

ΠΕΡΙΕΧΟΜΕΝΑ

1. ΓΝΩΡΙΜΙΑ ΜΕ ΤΗΝ ΣΥΣΚΕΥΗ LOGO!

Τα μοντέλα του LOGO!	1
Οι μονάδες επεκτάσεις του LOGO!	3
Η Μέγιστη σύνθεση του LOGO!	4
Ψηφιακές εισοδοί στο LOGO!	5
Αναλογικές εισοδοί στο LOGO!	6

2. ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΟΥ LOGO!

Γλώσσες προγραμματισμού του LOGO!.....	7
Η ΔΟΜΗ ΤΗΣ FBD.....	8
1. Σταθερές και Επαφές – Co.....	9
2. Βασικές Λειτουργίες – GF.....	10
3. Ειδικές Λειτουργίες – SF.....	16
ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΜΕΝΟΥ ΣΤΟ Logo Soft Comfort.....	19

3. Εφαρμογές με το Logo Soft Comfort

A. ΠΥΛΕΣ / GATES	24
B. ΧΡΟΝΙΚΑ / TIMERS	30
Γ. ΑΠΑΡΙΘΜΗΤΕΣ / COUNTERS.....	36
Δ. ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ / SET - RESET	43

4. Technical Data / Τεχνικά Χαρακτηριστικά

Πηγές.....	97
------------	----

Πρόλογος

Το αντικείμενο της πτυχιακής εργασίας μου είναι η μελέτη και η δημιουργία συστημάτων αυτοματισμού με την λογική μονάδα LOGO! της εταιρίας Siemens.

Η πτυχιακή χωρίζεται σε τέσσερα μέρη:

Το πρώτο μέρος αναφέρεται στο hardware κομμάτι του LOGO!, τις αναλογικές και ψηφιακές εισόδους του, τις επεκτάσεις του κλπ

Το δεύτερο μέρος αναφέρεται στον προγραμματισμό του LOGO! με το πρόγραμμα Logo Soft Comfort. Αναφέρονται οι γλώσσες προγραμματισμού LAD, FBD, η δομή της FBD, βασικές και ειδικές λειτουργίες της FBD.

Στο τρίτο μέρος έχουμε το πρακτικό κομμάτι αυτής της πτυχιακής με εφαρμογές πάνω στις Πύλες, Χρονικά, Απαριθμητές, Αυτοσυγκράτηση και διαφόρων άλλων λειτουργιών του LOGO!

Στο τέταρτο μέρος παραθέτω τα τεχνικά χαρακτηριστικά της συσκευής LOGO! καθώς και τις πηγές που χρησιμοποίησα για την περάτωση της εργασίας μου.

Ευχαριστίες

Στο σημείο αυτό θεωρώ υποχρέωση μου, να ευχαριστήσω τον επιβλέποντα καθηγητή αυτής της πτυχιακής κ Ν. Φραγκιαδάκη, για τη συνεχή καθοδήγηση, τις πολύτιμες συμβουλές και γενικά το αμέριστο ενδιαφέρον που επέδειξε καθ' όλη την διάρκεια εκπόνησης της εργασίας.

Abstract

The subject matter of my final project is the design and development of automatic control systems using the logic module LOGO! by the company Siemens.

The final project consists of four parts:

The first part refers to LOGO! hardware, its analog and digital inputs, its expansions, etc.

The second part refers to LOGO! programming using the program Logo Soft Comfort. The programming languages LAD, FBD, FBD structure, FBD general and special functions are covered.

The third part is the practical section of this final project with applications on Gates, Timers, Counters, Latching Relays and other LOGO! functions.

The fourth part contains the technical data of the LOGO! logic module as well as the sources used to complete this final project.

Acknowledgments

I would like to thank the supervising professor of this final project, Mr. N. Fragiadakis, for the continuous direction, valuable advice and generally the interest he has shown throughout the completion of this final project.

1. ΓΝΩΡΙΜΙΑ ΜΕ ΤΗΝ ΣΥΣΚΕΥΗ LOGO!

Τι είναι το *LOGO!*;

Το LOGO! είναι μια μικρή μονάδα λογικής απο την Siemens.

Το LOGO! παρέχει:

- Ενσωματωμένα πλήκτρα χειρισμών και φωτιζόμενη οθόνη
- Τροφοδοτικό
- Υποδοχή για εξωτερική μονάδα μνήμης και σύνδεση με H/Y
- Υποδοχή σύνδεσης μονάδων επέκτασης
- Ρολόι πραγματικού χρόνου
- Εισόδους/Εξόδους ανάλογα με το τύπο της συσκευής

Τι μπορεί να κάνει το LOGO!;

Το LOGO! μπορεί να χρησιμοποιηθεί σε πλήθος εφαρμογών όπως σε κτιριακές εγκαταστάσεις, για έλεγχο φωτισμού εσωτερικών ή εξωτερικών χώρων, για έλεγχο συστημάτων θέρμανσης και κλιματισμού, για έλεγχο αρδευτικών αντλιών, πυλών, θυρών, αυτοματισμούς σε μπάρες ασφάλειας κα.

Τα μοντέλα του LOGO!

Τα μοντέλα του LOGO! (συγκεκριμένα της σειράς 0BA5) διακρίνονται από την τροφοδοσία και τον τύπο εξόδων τους:

Τα μοντέλα του LOGO! – Βασικές συσκευές

Σύμβολο	Ονομασία	Τροφοδοσία	Είσοδοι	Έξοδοι	Χαρακτηριστικά
	LOGO! 12/24RC	12/24 V DC	8 ψηφιακές	4 ρελέ 230 V x 10 A	
	LOGO! 24	24 V DC	8 ψηφιακές	4 τρανζίστορ 24 V x 0.3 A	χωρίς ρολόι
	LOGO! 24RC	24 V AC	8 ψηφιακές	4 ρελέ 230 V x 10 A	
	LOGO! 230RC	115...240 V AC/DC	8 ψηφιακές	4 ρελέ 230 V x 10 A	
	LOGO! 12/24RCo	12/24 V DC	8 ψηφιακές	4 ρελέ 230 V x 10 A	χωρίς οθόνη/πλήκτρα
	LOGO! 24RCo	24 V AC	8 ψηφιακές	4 ρελέ 230 V x 10 A	χωρίς οθόνη/πλήκτρα
	LOGO! 230RCo	115...240 V AC/DC	8 ψηφιακές	4 ρελέ 230 V x 10 A	χωρίς οθόνη/πλήκτρα




Οι μονάδες επεκτάσεις του LOGO!

Το LOGO! είναι επεκτάσιμη μονάδα λογικής.

Αυτό σημαίνει ότι μπορούμε να επεκτείνουμε τη βασική συσκευή προσθέτοντας μονάδες επιπλέον εισόδων-εξόδων.

Στο LOGO! μπορούν να συνδεθούν οι παρακάτω μονάδες επέκτασης:

Μονάδες επέκτασης

Σύμβολο	Ονομασία	Τροφοδοσία	Είσοδοι	Έξοδοι
	LOGO! DM 8 12/24 R	12/24 V DC	4 ψηφιακές	4 ρελέ
	LOGO! DM 8 24	24 V DC	4 ψηφιακές	4 τρανζίστορ
	LOGO! DM 8 24 R	24 V AC/DC	4 ψηφιακές	4 ρελέ (5A)
	LOGO! DM 8 230R	115...240 V AC/DC	4 ψηφιακές	4 ρελέ
	LOGO! DM 16 24	24 V DC	8 ψηφιακές	8 τρανζίστορ 24V / 0.3A
	LOGO! DM 16 24R	24 V DC	8 ψηφιακές	8 ρελέ (5A)
	LOGO! DM 16 230R	115...240 V AC/DC	8 ψηφιακές	8 ρελέ (5A)
	LOGO! AM 2	12/24 V DC	2 αναλογικές 0-10 V ή 0-20 mA	καμμία
	LOGO! AM 2 PT100	12/24 V DC	2 Pt100 -50 °C ή +200 °C	καμμία
	LOGO! AM 2 AQ	24 V DC	καμία	2 αναλογικές 0 ... 10 V DC

Η Μέγιστη σύνθεση στο LOGO! φέρει τις 24 ψηφιακές εισόδους και 16 ψηφιακές εξόδους καθώς και 8 αναλογικές εισόδους:

Μέγιστη σύνθεση LOGO!

I1.....I6, I7, I8 AI1, AI2	I9...I12	I13...I16	I17...I20	I21...I24	AI3, AI4	AI5, AI6	AI7, AI8
LOGO! Basic	LOGO! DM8	LOGO! DM8	LOGO! DM8	LOGO! DM8	LOGO! AM2	LOGO! AM2	LOGO! AM2
Q1...Q4	Q5...Q8	Q9...Q12	Q13...Q16				

Ψηφιακές εισοδοι στο LOGO!

