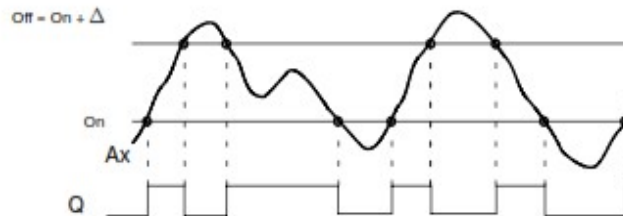
Parameter p (number of decimals)

Does not apply to the display of On, Off and Ax values in a message text.

Timing diagram A: Function with negative difference Δ



Timing diagram B: Function with positive difference Δ



Functional description

The function fetches the analog signal at input Ax.

Ax is multiplied by the value of the A (gain) parameter, and the value at parameter B (offset) is added to product, i.e. $(Ax \cdot \text{gain}) + \text{offset} = \text{actual value of Ax}$.

Output Q is set or reset, depending on the set (On) threshold and difference value (Δ). The function automatically calculates the Off parameter: $\text{Off} = \text{On} + \Delta$, whereby Δ may be positive or negative. See the calculation rule below.

Calculation rule

- When you set a negative differential value Δ , the On threshold \geq Off threshold, and:
Q = 1, if the actual value $Ax > \text{On}$
Q = 0, if the actual value $Ax \leq \text{Off}$.
See the timing diagram A.
- When you set a positive differential value Δ , the On threshold $<$ the Off threshold, and Q = 1, if:
 $\text{On} \leq \text{actual value } Ax < \text{Off}$.
See the timing diagram B.

Setting the Par parameter

The gain and offset parameters are used to adapt the sensors to the relevant application.

View in programming mode (example):

B3	1+	← Parameter protection mode
On	=+04000	← On/off threshold
Δ	=-02000	← Differential value for the on/off threshold

Press ►

B3	2	
A	=01.00	← Gain
B	=+00000	← Offset
p	=2	← Decimals in the message text

View in parameter assignment mode (example):

B3		
On	=+04000	← On threshold
Δ	=-02000	← Differential value for the off threshold
Ax	=+05000	← Q = 1 (Ax > On)


Press ▼

B3		
Off	=+02000	← Off threshold

4.4.18 Analog comparator

Short description

The output is set and reset depending on the difference $A_x - A_y$ and on two configurable thresholds.

Symbol in LOGO!	Wiring	Description
	Inputs A_x and A_y	<p>You apply the analog signals the difference of which you want to analyze at the inputs A_x and A_y.</p> <p>Use the analog inputs AI1...AI8 (*), the analog flags AM1...AM6, the block number of a function with analog output, or the analog outputs AQ1 and AQ2.</p>
	Parameter	<p>A: Gain Range of values: ± 10.00</p> <p>B: Zero offset Range of values: $\pm 10,000$</p> <p>On: On threshold Range of values: $\pm 20,000$</p> <p>Off: Off threshold Range of values: $\pm 20,000$</p> <p>p: Number of decimals Range of values: 0, 1, 2, 3</p>
	Output Q	Q is set or reset, depending on the difference $A_x - A_y$ and the set thresholds..

* AI1...AI8: 0...10 V corresponds with 0...1000 (internal value).

Gain and offset parameters

For more information on the gain and offset parameters, refer to Chapter 4.3.6.

Parameters On and Off

The on threshold On and the off threshold Off can be provided by the actual value of another already-programmed function. You can use the actual values of the following functions:

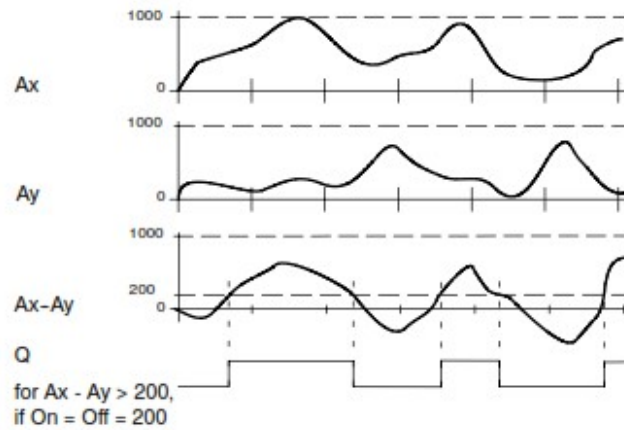
- Analog comparator (actual value $A_x - A_y$, see Chapter 4.4.18)
- Analog threshold trigger (actual value A_x , see Chapter 4.4.16)
- Analog amplifier (actual value A_x , see Chapter 4.4.20)
- Analog multiplexer (actual value AQ, see Chapter 4.4.26)
- Analog ramp (actual value AQ, see Chapter 4.4.27)
- Analog math (actual value AQ, see Chapter 4.4.30)
- PI controller (actual value AQ, see Chapter 4.4.28)
- Up/down counter (actual value Cnt, see Chapter 4.4.13)

Select the required function by the block number.

Parameter p (number of decimals)

Does not apply to Ax, Ay, On, Off and Δ values displayed in a message text.
Does not apply to the comparison of on and off values! (The compare function ignores the decimal point.)

Timing diagram



Functional description

The function fetches the analog values from the inputs Ax and Ay.

Ax and Ay are each multiplied by the value of the A (gain) parameter, and the value at parameter B (offset) is then added to the relevant product, i.e.

$(Ax \cdot \text{gain}) + \text{offset} = \text{actual value Ax}$ or

$(Ay \cdot \text{gain}) + \text{offset} = \text{actual value Ay}$.

The function forms the difference (" Δ ") between the actual values Ax - Ay.

Output Q is set or reset, depending on difference of the actual values Ax - Ay and the set thresholds. See the calculation rule below.

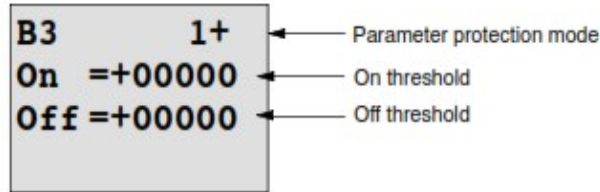
Calculation rule

- If the On threshold \geq Off threshold, then:
Q = 1, if:
 $(\text{actual value Ax} - \text{actual value Ay}) > \text{On}$
Q = 0, if:
 $(\text{actual value Ax} - \text{actual value Ay}) \leq \text{Off}$.
- If the On threshold < Off threshold, then Q = 1, if:
 $\text{On} \leq (\text{actual value Ax} - \text{actual value Ay}) < \text{Off}$.

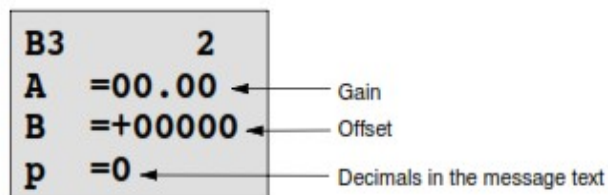
Setting the Par parameter

The gain and offset parameters are used to adapt the sensors to the relevant application.

View in programming mode:



Press ►



Example

In a heating control system, the supply T_v and return line temperatures T_r are to be compared, for example with a sensor at AI2.

A control signal is to be triggered (for example "heater On") when the difference between the supply and return line temperatures is greater than 15° . The control signal is reset when the difference is less than 5°C .

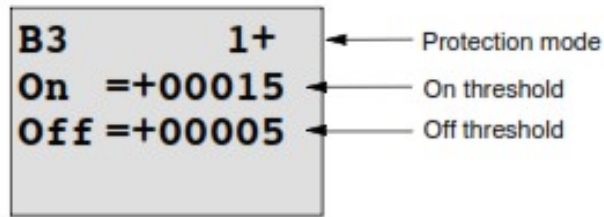
The process variable of the temperature is to be shown in parameter assignment mode.

The thermocouples available have the the following technical data: -30 to $+70^\circ\text{C}$, 0 to 10 VDC.

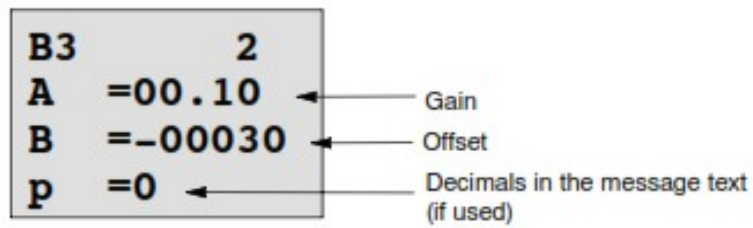
Application	Internal mapping
-30 to $+70^\circ\text{C} = 0$ to 10 V DC	0 to 1000
0°C	300 → Offset = -30
Range of values: -30 to $+70^\circ\text{C} = 100$	1000 → Gain = $100/1000 = 0.1$
On threshold = 15°C	Threshold = 15
Off threshold = 5°C	Threshold = 5

See also Chapter 4.3.6.

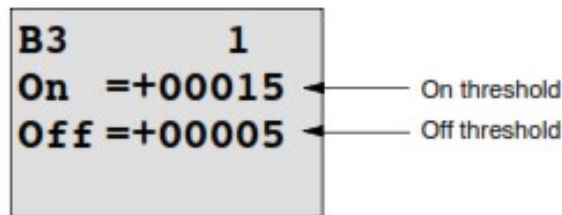
Configuration (example):



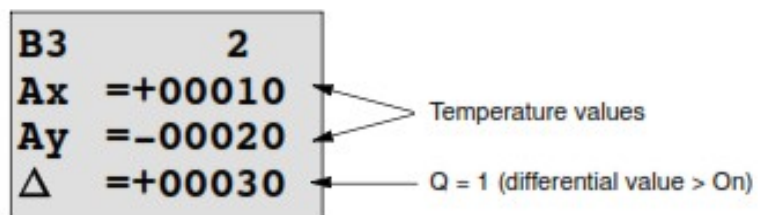
Press ►



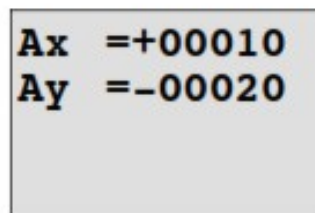
View in parameter assignment mode (example):



Press ▼



View in the message text (example):

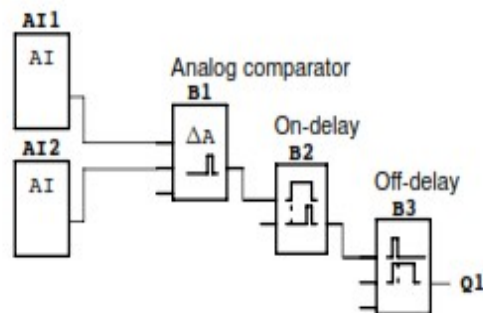


Reducing the input response of the analog comparator

You can selectively delay the output of an analog comparator by means of the "On-delay" and "Off-delay" special functions. With on-delay, output Q is only set if the pulse width of the triggering signal at input Trg (=analog comparator output) is longer than the on-delay time.

Using this method, you will obtain a virtual hysteresis and reduce the input response to short signals.


Function block diagram



4.4.19 Analog watchdog

Short description

This special function saves the process variable of an analog input to memory, and sets the output when the output variable exceeds or drops below this stored value plus a configurable offset.