Οι ψηφιακές εισοδοι του LOGO! “αντιλαμβάνονται” (“ανιχνεύουν”, “αναγνωρίζουν”) δύο διακριτές καταστάσεις: την κατάσταση “ON” και την κατάσταση “OFF” που πάντα αντιστοιχούν στην κατάσταση του λογικού 1 και 0 αντίστοιχα και που διοχετεύονται ως πληροφορίες μέσω της διέλευσης (ή όχι) ηλεκτρικού σήματος.

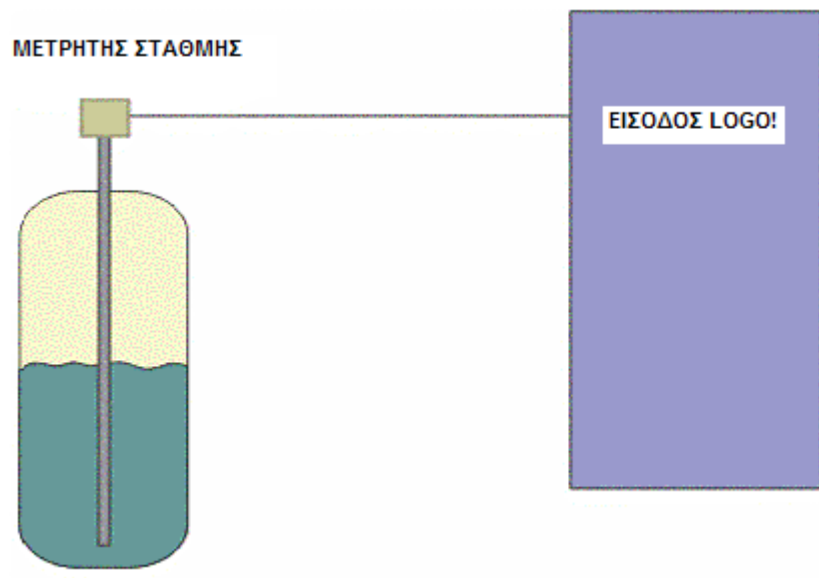
Στις ψηφιακές εισόδους του LOGO! μπορούμε να συνδέσουμε διαφόρων ειδών εξαρτήματα και υλικά που ανήκουν στην κατηγορία των αισθητηρίων (sensors) όπως μπουτόν, επαφές ρελέ, διακόπτες, τερματοδιακόπτες, διακόπτες προσέγγισης (proximity switch διαφόρων τύπων - χωρητικούς, επαγωγικούς κλπ.) φωτοκύτταρα και πλήθος ακόμα εξαρτήματα.

Αναλογικές εισόδους στο LOGO!

Σε αυτή την περίπτωση έχουμε το δεύτερο “είδος” εισόδων ενός LOGO! διαφορετικό από αυτό των ψηφιακών εισόδων. Οι αναλογικές εισόδους του LOGO! “αντιλαμβάνονται” (“ανιχνεύουν”, “αναγνωρίζουν”) όχι δύο διακριτές καταστάσεις – όπως στην περίπτωση των ψηφιακών εισόδων αλλά μια κατάσταση που συνεχώς μεταβάλλεται.

Ένα κλασσικό παράδειγμα είναι η μέτρηση στάθμης ενός υγρού υλικού σε μια δεξαμενή. Η μεταβαλλόμενη στάθμη του υγρού “μεταφράζεται” από το αισθητήριο σε ένα αντίστοιχα μεταβαλλόμενο ηλεκτρικό σήμα που κυμαίνεται σε μία τυποποιημένη κλίμακα έντασης ρεύματος (π.χ. 4 έως 20 mA) ή τάσης ρεύματος (π.χ. 0-10 V)

Η - ειδικού τύπου - αναλογική είσοδος του LOGO! “αντιλαμβάνεται” τις διαφοροποιήσεις (αυξομειώσεις του ηλεκτρικού σήματος-ρεύματος από π.χ. 4 έως 20 mA ή τάσης π.χ. 0-10 V) και τις “μεταφράζει” σε μεταβολές (αυξομειώσεις) του φυσικού φαινομένου, δηλαδή της στάθμης του υγρού.



Τα όσα αναφέρθηκαν πιο πάνω δεν πρέπει να δημιουργήσουν σύγχυση στον

σπουδαστή σε σχέση με τη βασική αρχή που λέει ότι το LOGO! “αντιλαμβάνεται” πληροφορίες μόνο στη μορφή “ON” - “OFF” ή λογικού 1 ή 0.

Απλά στην περίπτωση των αναλογικών σημάτων στη μονάδα των αναλογικών εισόδων υπάρχουν ειδικές ενδιάμεσες διατάξεις (ψηφιακά κυκλώματα) που ονομάζονται “Μετατροπείς Αναλογικών σε Ψηφιακά σήματα”

(Analog to Digital Converters / A/D Converters) που “μεταφράζουν” το συνεχές μεταβαλλόμενο αναλογικό σήμα με κωδικοποιημένο τρόπο (με αλληλουχίες συνδυασμών 0 και 1) σε ψηφιακό, σε αυτό δηλαδή που το LOGO! είναι σε θέση να “αντιληφθεί”.

2. ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΟΥ LOGO!

Γλώσσες προγραμματισμού του LOGO!

Το LOGO! προγραμματίζεται με 2 βασικές γλώσσες προγραμματισμού, την **LAD** (Ladder) και την **FBD** (Function Block Diagram).

Το LOGO! προγραμματίζεται με 2 τρόπους:

- Από την οθόνη του με την βοήθεια του ενσωματωμένου Cursor.
- Με τον H/Y με την βοήθεια του προγράμματος **Logo Soft Comfort**.

Για λόγους ευχρηστίας και δυνατοτήτων του *Logo Soft Comfort* επιλέγουμε τον προγραμματισμό του LOGO! με αυτό.

Η ΔΟΜΗ ΤΗΣ FBD

Για να μπορέσει ο σπουδαστής να δημιουργήσει ένα πρόγραμμα θα πρέπει να γνωρίζει την δομή της γλώσσας που προγραμματίζει.

Εμείς θα μάθουμε προγραμματίζουμε με την **FBD** καθώς είναι πιο πλούσια στην δομή της και πιο εύκολη στην κατανόηση από την **LAD**.

Η δομή της **FBD** έχει τις εξής κατηγορίες:

Σταθερές και Επαφές –**Co**

Βασικές Λειτουργίες –**GF**

Ειδικές Λειτουργίες –**SF**

1. Σταθερές και Επαφές – Co

Στην κατηγορία αυτή είναι οι εισοδοι, οι έξοδοι, τα βοηθητικά μνήμης

και οι σταθερές hi & lo.

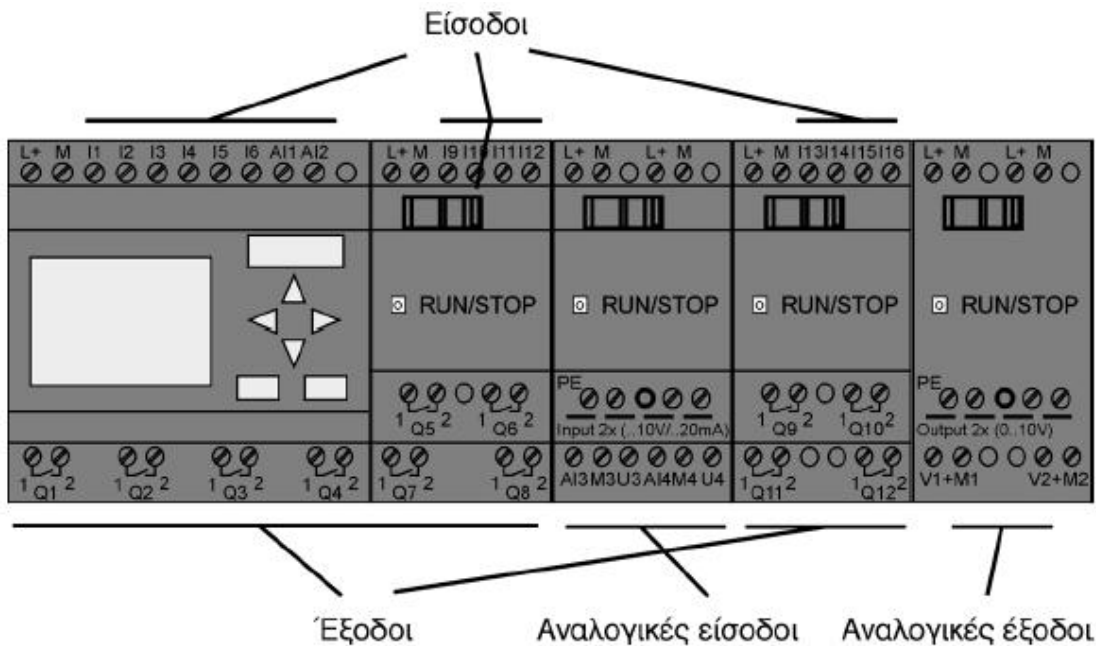
Είσοδοι:

1) Ψηφιακές εισοδοι

Οι ψηφιακές εισοδοι συμβολίζονται με το γράμμα **I**.
Οι αριθμοί των Ψηφιακών εισόδων (I1, I2, I3...) αντιστοιχούν στις κλέμες της βασικής συσκευής και των μονάδων επέκτασης, με την σειρά που έχουν τοποθετηθεί.

2) Αναλογικές εισοδοι

Οι Αναλογικές εισοδοι συμβολίζονται με το γράμμα **AI**.
Στην βασική συσκευή του LOGO! οι ψηφιακές εισοδοι I7 και I8 μπορούν να χρησιμοποιηθούν και ως αναλογικές εισοδοι AI1 και AI2 αν απαιτείται χρήση αναλογικής εισόδου.



Έξοδοι:

1) Ψηφιακές έξοδοι:

Οι ψηφιακές έξοδοι συμβολίζονται με το γράμμα **Q**.
Οι αριθμοί των Ψηφιακών εισόδων (Q1, Q2, Q3...) αντιστοιχούν στις κλέμες της βασικής συσκευής και των μονάδων επέκτασης, με την σειρά που έχουν τοποθετηθεί.

2) Αναλογικές έξοδοι:

Οι Αναλογικές έξοδοι συμβολίζονται με το γράμμα **AQ**.
Στο LOGO! μπορούμε να χρησιμοποιήσουμε μέχρι 2 αναλογικές εξόδους (AQ1, AQ2). Μία αναλογική έξοδος μπορεί να συνδεθεί μόνο με αναλογική είσοδο μιας λειτουργίας.

Βοηθητικά μνήμης:

Τα βοηθητικά μνήμης έχουν αντίστοιχη λειτουργία με αυτά των βοηθητικών ρελέ στα ηλεκτρικά κυκλώματα, συμβολίζονται με το γράμμα **M**. και είναι 24 ψηφιακά (M1..M24) και 6 αναλογικά (AM1..AM6).

Μνήμες καταχωριτή ολίσθησης

Οι μνήμες καταχωριτή ολίσθησης χρησιμοποιούνται μόνο για ανάγνωση (read-only) απο το πρόγραμμα, συμβολίζονται με το γράμμα **S** (S1..S8).

Σταθερές hi & lo

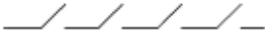
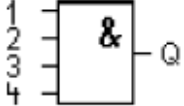
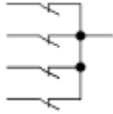




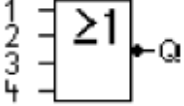

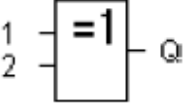
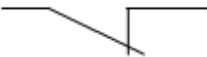
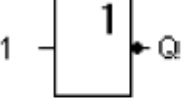
Υπάρχουν δύο σταθερές που συμβολίζονται με **hi** και **lo** και αντιστοιχούν σε κατάσταση λογικού "1" και λογικού "0" αντίστοιχα.

2. Βασικές Λειτουργίες – GF

Οι Βασικές λειτουργίες που περιέχει το LOGO! είναι μερικές από τις γνωστές στα ψηφιακά κυκλώματα πύλες.

Αυτές είναι οι AND, NAND, OR, NOR, XOR, NOT.

ΠΙΝΑΚΑΣ ΒΑΣΙΚΩΝ ΛΕΙΤΟΥΡΓΙΩΝ

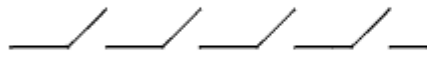
Διάγραμμα Ηλεκτρ. Κυκλώματ.	Αναπαράσταση στο LOGO!	Βασική Λειτουργία
		AND Κανονικά ανοιχτές επαφές (N.O.) συνδεδεμένες σε σειρά
		NAND Κανονικά κλειστές επαφές (N.C.) συνδεδεμένες παράλληλα
		OR Κανονικά ανοιχτές επαφές (N.O.) συνδεδεμένες παράλληλα
		NOR Κανονικά κλειστές επαφές (N.O.) συνδεδεμένες σε σειρά
		XOR Ταυτόχρονη αλλαγή κατάστασης επαφών
		NOT Αντιστροφείας

Σημείωση:

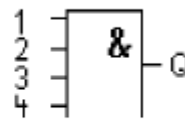
Κάθε μια από τις βασικές λειτουργίες αποτελεί ένα “ block ”.
Το LOGO! έχει διαθέσιμη μνήμη 130 block που αντιστοιχεί σε 2000 bytes.

AND

Οι εν σειρά συνδεδεμένες κανονικά ανοικτές επαφές συμβολίζονται ως εξής:



Σύμβολο LOGO!:



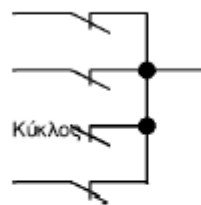
Στο block AND για να έχει η έξοδος (Q) την κατάσταση 1 πρέπει όλες οι είσοδοι να έχουν την κατάσταση 1.

Πίνακας Καταστάσεων του block AND:

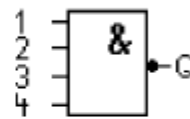
1	2	3	4	Q
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

NAND

Οι παράλληλα συνδεδεμένες κανονικά κλειστές επαφές συμβολίζονται ως εξής:



Σύμβολο LOGO!:



Στο block NAND η έξοδος (Q) έχει την κατάσταση 0 μόνο όταν οι είσοδοι I1 και I2 και I3 έχουν την κατάσταση 1.

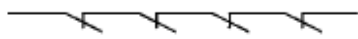
Στη NAND ισχύει: $x = 1$ (x : η είσοδος δεν χρησιμοποιείται).

Πίνακας καταστάσεων του block NAND

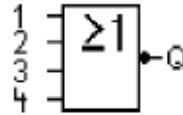
1	2	3	4	Q
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

OR

Οι παράλληλα συνδεδεμένες κανονικά ανοικτές επαφές συμβολίζονται ως εξής:



Σύμβολο LOGO!:



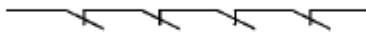
Στο block OR για να έχει η έξοδος (Q) την κατάσταση 1 πρέπει τουλάχιστον μία είσοδος να έχει την κατάσταση 1. Στην OR ισχύει: $x = 0$ (x : η είσοδος δεν χρησιμοποιείται).

Πίνακας καταστάσεων του block OR:

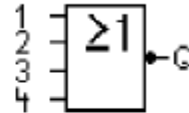
1	2	3	4	Q
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

NOR

Οι εν σειρά συνδεδεμένες κανονικά κλειστές επαφές συμβολίζονται ως εξής:



Σύμβολο LOGO!:



Στο block NOR η έξοδος έχει την κατάσταση 1 μόνο όταν **όλες** οι εισόδους έχουν την κατάσταση 0 και την κατάσταση 0 όταν μία από τις εισόδους ενεργοποιηθεί (γίνει 1).

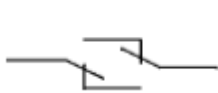
Στην NOR ισχύει: $x = 0$ (x : η είσοδος δεν χρησιμοποιείται).

Πίνακας Καταστάσεων του block NOR

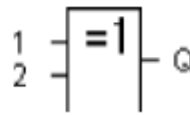
1	2	3	4	Q
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

XOR

Η ταυτόχρονη αλλαγή κατάστασης επαφών συμβολίζεται ως εξής:



Σύμβολο LOGO!:



Στο block XOR η έξοδος έχει την κατάσταση 1 όταν οι είσοδοι έχουν **διαφορετική** κατάσταση.

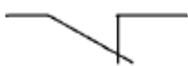
Στην XOR ισχύει: $x = 0$ (x : η είσοδος δεν χρησιμοποιείται).

Πίνακας Καταστάσεων του block XOR

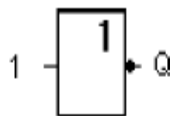
1	2	Q
0	0	0
0	1	1
1	0	1
1	1	0

NOT

Ο αντιστροφέας συμβολίζεται ως εξής:



Σύμβολο LOGO!:



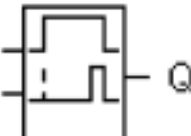
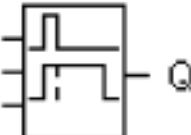

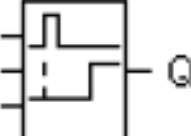
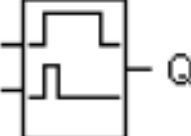


Στο block NOT η έξοδος (Q) έχει την κατάσταση 1 όταν η είσοδος έχει την κατάσταση 0 και αντίστροφα. Το block NOT, δηλαδή, αντιστρέφει την κατάσταση της εισόδου.