Symbol in LOGO!	Wiring	Description
	Input En	A positive edge (0 to 1 transition) at input En saves the analog value at input Ax ("Aen") to memory and starts monitoring of the analog range $Aen - \Delta_2$ to $Aen + \Delta_1$
	Input Ax	You apply the analog signal to be monitored at input Ax. Use the analog inputs AI1...AI8 (*), the analog flags AM1...AM6, the block number of a function with analog output, or the analog outputs AQ1 and AQ2.
	Parameter	A: Gain Range of values: ± 10.00 B: Zero offset Range of values: $\pm 10,000$ Δ_1 : Difference value above Aen: on/off threshold Range of values: 0-20,000 Δ_2 : Difference value below Aen: on/off threshold Range of values: 0-20,000 p: Number of decimals Range of values: 0, 1, 2, 3
	Output Q	Q is set/reset, depending on the stored analog value and the offset.

* AI1...AI8: 0...10 V corresponds with 0...1000 (internal value).

Gain and offset parameters

For more information on gain and offset parameters, refer to Chapter 4.3.6.

Parameters Delta1 and Delta2

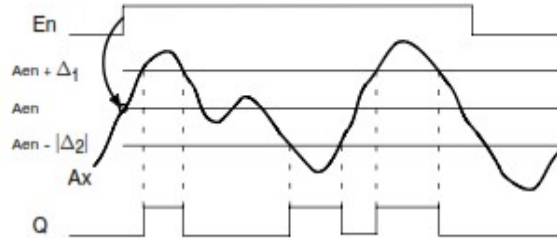
The Delta1 and Delta2 parameters can be provided by the actual value of another already-programmed function. You can use the actual value of the following functions:

- Analog comparator (actual value $Ax - Ay$, see Chapter 4.4.18)
- Analog threshold trigger (actual value Ax , see Chapter 4.4.16)
- Analog amplifier (actual value Ax , see Chapter 4.4.20)
- Analog multiplexer (actual value AQ , see Chapter 4.4.26)
- Analog ramp (actual value AQ , see Chapter 4.4.27)
- Analog math (actual value AQ , see Chapter 4.4.30)
- PI controller (actual value AQ , see Chapter 4.4.28)
- Up/down counter (actual value Cnt , see Chapter 4.4.13)

Select the required function by the block number.

Parameter p (number of decimals)

Applies only to the Aen, Ax, Δ_1 and Δ_2 values displayed in a message text.

Timing diagram**Functional description**

A 0 to 1 transition at input En saves the value of the signal at the analog input Ax. This saved process variable is referred to as "Aen".

Both the analog actual values Ax and Aen are multiplied by the value at parameter A (gain), and parameter B (offset) is then added to the product:

$(Ax \cdot \text{gain}) + \text{offset} = \text{Actual value } Aen$, when input En changes from 0 to 1, or

$(Ax \cdot \text{gain}) + \text{offset} = \text{Actual value } Ax$.

Output Q is set when the signal at input En = 1 and if the actual value at input Ax is out of range of $Aen - \Delta_2$ to $Aen + \Delta_1$.

Output Q is reset, when the actual value at input Ax lies within the range of $Aen - \Delta_2$ to $Aen + \Delta_1$, or when the signal at input En changes to 0.

Setting the Par parameter

The gain and offset parameters are used to adapt the used sensors to the respective application.

View in programming mode:

B3	1+	← Parameter protection mode
Δ1 =	00000	← Differential value for the on/off threshold
Δ2 =	00000	

Press ►

B3	2	
A =	00.00	← Gain
B =	+00000	← Offset
p =	0	← Decimals in the message text

View in parameter assignment mode (example):

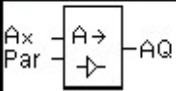
B3		
Δ1 =	00010	
Aen =	-00020	
Ax =	+00005	← Q = 1 (Ax is out of the range of Aen - Δ2 to Aen + Δ1)

B3	
Δ2 =	00010

4.4.20 Analog amplifier

Short description

This special function amplifies the value of an analog input and outputs the result at an analog output.

Symbol in LOGO!	Wiring	Description
	Input Ax	<p>You apply the analog signal to be amplified at input Ax.</p> <p>Use the analog inputs AI1...AI8 (*), the analog flags AM1...AM6, the block number of a function with analog output, or the analog outputs AQ1 and AQ2.</p>
	Parameter	<p>A: Gain Range of values: ± 10.00</p> <p>B: Zero offset Range of values: $\pm 10,000$</p> <p>p: Number of decimals Range of values: 0, 1, 2, 3</p>
	Output AQ	<p>This special function has an analog output! This output can only be connected with the analog input of a function, an analog flag or an analog output connector (AQ1, AQ2).</p> <p>Range of values for AQ: -32768...+32767</p>

* AI1...AI8: 0...10 V corresponds with 0...1000 (internal value).

Gain and offset parameters

Please note the information on gain and offset parameters in Chapter 4.3.6.

Parameter p (number of decimals)

Applies only to the AQ value in a message text.

Functional description

The function fetches the analog signal of input Ax.

This value is multiplied by the value of the A (gain) parameter, and parameter B (offset) is then added to the product: $(Ax \cdot \text{gain}) + \text{offset} = \text{actual value Ax}$.

The actual value Ax is output at AQ.

Analog output

If you interconnect this special function with a real analog output, note that the analog output can only process values between 0 and 1000. To do this, you may need to connect an additional amplifier between the analog output of the special function and the real analog output. Using this amplifier, you standardize the output range of the special function to a value range of 0 to 1000.

Scaling an analog input value

You can influence the analog input value of a potentiometer by interconnecting an analog input with an analog amplifier and an analog flag.

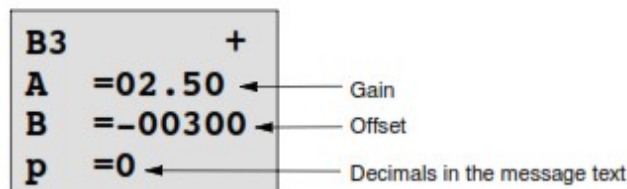
- Scale the analog value at the analog amplifier for further use.
- Connect, for example, the time base for parameter T of a time function (e.g. On-/Off-delay, Chapter 4.4.3) or the on and/or off limit specification of an up/down counter (Chapter 4.4.13) to the scaled analog value.

For more information with programming examples refer to the online help for LOGO!Soft Comfort.

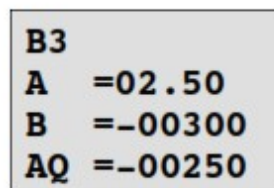
Setting the Par parameter

The gain and offset parameters are used to adapt the sensors to the relevant application.

View in programming mode (example):



View in parameter assignment mode (example):



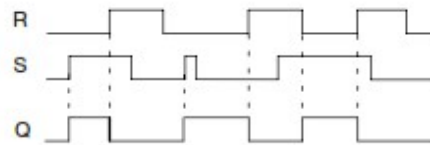
4.4.21 Latching relay

Short description

Input S sets output Q, input R resets output Q again.

Symbol in LOGO!	Wiring	Description
	Input S	You set output Q with a signal at input S.
	Input R	You reset output Q with a signal at input R. If S and R = 1, the output is reset.
	Parameter	Retentivity: / = No retentivity R = The status is retentive.
	Output Q	Q is set with a signal at input S, and reset with a signal at input R.

Timing diagram



Switching response

A latching relay represents a simple binary element. The output value depends on the status at the inputs and on the previous output status. The following table shows the logic once again:

S_n	R_n	Q	Comment
0	0	x	The status is retentive
0	1	0	Reset
1	0	1	Set
1	1	0	Reset (takes priority over Set)

When retentivity is enabled, the current status of the output signal is retained after a power failure.

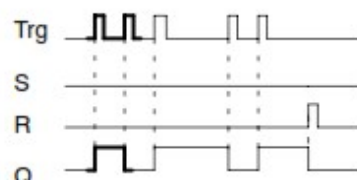
4.4.22 Pulse relay

Short description

A short pulse at the input sets and resets the output.

Symbol in LOGO!	Wiring	Description
	Input Trg	You set and reset output Q with a signal at input Trg (Trigger).
	Input S	You set output Q with a signal at input S.
	Input R	You reset output Q with a signal at input R.
	Parameter	Selection: RS (R input priority) or SR (S input priority) Retentivity: / = No retentivity R = The status is retentive.
	Output Q	Q is set with a signal at Trg, and reset with the next signal at Trg, if S and R = 0.

Timing diagram



The bold printed section of the timing diagram is also shown in the symbol for the pulse relay.

Functional description

Output Q changes its status; that is, the output is set or reset with each 0 to 1 transition at input Trg and if the inputs S and R = 0.

The signal at input Trg does not influence the special function when S or R = 1.

You set the pulse relay with a signal at input S. The output is set hi.

You reset the pulse relay with a signal at input R. The output is set lo.

Status diagram

Par	Q _{n-1}	S	R	Trg	Q _n
*	0	0	0	0	0
*	0	0	0	0 ->1	1**
*	0	0	1	0	0
*	0	0	1	0 ->1	0
*	0	1	0	0	1
*	0	1	0	0 ->1	1
RS	0	1	1	0	0
RS	0	1	1	0 ->1	0
SR	0	1	1	0	1
SR	0	1	1	0 ->1	1
*	1	0	0	0	1
*	1	0	0	0 ->1	0**
*	1	0	1	0	0
*	1	0	1	0 ->1	0
*	1	1	0	0	1
*	1	1	0	0 ->1	1
RS	1	1	1	0	0
RS	1	1	1	0 ->1	0
SR	1	1	1	0	1
SR	1	1	1	0 ->1	1

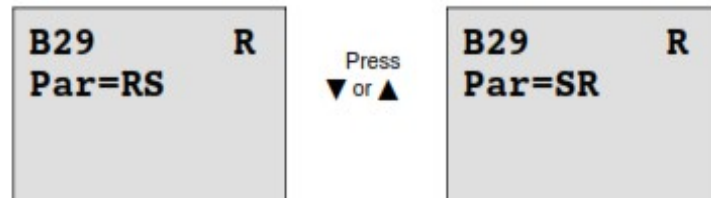
*: RS or SR

** : Triggering signal is effective, because S and R = 0.

Depending on your configuration, either input R takes priority over input S (input S is not effective when R = 1), or vice versa (input R is not effective when S = 1).

After a power failure, the pulse relay and output Q are reset if you have not enabled retentivity.

View in programming mode:



This special function is not available in parameter assignment mode.

Note

If Trg = 0 and Par = RS, the special function "Pulse relay" corresponds with the special function "Latching relay" (see Chapter 4.4.21).

4.4.23 Message texts

Short description

With the message text function block, you can configure a message that includes text and other parameters for LOGO! to display in RUN mode.

You can configure simple message texts from the LOGO! Display. LOGO!Soft Comfort provides an extended set of features for message texts: bar graph representation of data, names for digital I/O states and more. Refer to the LOGO!Soft Comfort documentation for information on these features.

Global Message Text Settings

You configure global parameters that apply to all message texts from the Msg Config selections on the Programming menu:

- Analog Time: refresh rate in milliseconds that specifies how frequently analog inputs in message texts are updated
- Tick time: frequency at which message texts scroll on and off the display
There are two ways that a message text can tick on and off the screen: line by line, or character by character, which are described in more detail below. A line of a text message, or each character of a text message in turn will tick on and off the LOGO! Display based on the tick time. For a message that ticks line by line, the actual tick time is ten times the configured tick time. For messages that tick character by character, the actual tick time is the configured tick time.