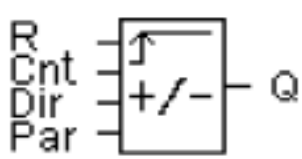
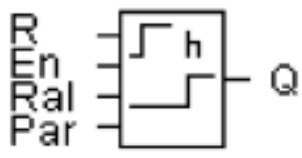

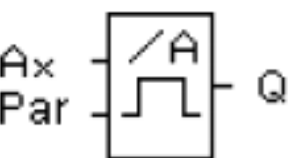
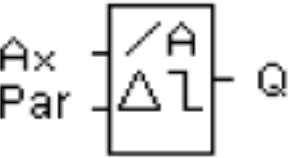
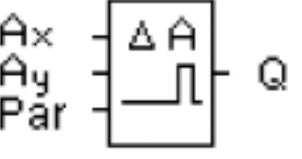
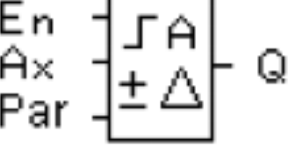
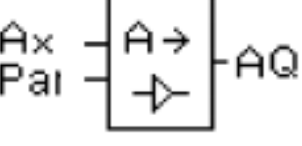
Έτσι στο LOGO! δε χρειάζονται κανονικά κλειστές επαφές. Αντί για αυτές μπορούν να χρησιμοποιηθούν κανονικά ανοιχτές επαφές και με το block NOT να μετατραπούν σε κανονικά κλειστές.

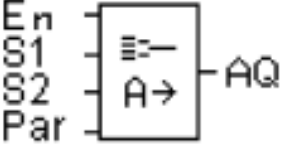
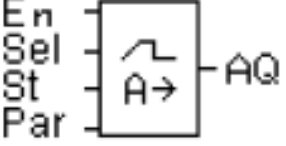

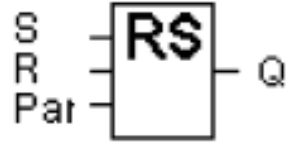
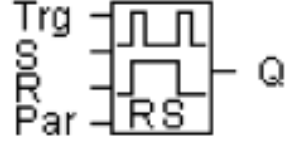
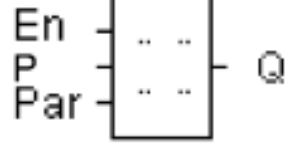
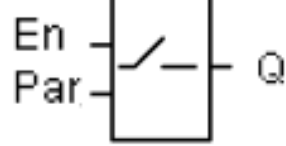

Πίνακας Καταστάσεων του block NOT

1	Q
0	1
1	0

ΠΙΝΑΚΑΣ ΕΙΔΙΚΩΝ ΛΕΙΤΟΥΡΓΙΩΝ

Αναπαράσταση στο LOGO!	Ειδική Λειτουργία
Χρόνοι	
Trg Par 	Χρονικό καθυστέρησης έλξης
Trg R Par 	Χρονικό καθυστέρησης πτώσης
Trg Par 	Χρονικό καθυστέρησης έλξης-πτώσης
Trg R Par 	Χρονικό καθυστέρησης έλξης με αυτοσυγκράτηση
Trg Par 	Χρονικό έναρξης-παύσης
Trg R Par 	Χρονικό έναρξης-παύσης με αναγνώριση αλλαγής κατάστασης
En Inv Par 	Γεννήτρια παλμοσειρών με ρύθμιση εύρους παλμού

Απαριθμήσεις	
	Απαριθμητής δύο κατευθύνσεων
	Ωρομετρητής λειτουργίας
	Διακόπτης συχνότητας
Αναλογικές τιμές	
	Αναλογικός μετρητής
	Αναλογικός διαφορικός μετρητής
	Αναλογικός συγκριτής
	Επιτήρηση αναλογικής τιμής
	Αναλογικός ενισχυτής

Αναπαράσταση στο LOGO!	Ειδική Λειτουργία
	Αναλογικός πολυπλέκτης
	Αναλογική ράμπα
	PI ελεγκτής
	Αυτοσυγκράτηση
	Χρονικό παλμού
	Μηνύματα
	Μπουτόν (Softkey)
	Καταχωρητής ολίσθησης

ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΜΕΝΟΥ ΣΤΟ Logo Soft Comfort

Το Logo Soft Comfort περιέχει όπως όλα τα προγράμματα **menu bar** (μπάρα βασικών λειτουργιών προγράμματος) και **toolbars** (μπάρα εργαλείων).

menu bar:

Το menu bar περιέχει διάφορα menu όπως το **File menu**, **Edit menu**, **Format menu**, **View menu**, **Tools menu**, **Window menu** και το **Help menu**

Τα menu που εστιάζουμε πιο πολύ την προσοχή μας είναι τα File menu, Edit menu, View menu, Tools menu.

File menu:

New | Open... | Close | Close All | Save | Save As... | Page Setup... | Print Preview | Print...
Properties... | compare | convert to LAD | Exit

Στο File menu περιέχονται όλες οι εντολές δημιουργίας, ανοίγματος/κλεισίματος, αποθήκευσης, εκτύπωσης και εξόδου αρχείου.

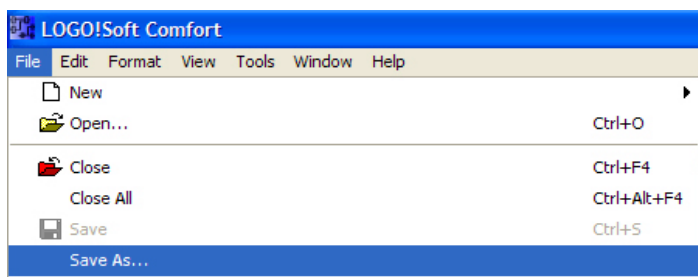
Επιλέγοντας πχ **File menu – New – Function Block Diagram**

Δημιουργούμε ένα FDB αρχείο πρόγραμμα.



Επιλέγοντας τώρα **File menu – Save as**

Αποθηκεύουμε το αρχείο που μόλις δημιουργήσαμε.



Επιλέγοντας πχ **File menu – Close** μπορούμε να κλείσουμε το αποθηκευμένο πρόγραμμα.

Επιλέγοντας πχ **File menu – Print...** μπορούμε να εκτυπώσουμε το πρόγραμμα.

Επιλέγοντας πχ **File menu – convert to LAD** μπορούμε να μετατρέψουμε το πρόγραμμα

από FBD σε LAD.

Επιλέγοντας πχ **File menu – Exit** επιλέγουμε την έξοδο μας από το αποθηκευμένο πρόγραμμα.

toolbars:

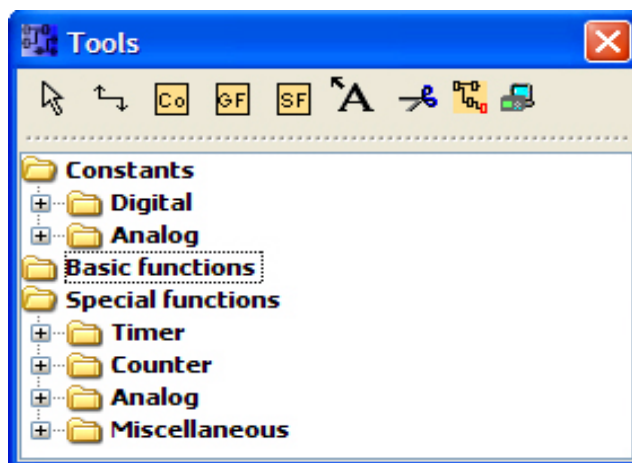
Το toolbars περιέχει 3 toolbar: Το **Standard toolbar**, το **Programming toolbox** και το **Simulation toolbar**.

Standard toolbar

Τα εικονίδια του Standard toolbar περιέχουν τις εντολές, οι οποίες είναι επίσης διαθέσιμες μέσω menu bar και παρέχουν τη γρήγορη πρόσβαση στις σχετικές λειτουργίες.



Programming toolbox (Tools)



Τα εικονίδια του Programming toolbox περιέχουν τις εντολές για τον προγραμματισμό σε FBD, δημιουργία σχόλιων, καθώς και προσομοίωση προγράμματος.

Το Logo Soft Comfort είναι πρόγραμμα τύπου drug and drop δηλαδή όταν ο χρήστης αρχίσει να προγραμματίζει ουσιαστικά 'μεταφέρει' τα blocks – εντολών που θα χρησιμοποιήσει στον χώρο εργασίας του.