- CharSets: The primary and secondary character sets from which you can configure message texts. CharSet1 and CharSet2 can be any of the supported character sets for LOGO!:

Character Set in LOGO!	Common Name	Supports Languages	Internet Reference
ISO8859-1	Latin-1	English, German, Italian, Spanish (partly), Dutch (partly)	http://en.wikipedia.org/wiki/ISO/IEC_8859-1
ISO8859-5	Cyrillic	Russian	http://en.wikipedia.org/wiki/ISO/IEC_8859-5
ISO8859-9	Latin-5	Turkish	http://en.wikipedia.org/wiki/ISO/IEC_8859-9
ISO8859-16	Latin-10	French	http://en.wikipedia.org/wiki/ISO/IEC_8859-16
GB-2312	Chinese	Chinese	http://en.wikipedia.org/wiki/GB2312

- Current Character Set: which character set is selected for the display of message texts

Of the fifty possible message texts that you can configure, you can select any number of them to be from the first language and any number from the second language. For example, you could configure fifty message text function blocks that have a single message text for Character Set 1. Alternatively, you could configure twenty-five message text function blocks, each of which has two message texts: one for Character Set 1 and one for Character Set 2. Any combination is valid such that the total does not exceed fifty.

Within a single message text, the text must be from one character set. You can edit message texts in any of the supported character sets from LOGO!Soft Comfort. From the LOGO! Basic Module, you can only edit text using characters from the ISO8859-1 character set.

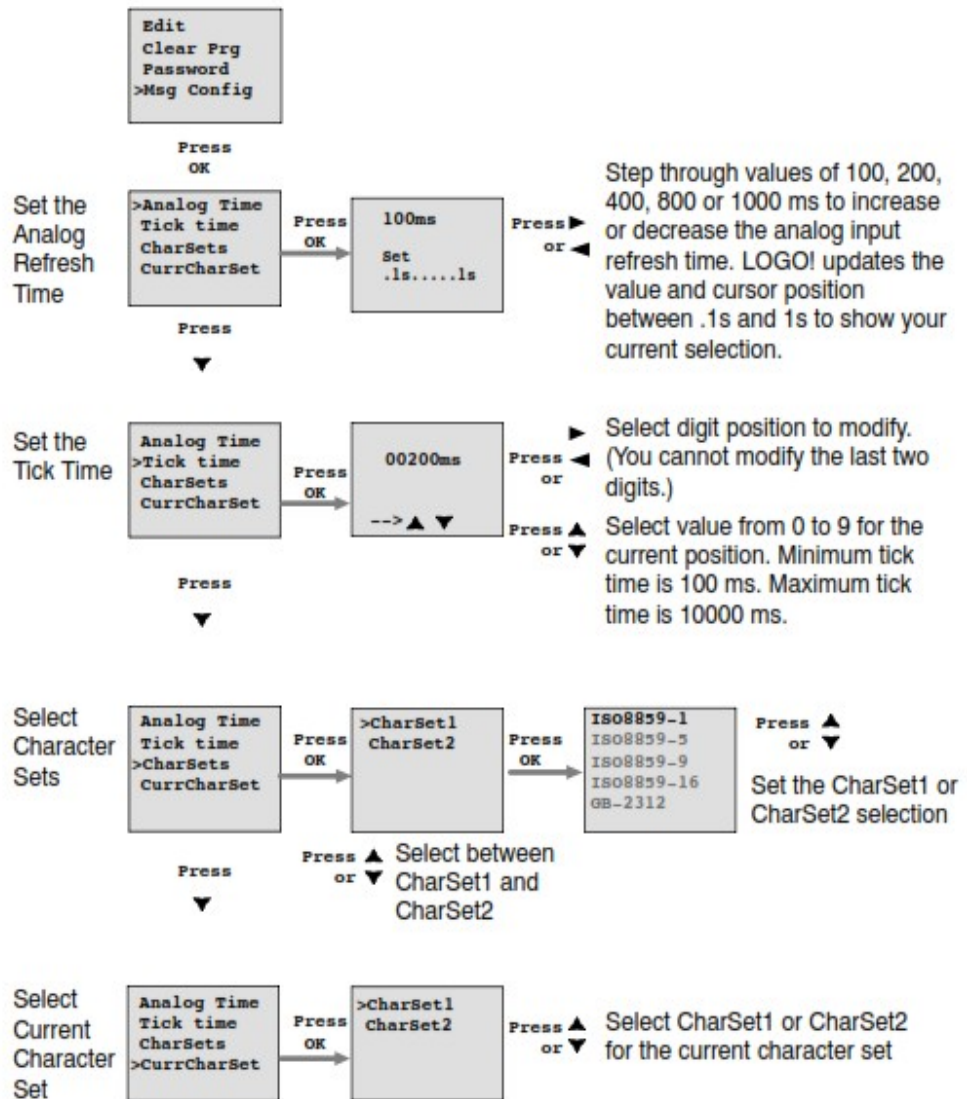
The language and therefore character set of a message text is independent of the language setting for the LOGO! display menus. They can be different.

Chinese Character Set

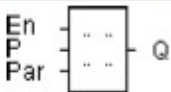
The LOGO! Basic module and LOGO! TD support the Chinese character set (GB-2312) for the People's Republic of China. The devices use Microsoft Windows encoding for this character set. The Windows encoding allows the devices to display the same characters as shown in the LOGO!Soft Comfort message text editor when you are using a Chinese emulator or a Chinese version of Microsoft Windows.

The Chinese character set requires a Chinese version of Windows or a Chinese emulator to properly display Chinese characters in the LOGO!Soft Comfort message text editor. You must start the Chinese emulator before you open the the message text function block in LOGO!Soft Comfort.

Programming Global Message Text Parameters



Message Text Function Block

Symbol in LOGO!	Wiring	Description
	Input En	A 0 to 1 transition at input En (Enable) starts the output of the message text.
	Input P	P: Priority of the message text Range of values: 0...127 Message destination Message tick settings Ack: Acknowledgment of the message text
	Parameter	Text: Input of the message text Par: Parameter or actual value of another already-programmed function (see "Visible parameters or process variables") Time: Display of the continuously updated time-of-day Date: Display of the continuously updated date EnTime: Display of the time of the 0 to 1 transition of the signal at input En EnDate: Display of the date of the 0 to 1 transition of the signal at input En I/O status names: Display of a digital input or output status name, for example "On" or "Off" Analog Input: Display of an analog input value to be shown in message text and updated according to the analog time. Note: You can only edit the Text parameter of the message from the LOGO! Basic module. ISO8859-1 is the only available character set for editing text. You can edit all other parameters, and other languages for the Text parameter from LOGO!Soft Comfort. See the online help for configuration details.
	Output Q	Q remains set as long as the message text is set.

Restriction

A maximum of 50 message texts are available.

Functional description

When LOGO! is in RUN mode, LOGO! displays the message text that you have configured along with its parameter values upon a 0 to 1 transition of the signal at input En.

Based on your setting for the message destination, the message text displays on the LOGO! Display, the LOGO! TD, or both.

If you use flag M27 in your circuit program, then if M27=0 (low) then LOGO! displays the message text only if it is from the primary character set (Character Set 1). If M27=1 (high), then LOGO! displays the message text only if it is from the secondary character set (Character Set 2). (See the M27 flag description in section 4.1).

If you have configured message ticking, the message will tick on and off the display according to your specifications, either a character at a time, or a line at a time.

If acknowledgment is disabled (Ack = Off), the message text is hidden when the status of the signal at input En changes from 1 to 0.

If acknowledgment is enabled (Ack = On) and the status of the signal at input En changes from 1 to 0, the message text is output until it is acknowledged with **OK**. When En = 1, you cannot acknowledge the message text.

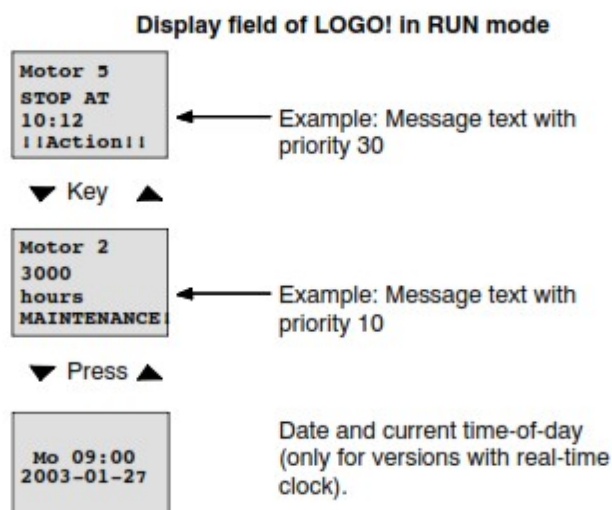
When multiple message text functions are triggered with En=1, LOGO! displays the message text with the highest priority (0 = lowest, 127 = highest). This also means that LOGO! displays a newly activated message text only if its priority is higher than that of previously activated message texts.

After a message text is disabled or acknowledged, the function automatically shows the previously active message text that takes the highest priority.

You can change the view and the message texts by pressing the keys ▲ and ▼.

Example

This is how two message texts could be shown:



Message Ticking

You can configure message text lines to tick or not tick. Two types of message ticking exist:

- Character by character
- Line by line

Messages that tick character by character scroll off the characters of the message line one character at a time to the left with the additional characters scrolling in one at a time from the right. The time interval for the tick is specified by the TickTime message text setting.

Messages that tick line by line by line scroll one half of the message off the display to the left with the second half of the message scrolling in from the right. The time interval for the tick is ten times the TickTime parameter. The two halves of the message simply alternate on the LOGO! Display or LOGO! TD.

Example: Tick Message Character by Character

The following illustration shows a one-line, 24-character message text:

X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁	X ₁₂	X ₁₃	X ₁₄	X ₁₅	X ₁₆	X ₁₇	X ₁₈	X ₁₉	X ₂₀	X ₂₁	X ₂₂	X ₂₃	X ₂₄	

If you set this message to tick "character by character" with a tick interval of 0.1 seconds, then the initial appearance of this message line on the LOGO! Display or LOGO! TD is as shown in this illustration:

X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁	X ₁₂	X ₁₃	X ₁₄	X ₁₅	X ₁₆	X ₁₇	X ₁₈	X ₁₉	X ₂₀	X ₂₁	X ₂₂	X ₂₃	X ₂₄
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

After 0.1 second, one character of the message line ticks. The message appears as follows on the LOGO! Display or LOGO! TD:

X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁	X ₁₂	X ₁₃	X ₁₄	X ₁₅	X ₁₆	X ₁₇	X ₁₈	X ₁₉	X ₂₀	X ₂₁	X ₂₂	X ₂₃	X ₂₄	X ₁
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	----------------

Example: Tick Message Line by Line

The following example uses the same message configuration as the previous example:

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16	X17	X18	X19	X20	X21	X22	X23	X24	

If you set this message to tick "line by line" with a tick interval of 0.1 seconds, then the initial appearance of this message on the LOGO! Display or LOGO! TD is the left half of the message as shown in this illustration:

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16	X17	X18	X19	X20	X21	X22	X23	X24
----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

After 1 second (10 x 0.1 second), the message ticks to show the right half of the message as shown in this illustration:

X13	X14	X15	X16	X17	X18	X19	X20	X21	X22	X23	X24	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	----	----	----	----	----	----	----	----	----	-----	-----	-----

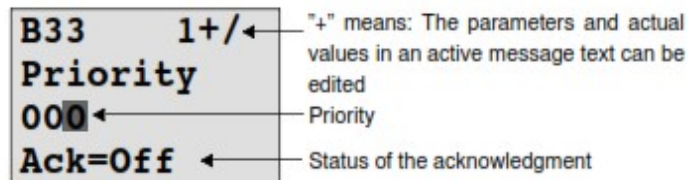
The screen display alternates between the two message halves every second. You can configure each individual line of a message text to tick or not tick. The "character by character" or "line by line" setting applies to all lines that you configure to tick.

Input P configuration

From the input P, you configure the following characteristics of the message text:

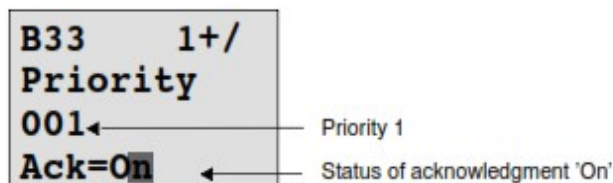
- Priority
- Acknowledgement
- Message destination
- Tick type, and tick setting for each line

To configure the priority and the acknowledgment (programming mode):



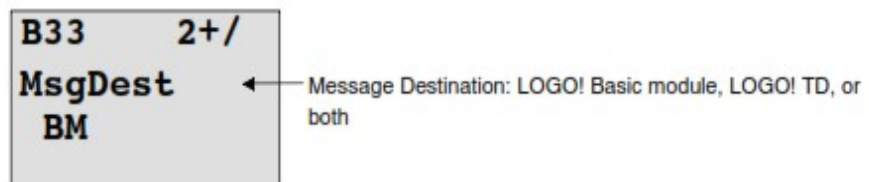
1. Increase the priority to 1:
Cursor on '0' + ▲
2. Change to 'Ack':
Press ►
3. Enable 'Ack':
Press ▲ or ▼

LOGO! shows:

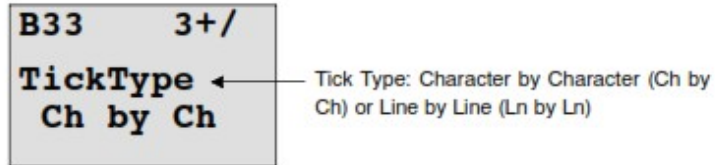


To configure the message destination and tick type (programming mode):

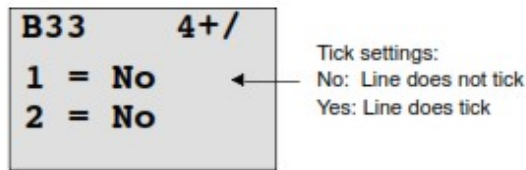
1. From the Priority and Acknowledgment screen form, press ► to access the Message Destination screen form.



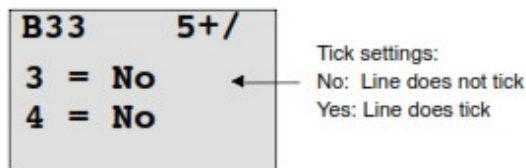
2. Press ► to position the cursor on the "BM" line.
3. Press ▲ or ▼ to toggle through the three choices for message destination: BM, TD, or BM & TD.
4. From the Message Destination screen form, press ► to access the Tick Type screen form.



5. If your message text has lines that tick, press ► to position the cursor on the "Ch by Ch" line and then press ▲ or ▼ to select either "Ch by Ch" or "Ln by Ln" for the TickType.
6. From the Tick T ype screen, press ► to enable or disable ticking for each line of the message text. LOGO! displays the following screen form:



7. Press ▲ or ▼ to choose between "No" and "Yes" to define whether line 1 ticks.
8. Press ► to move the cursor to the second line, and press ▲ or ▼ to choose between "No" and "Yes" for line 2. From the last line press ► to go to the screen from for lines 3 and 4. Configure line ticking for lines 3 and 4 in the same way as for lines 1 and 2.



9. Press **OK** to confirm the complete message text configuration.

Visible parameters or process variables

The following parameters or process variables can be displayed in a message text, as either numerical values or bar-graph representations of values:

Special function	Parameter or process variable visible in a message text
Timers	
On-delay	T, T _a
Off-delay	T, T _a
On-/Off-delay	T _a , T _H , T _L
Retentive on-delay	T, T _a
Wiping relay (pulse output)	T, T _a
Edge triggered wiping relay	T _a , T _H , T _L
Asynchronous pulse generator	T _a , T _H , T _L

Special function	Parameter or process variable visible in a message text
Random generator	T _H , T _L
Stairway lighting switch	T _a , T, T _! , T _{!L}
Multiple function switch	T _a , T, T _L , T _! , T _{!L}
Weekly timer	3*on/off/day
Yearly timer	On, Off
Counter	
Up/down counter	Cnt, On, Off
Hours counter	MI, Q, OT
Threshold trigger	f _a , On, Off, G_T
Analog	
Analog threshold trigger	On, Off, A, B, Ax
Analog differential trigger	On, Δ, A, B, Ax, Off
Analog comparator	On, Off, A, B, Ax, Ay, ΔA
Analog value monitoring	Δ, A, B, Ax, Aen
Analog amplifier	A, B, Ax
Analog multiplexer	V1, V2, V3, V4, AQ
Analog ramp	L1, L2, MaxL, StSp, Rate, A, B, AQ
PI controller	SP, Mq, KC, TI, Min, Max, A, B, PV, AQ
Analog math	V1, V2, V3, V4, AQ
PWM (Pulse Width Modulator)	A, B, T, Ax amplified
Miscellaneous	
Latching relay	-
Pulse relay	-
Message texts	-
Softkey	On/Off
Shift register	-

For timers, a message text can also display the remaining time. "Remaining time" refers to how much time of the parameter setting remains.

Bar graphs can be either horizontal or vertical representations of the current or actual value scaled between the minimum and maximum value. For more information on configuring and displaying bar graphs in message texts, refer to the LOGO!Soft Comfort online help.

Editing message texts

Only simple message texts can be edited from the LOGO! Basic module. Text messages created in LOGO!Soft Comfort that make use of new features such as bar graphs, I/O status names, and others can not be edited from the LOGO! Basic module.

Also, you cannot edit message texts from the LOGO! Basic module that contain any of the following parameters:

- Par
- Time
- Date
- EnTime
- EnDate

You can only edit such message texts from LOGO!Soft Comfort.

Changing parameters in the active message text

When the message text is active, press **ESC** to select the editing mode.

Note

You must keep the **ESC** key pressed for at least one second

Press ◀ and ▶ to select the relevant parameter. Press **OK** to change the parameter. Use the ▲ and ▼ keys to edit a parameter.

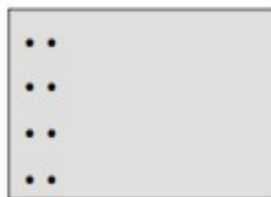
Confirm your changes with **OK**. You can now edit further parameters in the message text (if any exist). Press **ESC** to exit editing mode.

Key input simulation in the active message text

You can enable the four cursor keys C ▲, C ▶, C ▼ and C ◀ in an active message text by pressing **ESC** plus the relevant cursor key.

Setting the Par parameter

View in programming mode:



Parameter assignment
screen form for Par

Press **▶** to select a line for the message text.

Press **▲** and **▼** to select the letter to be displayed in the text. To move the cursor from one position to the other, press **◀** and **▶**.

The list of available characters is the same as for the circuit program name. The character set is found in Chapter 3.7.4. When you enter message text from the LOGO! Basic module, you can only enter characters from the ISO8859-1 character set. To enter text from another language, you must enter the text in LOGO!Soft Comfort.

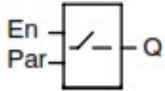
Note that the number of characters per line of message text can be greater than the number of character positions on the LOGO! Display.

Press **OK** to confirm your changes, and press **ESC** to exit the editing mode.

4.4.24 Softkey

Short description

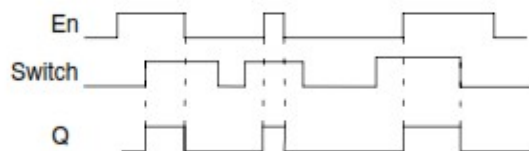
This special function has the effect of a mechanical pushbutton or switch.

Symbol in LOGO!	Wiring	Description
	Input En	Output Q is set with a 0 to 1 transition of the signal at input En (Enable), and if 'Switch=On' was confirmed in parameter assignment mode.
	Parameter	Programming mode: Selecting the function for pushbutton action for the duration of one cycle, or for switching action. Start: On or off state, initialized at the first start of the program if retentivity is disabled. Retentivity: / = No retentivity R = The status is retentive. Parameter assignment mode (RUN mode): Switch: Switches the momentary pushbutton (switch) on or off.
	Output Q	Switches on if En=1 and Switch=On was confirmed with OK .

Factory setting

The default parameter setting is switching action.

Timing diagram



Functional description

In parameter assignment mode, the output is set with a signal at input En, if the "Switch" parameter is set to 'On' and confirmed with **OK**. Whether the function was configured for pushbutton or switching action is of no concern here.

The output is reset to '0' in the following three cases:

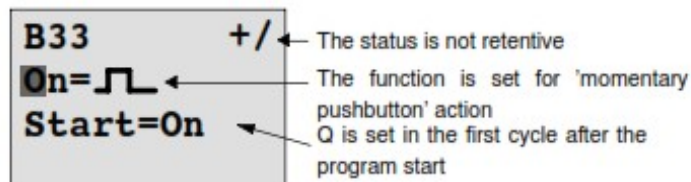
- After a 1 to 0 transition at input En
- When the function was configured for momentary pushbutton action, and one cycle has expired since it was switched on
- When the position 'Off' was selected at the 'Switch' parameter and confirmed with **OK** in parameter assignment mode

If retentivity is not set, output Q is initialized after a power failure according to your configuration at the "Start" parameter.

Setting the Par parameter

View in programming mode (example):

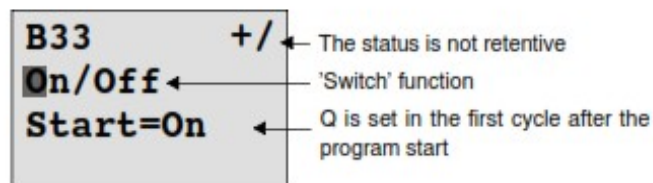
1. Select the 'Softkey' function.
2. Select input En and confirm with **OK**. The cursor is now positioned below 'Par'.
3. Change to the input mode of 'Par':
Confirm with **OK**
(the cursor is now positioned to 'On')



To change 'Par' to 'Switch' action and the initialization status after the program start:

4. To select 'Momentary pushbutton' or 'Switch' action:

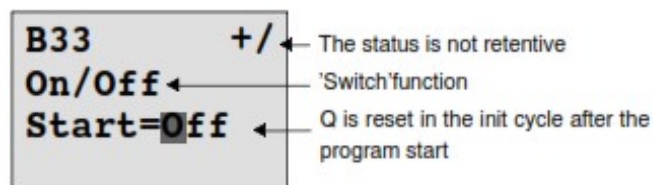
Press ▲ or ▼



5. To change to the start state:
Press ◀ or ▶

6. To change the start state:

Press ▲ or ▼

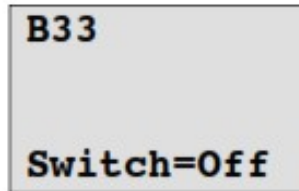


7. Confirm your entries with

OK

View in parameter assignment mode (example):

Here, you can set or reset the 'Switch' parameter (On/Off). When in RUN, LOGO! shows the following display:

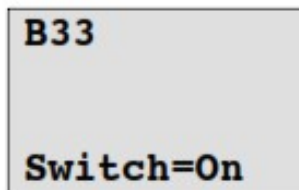


The pushbutton/switch here is switched off

Let us assume you want to set 'Switch' (On).