Ένα παράδειγμα χρήσης μιας πύλης OR δύο εισόδων I1, I2 και μιας εξόδου Q1 με προσομοίωση ακολουθεί:

1. Δημιουργούμε ένα FDB αρχείο πρόγραμμα.
File menu – New – Function Block Diagram

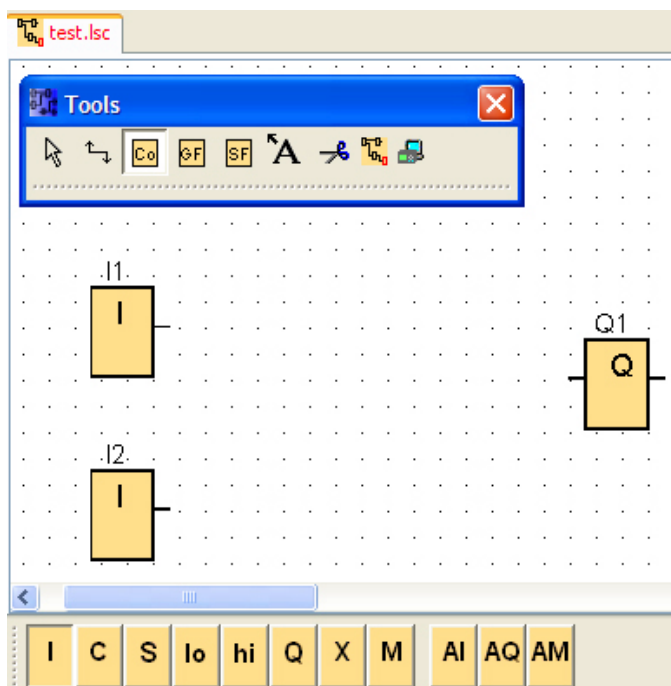


2. Αποθηκεύουμε το αρχείο που μόλις δημιουργήσαμε με το όνομα 'test'.

File menu – Save as

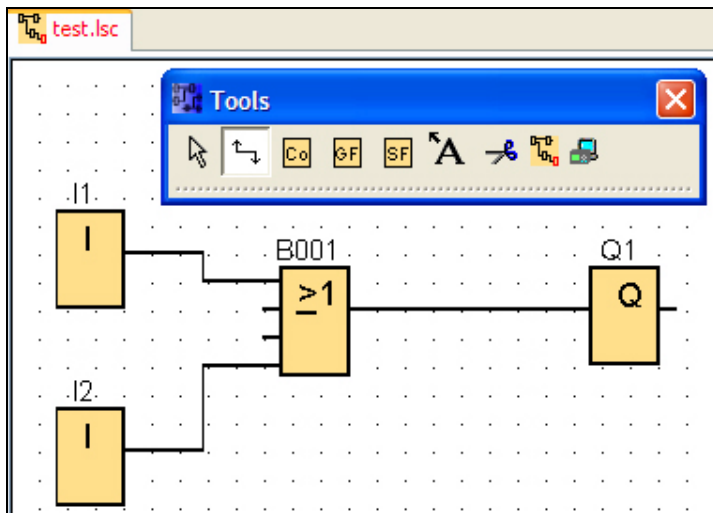
3. Σχεδιάζουμε το πρόγραμμα..

Από το **Tools – Co** (Constants) επιλέγουμε-‘μεταφέρουμε’ (drug and drop) τις εισόδους I1, I2 και την έξοδο Q1.



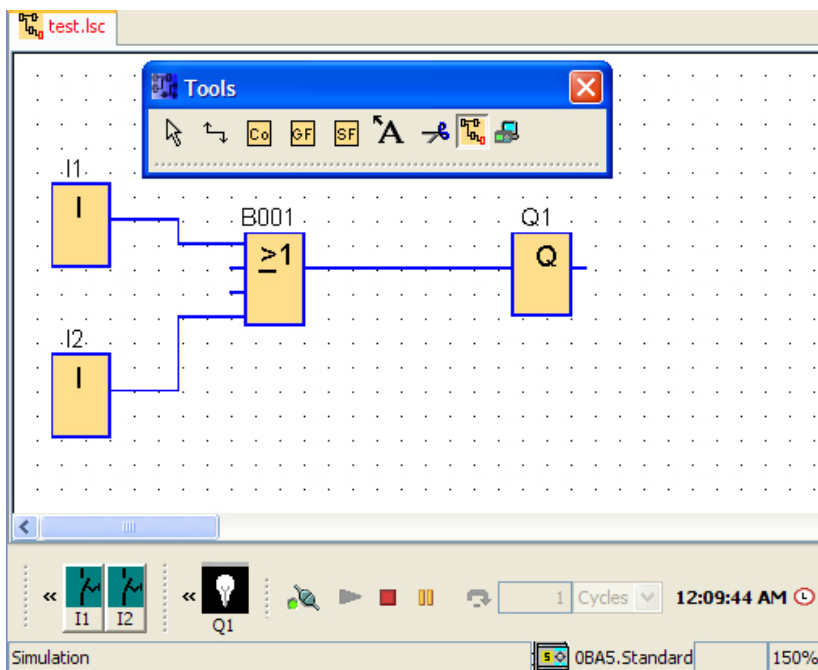
Από το **Tools – GF** (General Functions) επιλέγουμε-‘μεταφέρουμε’ το block της πύλης OR.

Αφού σχεδιάσουμε και το τελευταίο block επιλέγουμε από το **Tools - connect** και ενώνουμε τα blocks μεταξύ τους.



4. Προσομοίωση Προγράμματος

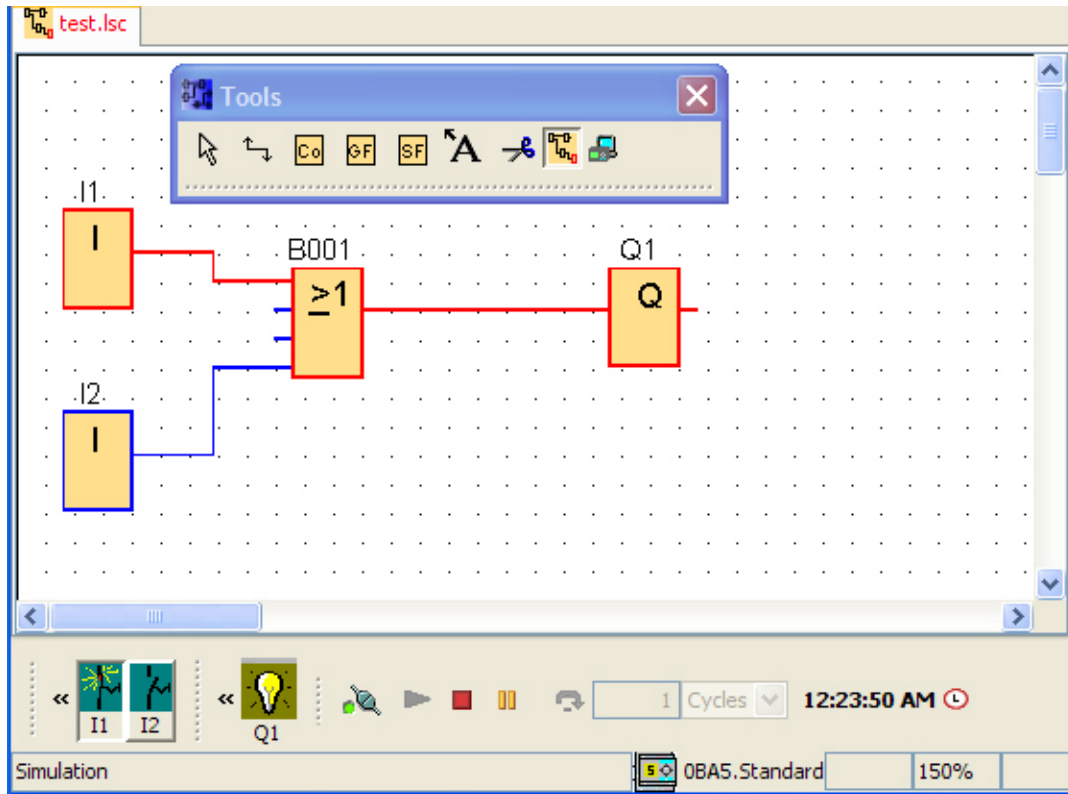
Αφού καταφέραμε να σχεδιάσουμε το πρόγραμμα μας δεν μένει τίποτε άλλο παρά να δούμε αν δουλεύει όπως θέλουμε. Αυτό επιτυγχάνεται με το Simulation (Προσομοίωση).



Επιλέγοντας Προσομοίωση, εμφανίστηκε το Simulation toolbox το οποίο μας δίνει την δυνατότητα να δούμε σε λειτουργία το πρόγραμμα.

Μπορούμε να ενεργοποιήσουμε /απενεργοποιήσουμε τις εισόδους του, παρατηρώντας ταυτόχρονα τις εξόδους του.