1. Change to the editing mode:
Confirm with **OK**
(the cursor is now positioned on 'Off')
2. To change from 'Off' to 'On':
Press ▲ or ▼
3. Confirm your entries with

Press **OK**

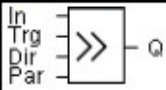


The momentary pushbutton/switch is here switched on

4.4.25 Shift register

Short description

You can use the shift register function to read the value of an input and to shift its bits left or right. The output value corresponds with the configured shift register bit. The shifting direction can be changed at a special input.

Symbol in LOGO!	Wiring	Description
	Input In	Input read at the start of the function.
	Input Trg	A positive edge (0 to 1 transition) at input Trg (Trigger) starts the special function. 1 to 0 transitions are irrelevant.
	Input Dir	The signal at input Dir determines the shifting direction for the shift register bits S1...S8: Dir = 0: Shift up (S1 >> S8) Dir = 1: Shift down (S8 >> S1)
	Parameter	Shift register bit that determines the value at output Q. Possible settings: S1 ... S8 Retentivity: / = No retentivity R = The status is retentive.
	Output Q	The output value corresponds with the configured shift register bit.

Functional description

The function reads the value at input In with a positive edge (0 to 1 transition) at input Trg (Trigger).

This value is applied to shift register bit S1 or S8, depending on the shifting direction:

- Shift up: The value at input In is set at S1; the previous value at S1 is shifted to S2; the previous value at S2 is shifted to S3 etc.
- Shift down: The value at input In is set at S8; the previous value at S8 is shifted to S7; the previous value at S7 is shifted to S6 etc.

Output Q returns the value of the configured shift register bit.

If retentivity is disabled, the shift function restarts at S1 or S8 after a power failure. When enabled, retentivity always applies to all shift register bits.

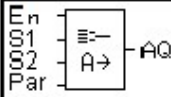
Note

The special function shift register can be used only once in the circuit program.

4.4.26 Analog Multiplexer

Short description

This special function outputs one of four predefined analog values or 0 at the analog output.

Symbol in LOGO!	Wiring	Description
	Input En	A change in status from 0 to 1 at input En (Enable) switches a parameterized analog value to the output AQ, depending on the value of S1 and S2.
	Inputs S1 and S2	S1 and S2 (selectors) for selecting the analog value to be issued. <ul style="list-style-type: none"> • S1 = 0 and S2 = 0: Value 1 is issued • S1 = 0 and S2 = 1: Value 2 is issued • S1 = 1 and S2 = 0: Value 3 is issued • S1 = 1 and S2 = 1: Value 4 is issued
	Parameter	V1...V4: Analog values that will be issued. Range of values: -32768...+32767 p: Number of decimals Range of values: 0, 1, 2, 3
	Output AQ	This special function has an analog output. This output can only be connected with the analog input of a function, an analog flag or an analog output connector (AQ1, AQ2). Range of values for AQ: -32768...+32767

Parameters V1...V4

The analog values for the parameters V1...V4 can be derived from another already-programmed function. You can use the actual values of the following functions:

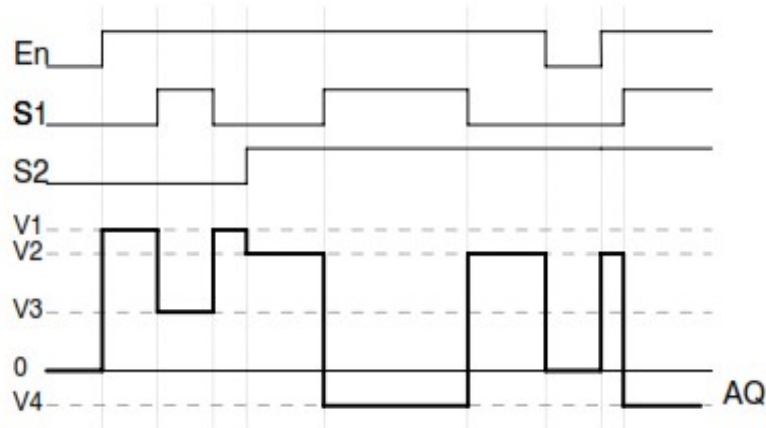
- Analog comparator (actual value Ax - Ay, see Chapter 4.4.18)
- Analog threshold trigger (actual value Ax, see Chapter 4.4.16)
- Analog amplifier (actual value Ax, see Chapter 4.4.20)
- Analog multiplexer (actual value AQ)
- Analog ramp (actual value AQ, see Chapter 4.4.27)
- Analog math (actual value AQ, see Chapter 4.4.30)
- PI controller (actual value AQ, see Chapter 4.4.28)
- Up/down counter (actual value Cnt, see Chapter 4.4.13)

Select the required function by the block number. For information on parameter defaults, refer to Chapter 4.4.1.

Parameter p (number of decimals)

Applies only to the values displayed in a message text.

Timing diagram



Functional description

If input En is set, then the function issues one of 4 possible analog values V1 to V4 at the output AQ, depending on the value of S1 and S2.

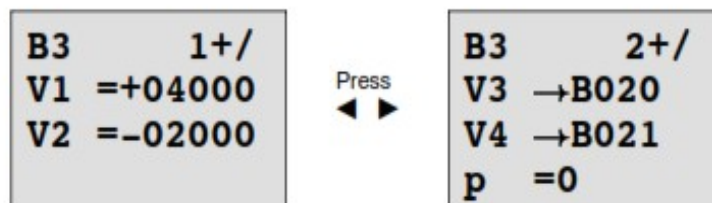
If the input En is not set, then the function issues the analog value 0 at output AQ.

Analog output

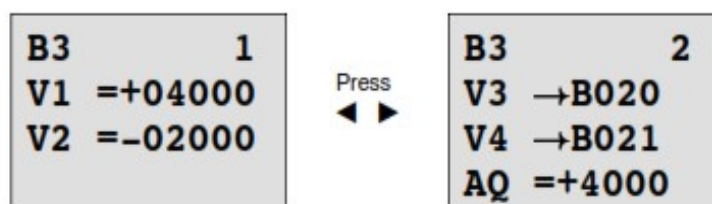
If you interconnect this special function with a real analog output, note that the analog output can only process values between 0 and 1000. To do this, you may need to connect an additional amplifier between the analog output of the special function and the real analog output. Using this amplifier, you standardize the output range of the special function to a value range of 0 to 1000.

Setting the Par parameter

View in programming mode (example):



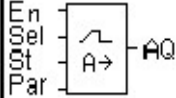
View in parameter assignment mode:



4.4.27 Analog Ramp

Short description

The Analog Ramp instruction allows the output to be changed from the current level to the selected level at a specified rate.

Symbol in LOGO!	Wiring	Description
	Input En	<p>A change in the status from 0 to 1 at input En (Enable) applies the start/stop level (Offset "B" + StSp) to the output for 100 ms and starts the ramp operation to the selected level.</p> <p>A change in the status from 1 to 0 immediately sets the current level to Offset "B", which makes output AQ equal to 0.</p>
	Input Sel	<p>Sel = 0: Level 1 is selected</p> <p>Sel = 1: Level 2 is selected</p> <p>A change in status of Sel causes the current level to start changing to the selected level at the specified rate.</p>
	Input St	<p>A change in the status from 0 to 1 at input St (Decelerated Stop) causes the current level to decrease at a constant rate until the start/stop level (Offset "B" + StSp) is reached. The start/stop level is maintained for 100ms and then the current level is set to Offset "B", which makes output AQ equal to 0.</p>

Symbol in LOGO!	Wiring	Description
	Parameter	<p>Level 1 and Level 2: Levels to be reached Range of values for each level: -10,000 to +20,000</p> <p>MaxL: Maximum value that must not be exceeded under any circumstances. Range of values: -10,000 to +20,000</p> <p>StSp: Start/Stop offset: value that is added to Offset "B" to create the start/stop level. If the Start/Stop offset is 0, then the start/stop level is Offset "B".. Range of values: 0 to +20,000</p> <p>Rate: Acceleration with which level 1, level 2 or Offset is reached. Steps/seconds are issued. Range of values: 1 to 10,000</p> <p>A: Gain Range of values: 0 to 10.00</p> <p>B: Offset Range of values: $\pm 10,000$</p> <p>p: Number of decimals Range of values: 0, 1, 2, 3</p>
	Output AQ	<p>The output AQ is scaled using the formula: Range of values for AQ: 0 to +32767</p> <p>$(\text{Current Level} - \text{Offset "B"}) / \text{Gain "A"}$ Range of values: 0 to +32767</p> <p>Note: When AQ is displayed in parameter mode or message mode, it is displayed as an unscaled value (engineering units: current level).</p>

Parameters L1, L2

The analog values for the parameters L1 and L2 can be derived from another already-programmed function. You can use the actual values of the following functions:

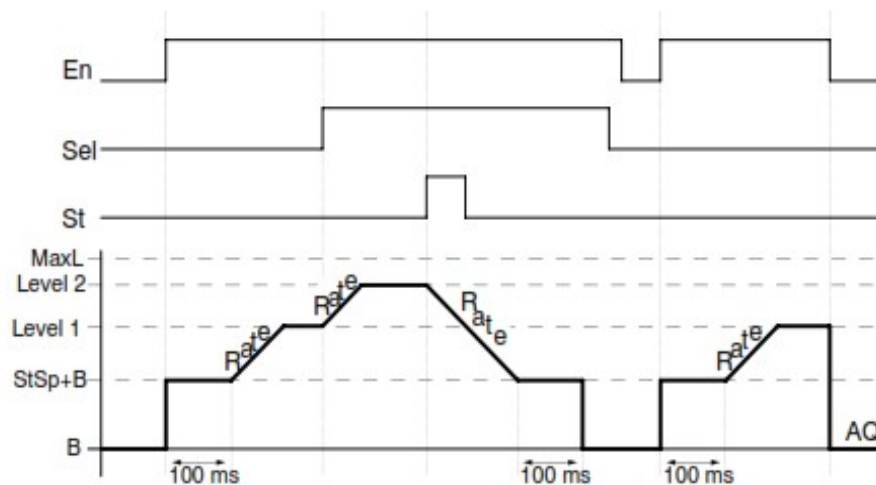
- Analog comparator (actual value Ax - Ay, see Chapter 4.4.18)
- Analog threshold trigger (actual value Ax, see Chapter 4.4.16)
- Analog amplifier (actual value Ax, see Chapter 4.4.20)
- Analog multiplexer (actual value AQ, see Chapter 4.4.26)
- Analog ramp (actual value AQ)
- Analog math (actual value AQ, see Chapter 4.4.30)
- PI controller (actual value AQ, see Chapter 4.4.28)
- Up/down counter (actual value Cnt, see Chapter 4.4.13)

Select the required function by the block number. For information on parameter defaults, refer to Chapter 4.4.1.

Parameter p (number of decimals)

Applies only to the AQ, L1, L2, MaxL, StSp and Rate values displayed in a message text.

Timing diagram for AQ



Functional description

If the input En is set, then the function sets the current level to StSp + Offset "B" for 100 ms.

Then, depending on the connection of Sel, the function runs from the level StSp + Offset "B" to either level 1 or level 2 at the acceleration set in Rate.

If the input St is set, the function runs to a level of StSp + Offset "B" at the acceleration set in Rate. Then the function holds the level at StSp + Offset "B" for 100 ms. After 100 ms, the level is set to Offset "B". The scaled value (output AQ) is 0.

If the input St is set, the function can only be restarted after the inputs St and En have been reset.

If input Sel has been changed, depending on the connection of Sel, the function runs from the current target level to the new target level at the rate that is specified.

If the input En is reset, the function immediately sets the current level to Offset "B".

The current level is updated every 100 ms. Note the relationship between output AQ and the current level:

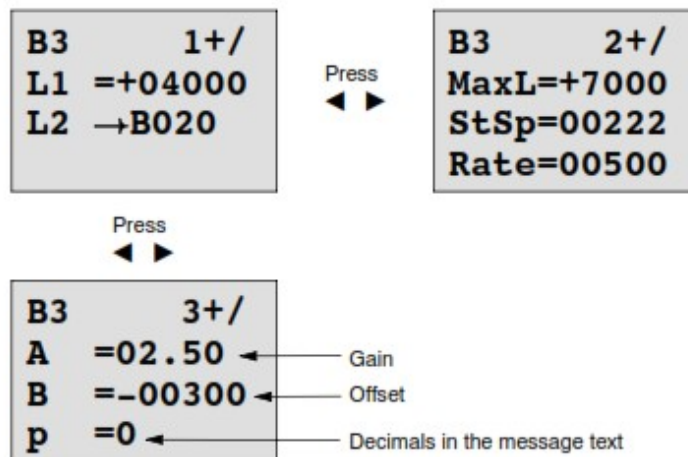
$$\text{Output AQ} = (\text{current level} - \text{Offset "B"} / \text{Gain "A"})$$

Note

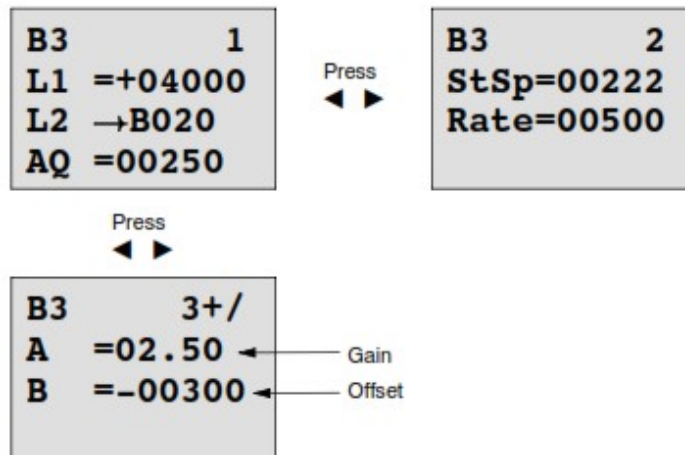
For further information on analog value processing please refer to the online help for LOGO!Soft Comfort.

Setting the Par parameter

View in programming mode (example):



View in parameter assignment mode:



4.4.28 PI controller

Short description

Proportional-action and integral-action controllers. You can use both types of controller individually or combined.

Symbol in LOGO!	Wiring	Description
	Input A/M	Set the mode of the controller: 1: automatic mode 0: manual mode
	Input R	Use the input R to reset the output AQ. As long as this input is set, the input A/M is disabled. Output AQ is set to 0.
	Input PV	Analog value: process variable, influences the output

Symbol in LOGO!	Wiring	Description
	Parameter	<p>SP: Set-value assignment Range of values: -10,000 to +20,000</p> <p>KC: Gain Range of values: 00.00 to 99.99</p> <p>Ti: Integral time Range of values: 00:01 to 99:59 m</p> <p>Dir: Action direction of the controller Range of values: + or -</p> <p>Mq: Value from AQ with manual mode Range of values: 0 to 1000</p> <p>Min: Minimum value for PV Range of values: -10,000 to +20,000</p> <p>Max: Maximum value for PV Range of values: -10,000 to +20,000</p> <p>A: Gain Range of values: ± 10.00</p> <p>B: Offset Range of values: $\pm 10,000$</p> <p>p: Number of decimals Range of values: 0, 1, 2, 3</p>
	Output AQ	<p>This special function has an analog output (= manipulated variable). This output can only be connected with the analog input of a function, an analog flag or an analog output connector (AQ1, AQ2). Range of values for AQ: 0...1000</p>

Parameters SP and Mq

The set-value SP and the value for Mq can be provided by another already-programmed function. You can use the actual values of the following functions:

- Analog comparator (actual value Ax - Ay, see Chapter 4.4.18)
- Analog threshold trigger (actual value Ax, see Chapter 4.4.16)
- Analog amplifier (actual value Ax, see Chapter 4.4.20)
- Analog multiplexer (actual value AQ, see Chapter 4.4.26)
- Analog ramp (actual value AQ, see Chapter 4.4.27)
- Analog math (actual value AQ, see Chapter 4.4.30)
- PI controller (actual value AQ)
- Up/down counter (actual value Cnt, see Chapter 4.4.13)

Select the required function by the block number. For information on parameter defaults, refer to Chapter 4.4.1.

Parameters KC, TI

Please note:

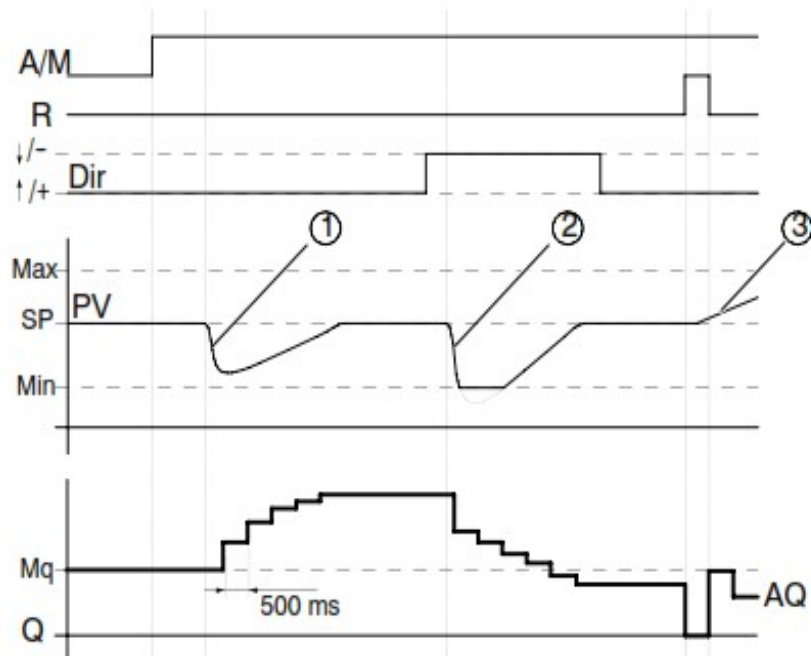
- If parameter KC has value 0, the "P" function (proportional control) will not be executed.
- If parameter TI has value 99:59 m, the "I" function (integral-action control) will not be executed.

Parameter p (number of decimals)

Applies only to the PV, SP, Min and Max values displayed in a message text.

Timing diagram

The nature, manner and speed with which the AQ changes depend on the parameters KC and TI. Thus, the course of AQ in the diagram is merely an example. A control action is continuous; therefore the diagram portrays just an extract.



1. A disturbance causes the PV to drop, as Dir is positioned upwards, AQ increases until PV corresponds again to SP.
2. A disturbance causes the PV to drop, as Dir is positioned downwards, AQ decreases until PV corresponds again to SP.
It is not possible to change the direction (Dir) at runtime of the function. The change is shown here for illustrative purposes only.
3. As AQ is set to 0 by means of the input R, PV changes. This is based on the fact that PV increases, which on account of Dir = upwards causes AQ to drop.

Functional description

If the input A/M is set to 0, then the special function issues output AQ with the value that you set with parameter Mq.

If the input A/M is set to 1, then automatic mode commences. As an integral sum the value Mq is adopted, the controller function begins the calculations.

Note

For further information on the controller basics please refer to the online help for LOGO!Soft Comfort.

The updated value PV is used to calculate in the formulas:

$$\text{Updated value PV} = (\text{PV} \cdot \text{gain}) + \text{offset}$$

- If the updated value PV = SP, then the special function does not change the value of AQ.
- Dir = upwards (+) (timing diagram numbers 1. and 3.)
 - If the updated value PV > SP, then the special function reduces the value of AQ.
 - If the updated value PV < SP, then the special function increases the value of AQ.
- Dir = downwards (-) (timing diagram number 2.)
 - If the updated value PV > SP, then the special function increases the value of AQ.
 - If the updated value PV < SP, then the special function reduces the value of AQ.

With a disturbance, AQ continues to increase / decrease until the updated value PV again corresponds to SP. The speed with which AQ changes depends on the parameters KC and TI.

If the input PV exceeds the parameter Max, then the updated value PV is set to the value of Max. If the PV falls short of the parameter Min, then the updated value PV is set to the value of Min.

If the input R is set to 1, then the AQ output is reset. As long as R is set, the input A/M is disabled.

Sampling time

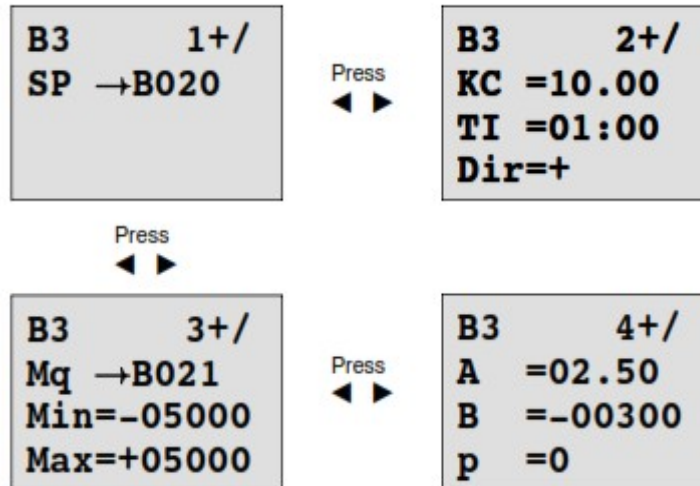
The sampling time is fixed at 500 ms.