Παρατηρούμε ότι I1, I2, Q1 είναι απενεργοποιημένες (OFF).



I1	ενεργοποιημένη	ON
I2	απενεργοποιημένη	OFF
Q1	ενεργοποιημένη	ON

3. Εφαρμογές με το Logo Soft Comfort

A. ΠΥΛΕΣ / GATES

1. Σε μια διαδικασία ελέγχου θέλουμε όταν ο διακόπτης I1 κλείσει πρέπει να ενεργοποιούνται οι κινητήρες K1 και K2. Όταν οι K1 και K2 είναι ενεργοποιημένοι και κλείσει και ο διακόπτης I2 πρέπει να ενεργοποιηθεί η γεννήτρια K3.

ΛΥΣΗ:

ΕΙΣΟΔΟΙ: Διακόπτες : **I1, I2**

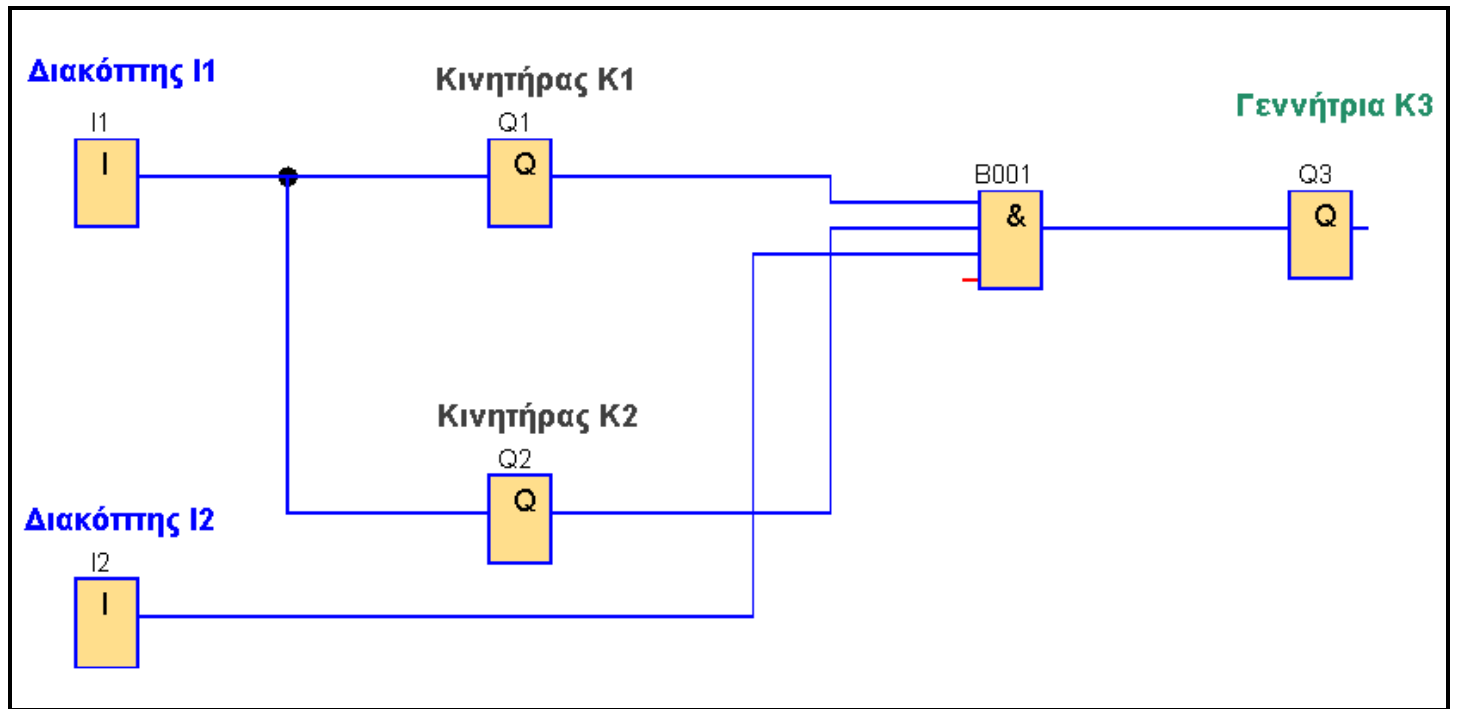
ΕΞΟΔΟΙ: Κινητήρες, Γεννήτρια // K1, K2, K3: **Q1, Q2, Q3**

ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ

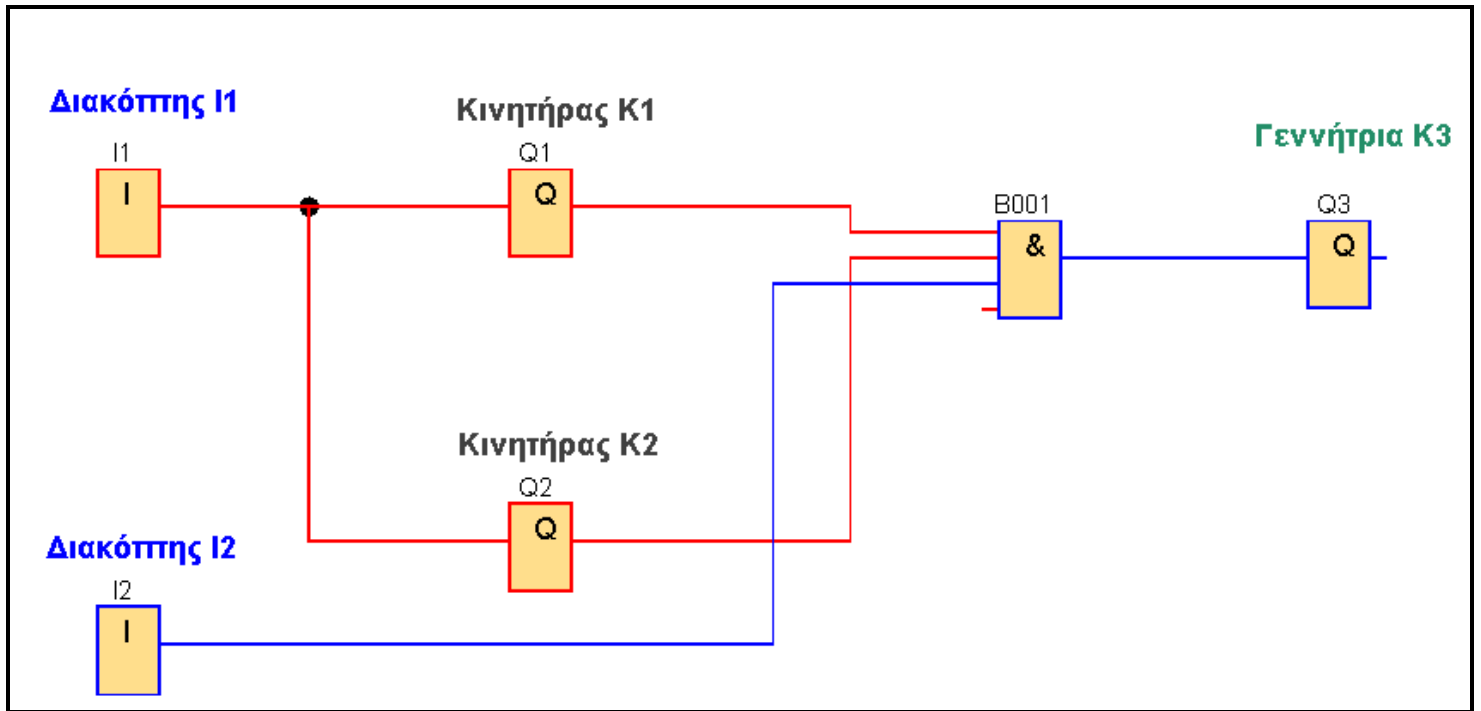
Διακόπτης (I1)	Διακόπτης (I2)	Κινητήρας (K1)	Κινητήρας (K2)	Γεννήτρια (K3)
<i>I1</i>	<i>I2</i>	<i>Q1</i>	<i>Q2</i>	<i>Q3</i>