Parameter sets

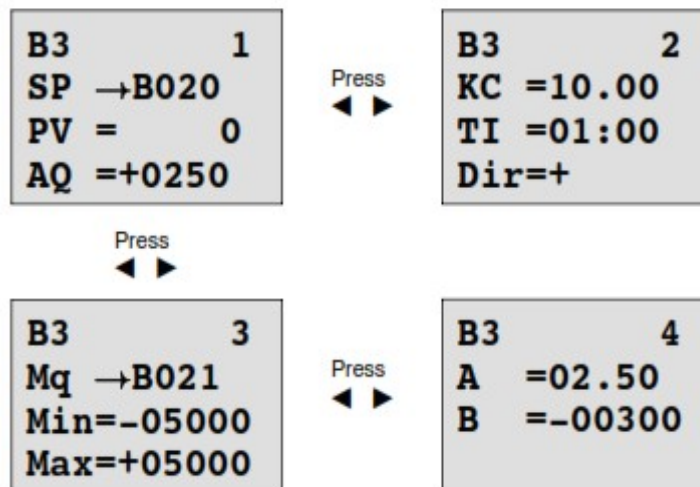
For more information and application examples with application-related parameter sets for KC, TI and Dir refer to the online help for LOGO!Soft Comfort.

Setting the Par parameter

View in programming mode (example):



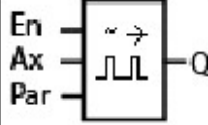
View in parameter assignment mode:



4.4.29 Pulse Width Modulator (PWM)

Short description

The Pulse Width Modulator (PWM) instruction modulates the analog input value A_x to a pulsed digital output signal. The pulse width is proportional to the analog value A_x .

Symbol in LOGO!	Wiring	Description
	Input A_x	Analog signal to be modulated to a pulsed digital output signal.
	Parameter	A: Gain Range of values: +- 10.00 B: Zero offset Range of values: +- 10,000 T: Periodic time over which the digital output is modulated p: Number of decimals Range of values: 0, 1, 2, 3
	Output Q	Q is set or reset for the proportion of each time period according to the proportion of the standardized value A_x to the analog value range.

Parameter T

Note the defaults of the T parameters listed in Chapter 4.3.2.

The periodic time T can be provided by the actual value of another already-programmed function. You can use the actual value of the following functions:

- Analog comparator (actual value $A_x - A_y$, see Chapter 4.4.18)
- Analog threshold trigger (actual value A_x , see Chapter 4.4.16)
- Analog amplifier (actual value A_x , see Chapter 4.4.20)
- Analog multiplexer (actual value A_Q , see Chapter 4.4.26)
- Analog ramp (actual value A_Q , see Chapter 4.4.27)
- Analog math (actual value A_Q , see Chapter 4.4.30)
- PI controller (actual value A_Q , see Chapter 4.4.28)
- Up/down counter (actual value Cnt, see Chapter 4.4.13).

Select the required function by the block number. The timebase is configurable. For information on valid ranges and parameter defaults, refer to Chapter 4.4.1.

Parameters p (number of decimals)

Parameter p applies only to the display of the A_x value in a message text

Functional description

The function reads the value of the signal at the analog input Ax.

This value is multiplied by the value of parameter A (gain). Parameter B (offset) is added to the product, as follows:

$$(Ax * Gain) + Offset = Actual\ value\ Ax$$

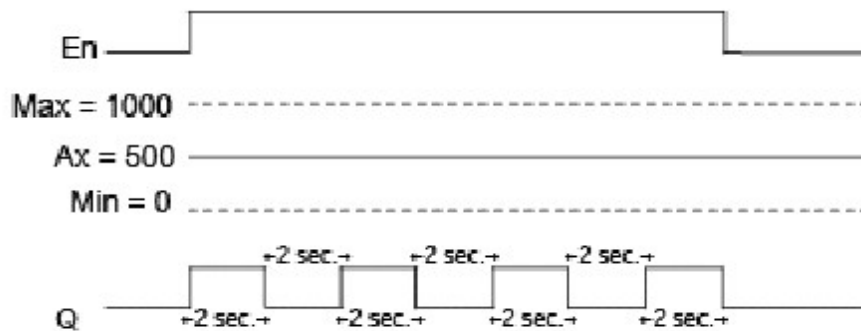
The function block calculates the proportion of the actual value Ax to the range.

The block sets the digital output Q high for the same proportion of the T (periodic time) parameter, and sets Q low for the remainder of the time period.

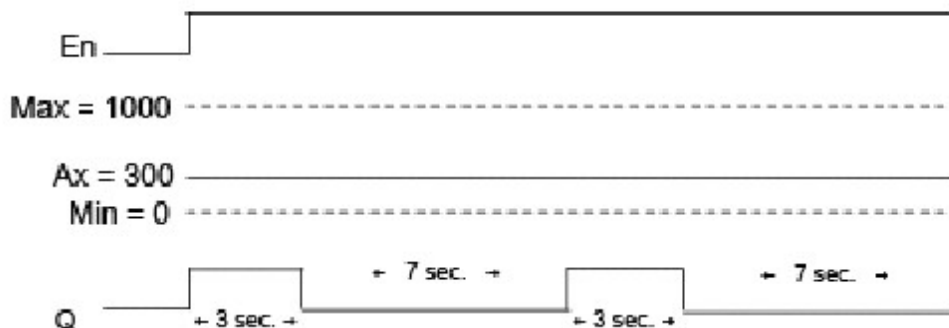
Examples with Timing Diagrams

The following examples show how the PWM instruction modulates a digital output signal from the analog input value:

1. An analog value of 500 (range 0...1000) as the value for Ax must be modulated to a digital signal string. The user-defined T (periodic time) parameter is 4 seconds. At the digital output of the PWM function the digital signal string is 2 seconds high, 2 seconds low, 2 seconds high, 2 seconds low and continues in that pattern as long as parameter En = high.



2. An analog value of 300 (range 0...1000) as the value for Ax must be modulated to a digital signal string. The user-defined T (periodic time) parameter is 10 seconds. At the digital output of the PWM function the digital signal string is 3 seconds high, 7 seconds low, 3 seconds high, 7 seconds low and continues in that pattern as long as parameter "En" = high.



Calculation rule

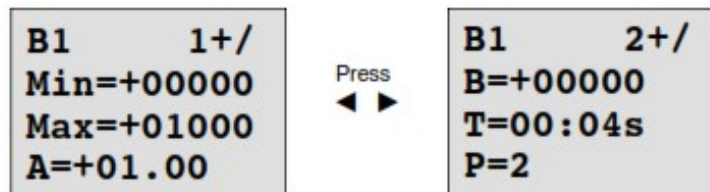
$Q = 1$, for $(Ax - \text{Min}) / (\text{Max} - \text{Min})$ of time period T , when $\text{Min} < Ax < \text{Max}$

$Q = 0$, for $PT - [(Ax - \text{Min}) / (\text{Max} - \text{Min})]$ of time period T .

Note: Ax in this calculation refers to the actual value Ax as calculated using the Gain and Offset.

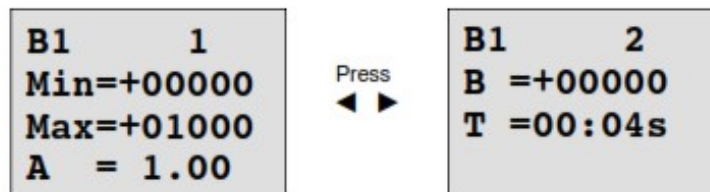
Setting the Par Parameter

The following illustration shows the view in programming mode that corresponds to the first example:



Use the ◀ and ▶ keys to navigate to the Min, Max, A, B, T and P parameters. For each digit of a value, use the ▲ and ▼ keys to scroll through value choices. Use the ▶ key to navigate to the second screen from the last line of the first screen, and the ◀ key to navigate from the top line of the second screen to the first screen. Use the OK key to accept changes.

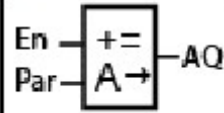
View in parameter assignment mode:



4.4.30 Analog math

Short description

The analog math block calculates the value AQ of an equation formed from the user-defined operands and operators.

Symbol in LOGO!	Wiring	Description
	Input En	A change in the status from 0 to 1 at input En (Enable) enables the analog math function block.
	Parameter	V1: First operand value V2: Second operand value V3: Third operand value V4: Fourth operand value Op1: First operator Op2: Second operator Op3: Third operator Pr1: Priority of first operation Pr2: Priority of second operation Pr3: Priority of third operation Gen→0: 0: Reset value of AQ to 0 when En=0 1: Retain last value of AQ when En=0 p: Number of decimals Range of values: 0, 1, 2, 3
	Output AQ	The output AQ is the result of the equation formed from the operand values and operators. AQ will be set to 32767 if a divide by 0 or overflow occurs, and -32768 if a negative overflow (underflow) occurs.

Parameters V1...V4

The analog values for the parameters V1...V4 can be derived from another already-programmed function. You can use the actual values of the following functions:

- Analog comparator (actual value Ax - Ay, see Chapter 4.4.18)
- Analog threshold trigger (actual value Ax, see Chapter 4.4.16)
- Analog amplifier (actual value Ax, see Chapter 4.4.20)
- Analog multiplexer (actual value AQ)
- Analog ramp (actual value AQ, see Chapter 4.4.27)
- Analog math (actual value AQ, see Chapter 4.4.30)
- PI controller (actual value AQ, see Chapter 4.4.28)
- Up/down counter (actual value Cnt, see Chapter 4.4.13).

Select the required function by the block number. For information on parameter defaults, refer to Chapter 4.4.1.

Parameters p (number of decimals)

Parameter p applies only to the display of Value1, Value2, Value3, Value4 and AQ in a message text.

Functional Description

The analog math function combines the four operands and three operators to form an equation. The operator can be any one of the four standard operators: +, -, *, or /. For each operator, you must set a unique priority of High (H), Medium(M), or Low(L). The high operation will be performed first, followed by the medium operation, and then by the low operation. You must have exactly one operation of each priority. The operand values can reference another previously-defined function to provide the value.

The number of operand values is fixed at four and the number of operators is fixed at 3. If you need to use fewer operands, use constructions such as + 0 or * 1 to fill the remaining parameters.

You can also configure the behavior of the function when the Enable parameter $En=0$. The function block can either retain its last value, or be set to 0. If the parameter $Qen \rightarrow 0 = 0$, then the function sets AQ to 0 when $En=0$. If the parameter $Qen \rightarrow 0 = 1$, then the function leaves AQ at its last value when $En=0$.

Possible errors: Zero division and overflow

If the analog math function block execution results in zero division or overflow, it sets internal bits that indicate the type of error that occurred. You can program an analog math error detection function block in your circuit program to detect these errors, and to control the program behavior as needed. You program one analog math error detection function block to reference one specific analog math function block.

Examples

The following tables show some simple example analog math block parameters, and the resulting equations and output values:

V1	Op1 (Pr1)	V2	Op2 (Pr2)	V3	Op3 (Pr3)	V4
12	+ (M)	6	/ (H)	3	- (L)	1

Equation: $(12 + (6 / 3)) - 1$

Result: 13

V1	Op1 (Pr1)	V2	Op2 (Pr2)	V3	Op3 (Pr3)	V4
2	+ (L)	3	* (M)	1	+ (H)	4

Equation: $2 + (3 * (1 + 4))$

Result: 17

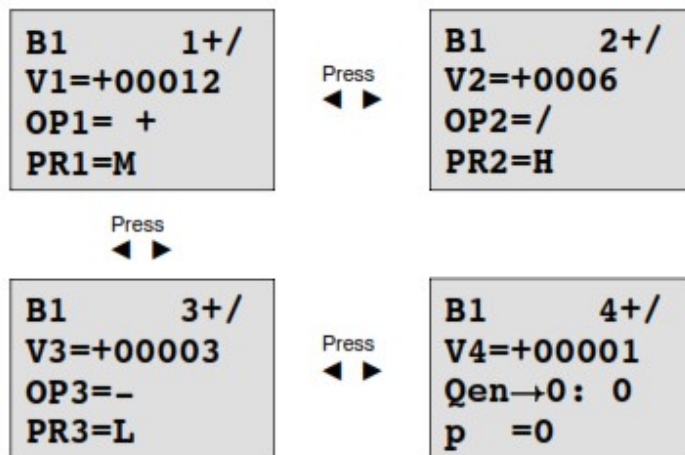
V1	Op1 (Pr1)	V2	Op2 (Pr2)	V3	Op3 (Pr3)	V4
100	- (H)	25	/ (L)	2	+ (M)	1

Equation: $(100 - 25) / (2 + 1)$

Result: 25

Setting the Par parameter

The following illustration shows the view in programming mode that corresponds to the first example $(12 + (6 / 3)) - 1$:

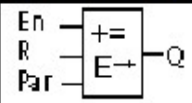


Use the ◀ and ▶ keys to navigate between the operand value, operator, and operation priority. To change a value, use the ▲ and ▼ keys to scroll through value choices for each value. Use the ◀ key to navigate from one screen to the previous screen when the cursor is on the V1..V4 line, and the ▶ key to navigate to the next screen from the PR1..PR3 line. Use the OK key to accept changes.

4.4.31 Analog math error detection

Short description

The analog math error detection block sets an output if an error has occurred in the referenced analog math function block.

Symbol in LOGO!	Wiring	Description
	Input En	A change in the status from 0 to 1 at input En (Enable) enables the analog math error detection block.
	Input R	A signal at input R resets the output.
	Parameter	MathBN: block number of an analog math instruction Err: ZD: Divide by 0 error OF: Overflow error ZD/OF: (Divide by 0 error) OR (Overflow error) AutoRst: Reset the output before the next execution of the analog math error function block. Y = yes; N = no
	Output Q	Q is set high if the error to detect occurred in the last execution referenced analog math function block

Parameter MathBN

The value for the MathBN parameter references be the block number of an already-programmed analog math function block.

Functional description

The analog math error detection block sets the output when the referenced analog math function block has an error. You can program the function to set the output on a zero division error, an overflow error, or when either type of error occurs.

If AutoRst is set, the output is reset prior to the next execution of the function block. If AutoRst is not set, then whenever the output is set it remains set until the analog math error detection block is reset with the R parameter. This way, even if the error subsequently clears, the circuit program still has knowledge that an error did occur at some point.

In any scan cycle, if the referenced analog math function block executes before the analog math error detection function block, the error is detected in the same scan cycle. If the referenced analog math function block executes after the analog math error detection function block, the error is detected in the next scan cycle.

Analog math error detection logic table

In the table below, Err represents the parameter of the analog math error detection instruction that selects which type of error to detect. ZD represents the zero division bit set by the analog math instruction at the end of its execution: 1 if the error occurred, 0 if not. OF represents the overflow bit set by the analog math instruction: 1 if the error occurred, 0 if not. The ZD/OF Err parameter represents the logical OR of the zero division bit and overflow bit of the referenced analog math instruction. Q represents the output of the analog math error detection function. An "x" indicates that the bit can be either 0 or 1 with no influence on the output.

Err	ZD	OF	Q
ZD	1	x	1
ZD	0	x	0
OF	x	1	1
OF	x	0	0
ZD/OF	1	0	1
ZD/OF	0	1	1
ZD/OF	1	1	1
ZD/OF	0	0	0

If the MathBN parameter is null, then the output Q is always 0.

Setting the Par parameter

The parameters MathBN, AutoRst, and Err can be set in programming mode or parameter assignment mode.

View in programming mode (example):

B3 **+ /**
MathBN=B001
AutoRst=N
Err=ZD/OF

- ← Block number of an already-programmed analog math instruction
- ← Auto Reset (Y or N)
- ← ZD, OF, or ZD/OF

Use the ◀ and ▶ keys to navigate between the MathBN, AutoRst, and Err parameters. To change a value, use the ▲ and ▼ keys to scroll through value choices for each value. Use the OK key to accept changes.

View in parameter assignment mode (example):

B3
MathBN=B001
AutoRst=N
Err=ZD/OF

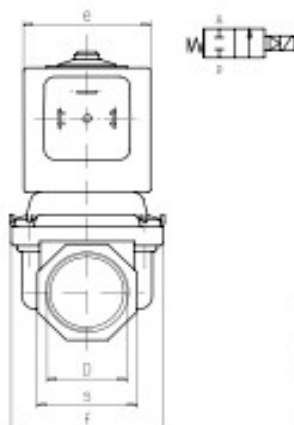
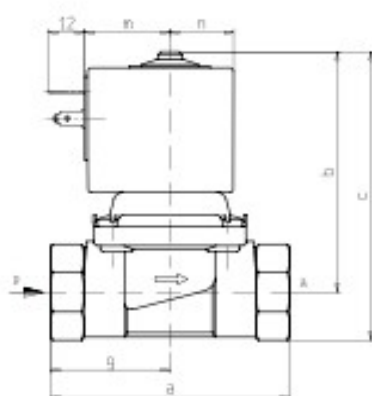
- ← Block number of an analog math instruction
- ← Auto Reset (Y or N)
- ← ZD, OF, or ZD/OF

ΠΑΡΑΡΤΗΜΑ Β



SOLENOID VALVE 2/2 - NC (Normally closed) Pilot operated hung diaphragm G3/8 ÷ 1

L133



D	a	b	c	e	f	m	n	s	g
G 3/8	60	69	80	30	41	21,6	19,9	22	25,5
G 1/2	66	74,5	90	42	41	28	21	27	-
G 3/4	79	81	98	42	51	28	21	33	-
G 1	105	100	121	48,6	71	35	24,3	42	48

► GENERAL FEATURES

Pilot operated hung diaphragm valve with full orifice.
Designed for closed circuit hydraulic systems and for vessels draining.
Suitable to shut off liquid and gaseous fluids (verify the compatibility of fluid with materials in contact).

► TECHNICAL FEATURES

Maximum allowable pressure (PS) 16 bar
Opening time from ~100ms to ~150ms
Closing time from ~100ms to ~400ms
Fluid temperature -10°C +90°C (NBR)
0°C +130°C (FPM)
Max viscosity 5°E (~37 cStokes or mm²/s)

► MATERIALS IN CONTACT WITH FLUID

Body Brass
Sealing NBR or FPM
Internal components Stainless steel and brass
Seat Brass
Core tube Stainless steel
Shading coil Copper

► COIL

Continuous duty ED 100%
Encapsulation material PET (polyethylene terephthalate) fiberglass reinforced
Insulation class F (140 °C) on request class H (165 °C) - UL
ZA only: F (155 °C) on request class H (180 °C) - UL
Ambient temperature -10 C° +60 °C
ZA only: -10 C° +50 °C
Electric connections DIN 46340- 3 poles connectors (DIN 43650)
Protection degree IP 65 (EN 60529) with plug connector
Voltages DC 12-24V (+10% -5%)
AC 24V/50Hz - 110V/50Hz (120V/60Hz) - 230V/50Hz (+10% -15%)
(Other voltages and frequencies on request).

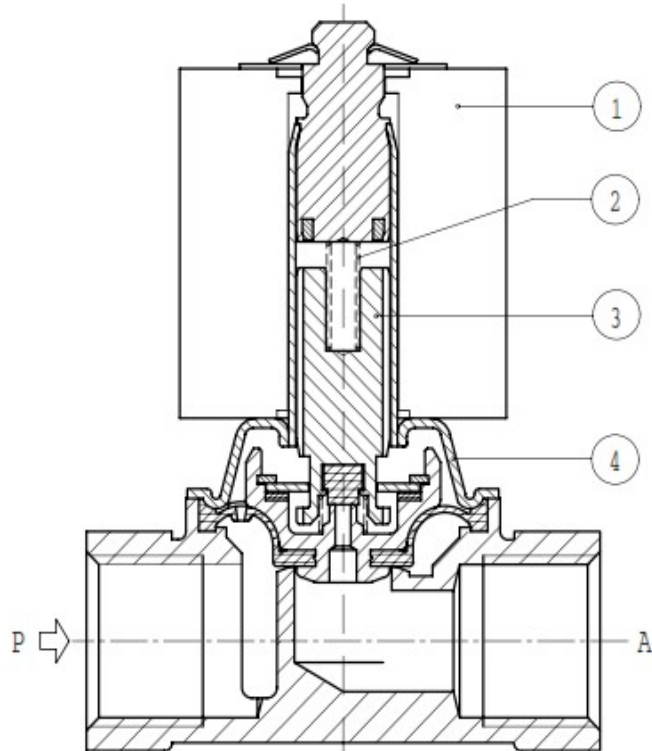
Port size ISO 228	Orifice size (mm)	Differential pressure (bar)				Kv (m ³ /h)	Series and type		Power absorption			Sealings	Notes	Weight (kg)
		Δp max					Valve	Coil	AC (VA)		c.c. (W)			
		Gas		Liquids					Inrush	Holding				
G 3/8	10	0	10	3	10	3	L133B10	ZA30A	23	14	9	1	0,440	
	L133V10													
G 1/2	12,5						L133B07	Z130A	44	24	13			NBR
	L133V07													
G 3/4	17						L133B07	Z923E	65	33	-			NBR
	L133V07													
G 1	24	10	-	10	-	9	L133B06	Z923A	-	-	17	-	1,610	
		-	3	-	3		L133V06							
		-	3	-	3		L133B06	L133V06						

► NOTES

- Sealings : NBR = Nitrile-butylene elastomer FPM = Fluoro-carbon elastomer
The nominal flow is guaranteed with Δp min ≥ 0,3 bar. Contact us in case of lower Δp min values.
1 - The valves fitted with 50Hz coil can operate at 60 Hz but in this case the Δp max corresponds to the nominal value less 20%.

L133

► SPARE PARTS



Kit description		Kit P.N.	Consisting of:
Core diaphragms kit	G 3/8 L133V10 G 3/8 L133B10 G 1/2 L133V07 G 1/2 L133B07 G 3/4 L133V07 G 3/4 L133B07 G 1 L133V06 G 1 L133B06	G3104302 G3104301 G2990102 G2990101 G2990202 G2990201 G2991902 G2991901	Core return spring pos.2 Diaphragm assembly with core assembly pos.3
Core return spring kit	G 3/8 L133B-V10 G 1/2 L133B-V07 G 3/4 L133B-V07 G 1 L133B-V06	G3103701 G2139901 G2918601 G2955801	n°10 core return springs pos. 2
Guide assembly	G 3/8 L133B-V10 G 1/2 L133B-V07 G 3/4 L133B-V07 G 1 L133B-V06	3104101R 3077801R 3077701R 2408202R	Guide assembly pos. 4
Coil	L133B-V10 L133B-V07 L133B-V06 (DC) L133B-V06 (AC)	ZA30A Z130A Z923A Z923E	Coil pos. 1

► INSTALLATION

Solenoid valve can be mounted in any position; vertical with coil upwards preferred.