0	0	0	0	0
1	0	1	1	0
0	1	0	0	0
1	1	1	1	1

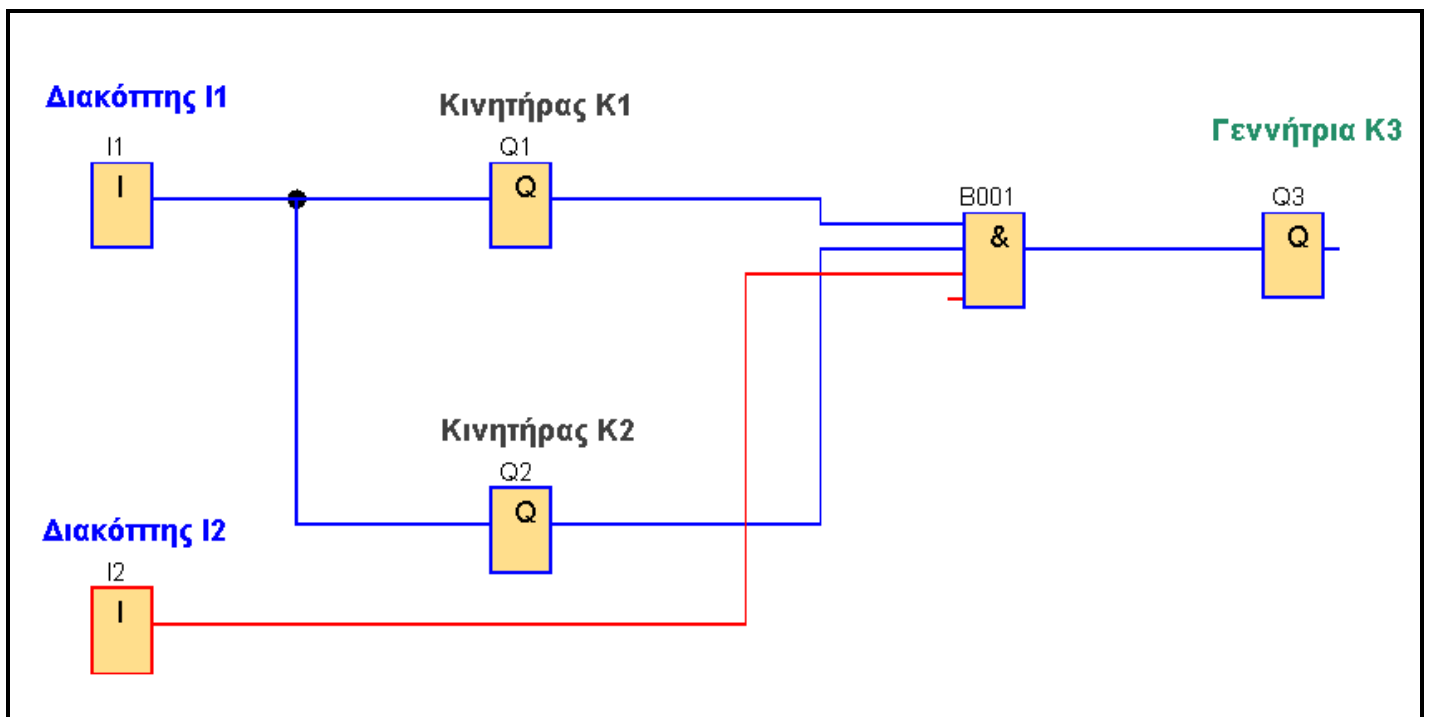
Είσοδοι I1 off και I2 off, *Εξοδοι* Q1 off και Q2 off και Q3 off



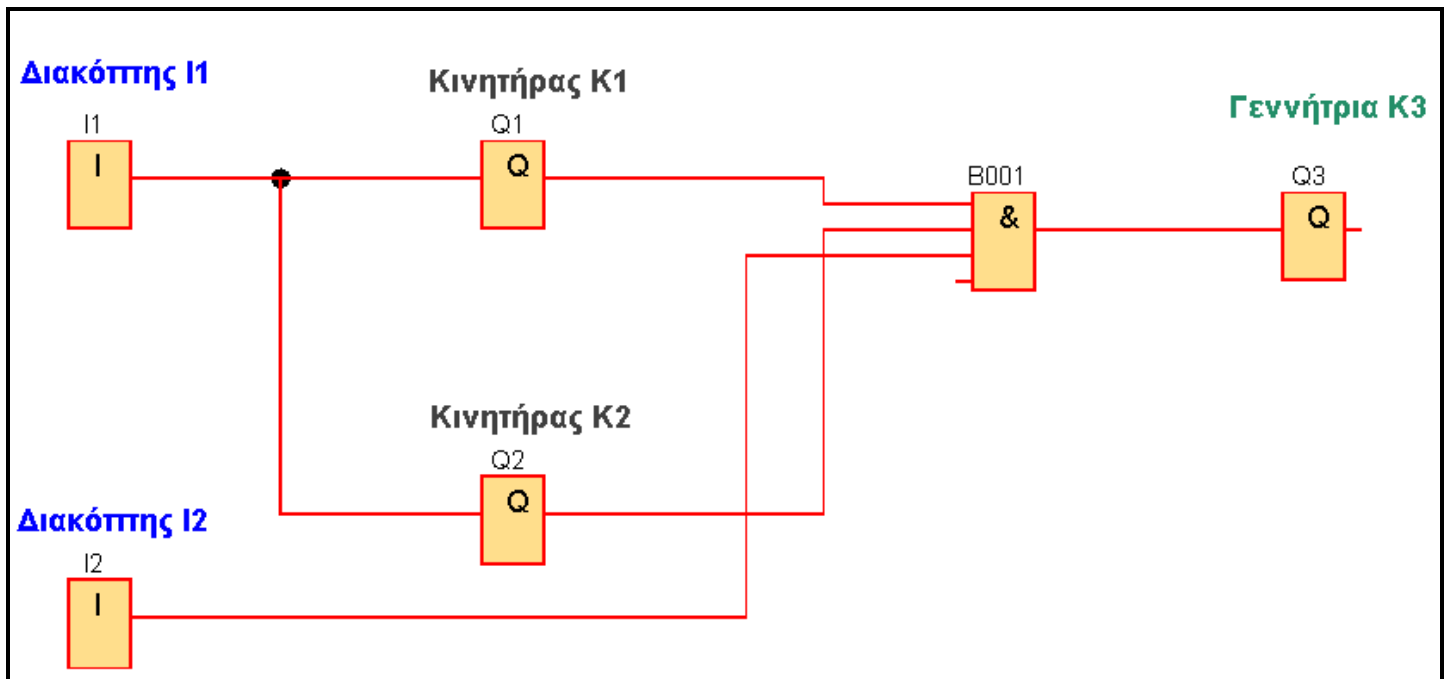
Είσοδοι I1 on και I2 off, *Εξοδοι* Q1 on και Q2 on και Q3 off



Είσοδοι I1 off και I2 on, *Έξοδοι* Q1 off και Q2 off και Q3 off



Είσοδοι I1 on και I2 on, *Έξοδοι* Q1 on και Q2 on και Q3 on



2. Σε ένα ΓΚΑΡΑΖ θέλουμε να φτιαχτεί ένα αυτόματο σύστημα πάνω σε **ΜΙΑ (κοινή) Θύρα (ΕΙΣΟΔΟΥ / ΕΞΟΔΟΥ)** όπου τα αυτοκίνητα θα εισέρχονται / εξέρχονται από το κτίριο με την βοήθεια ένδειξης Σηματοδότη.
Σε περίπτωση που ανιχνευτεί κοινή προσέλευση οχημάτων στην Θύρα από τους αισθητήρες να δίνεται προτεραιότητα στα οχήματα που εξέρχονται.

Σημείωση: Να υλοποιηθεί με χρήση OR και AND.

ΛΥΣΗ:

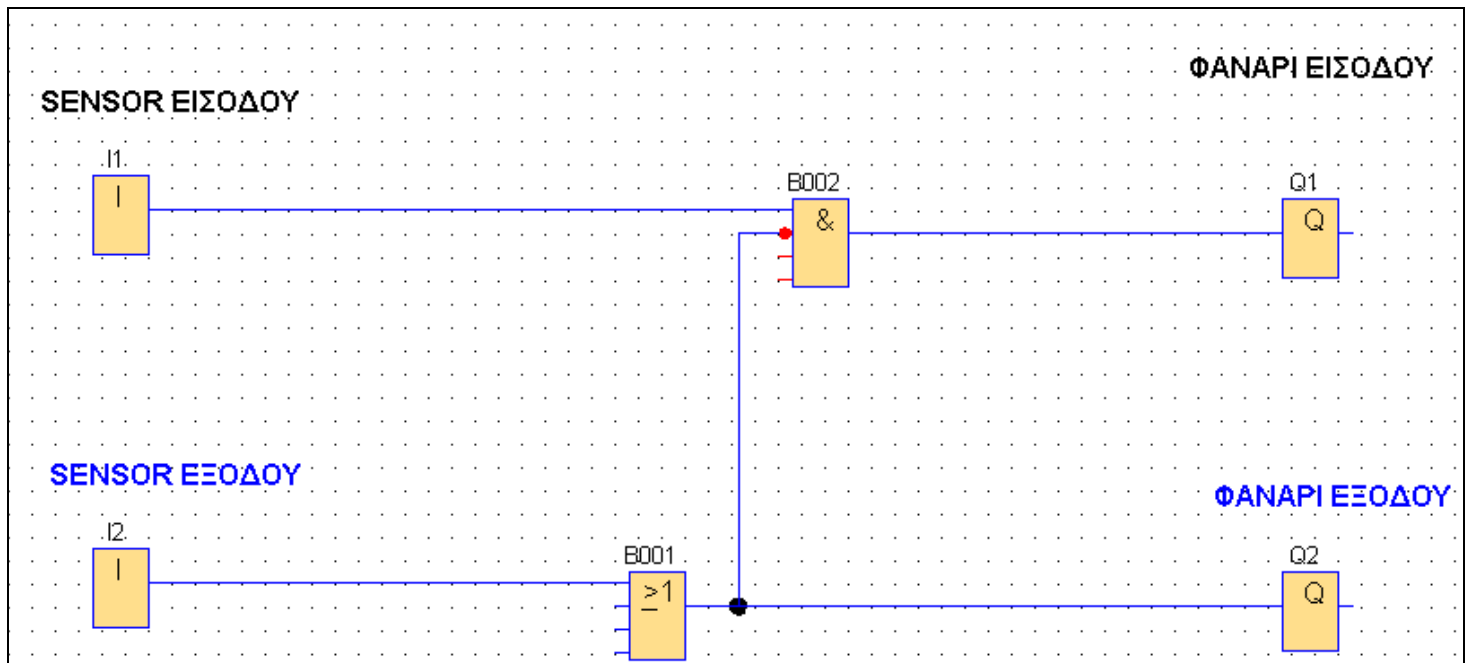
ΕΙΣΟΔΟΙ: ΑΙΣΘΗΤΗΡΙΑ ΕΙΣΟΔΟΥ/ΕΞΟΔΟΥ, **I1, I2**

ΕΞΟΔΟΙ: ΣΗΜΑΤΟΔΟΤΕΣ (ΦΑΝΑΡΙΑ), **Q1, Q2**

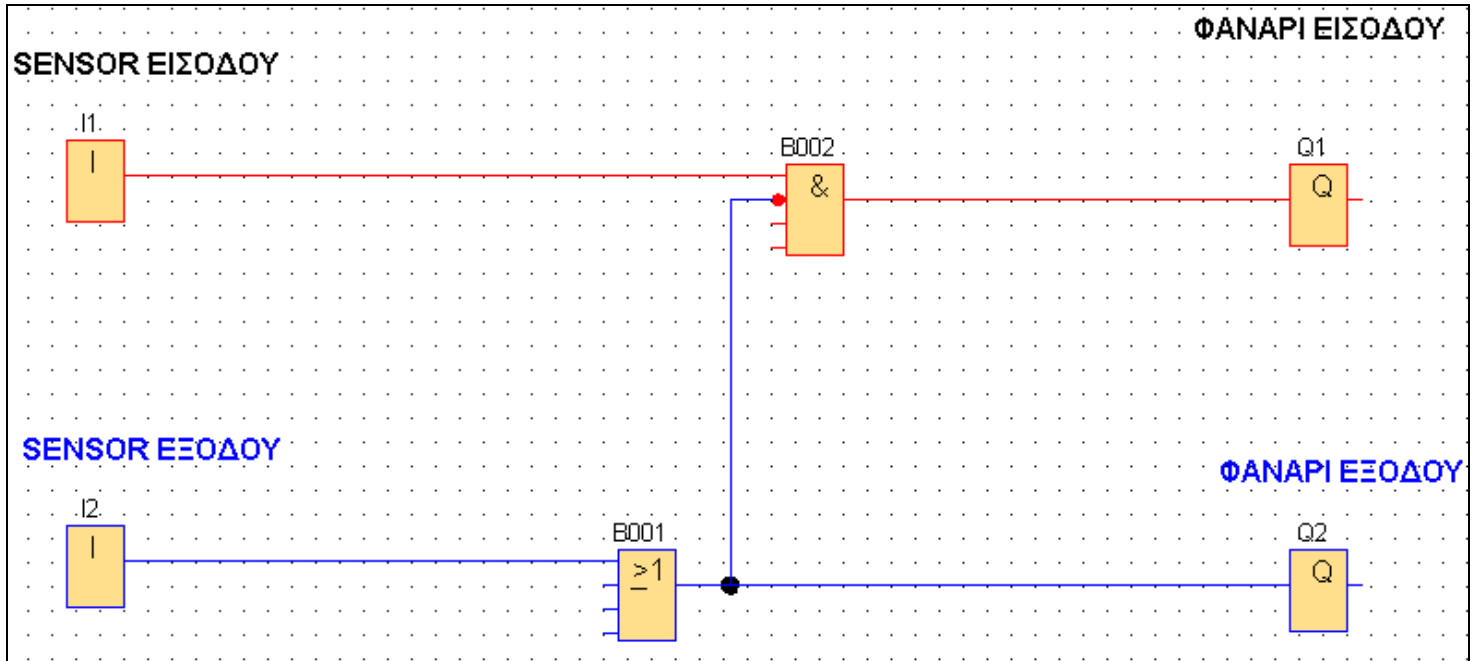
ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ

ΑΙΣΘΗΤΗΡΙΟ ΕΙΣΟΔΟΥ	ΑΙΣΘΗΤΗΡΙΟ ΕΞΟΔΟΥ	ΦΑΝΑΡΙ ΕΙΣΟΔΟΥ	ΦΑΝΑΡΙ ΕΙΣΟΔΟΥ
<i>I1</i>	<i>I2</i>	<i>Q1</i>	<i>Q2</i>
0	0	0	0
0	1	0	1
1	0	1	0
1	1	0	1

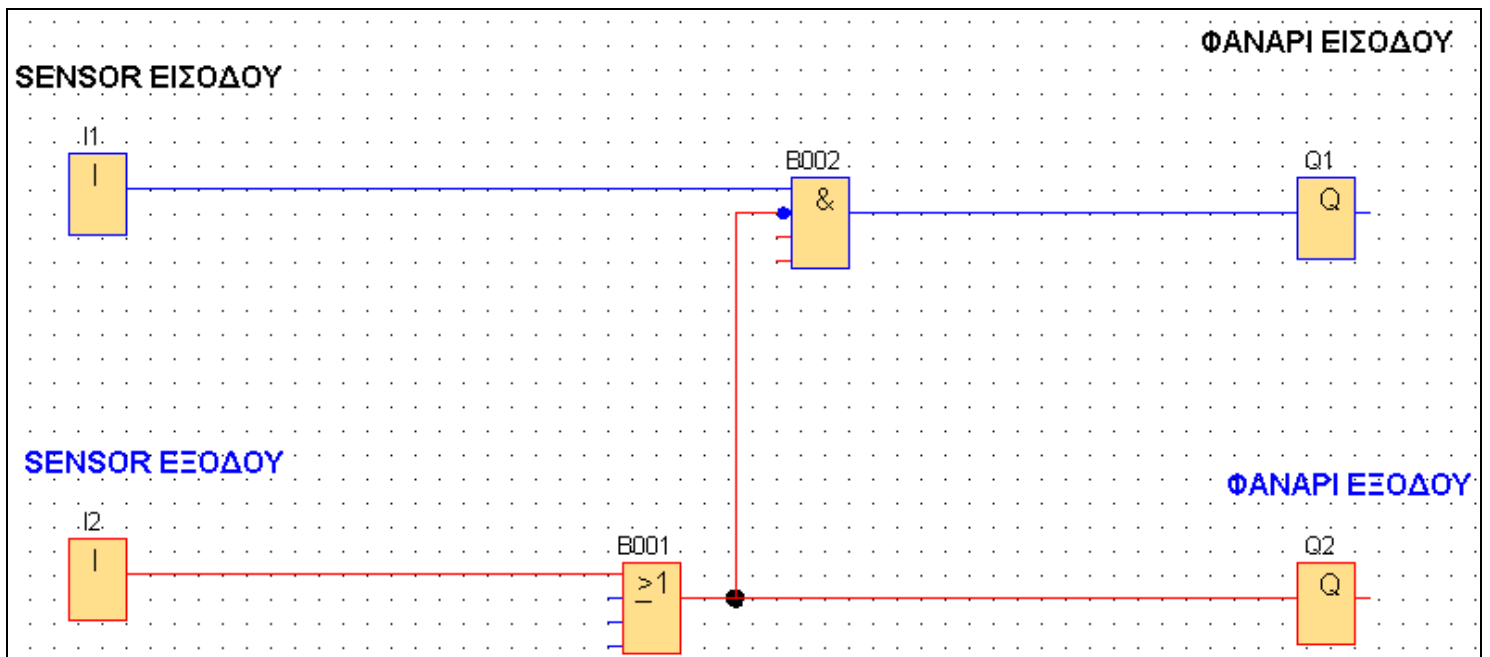
Είσοδοι I1 off και I2 off, Έξοδοι Q1 και Q2 off



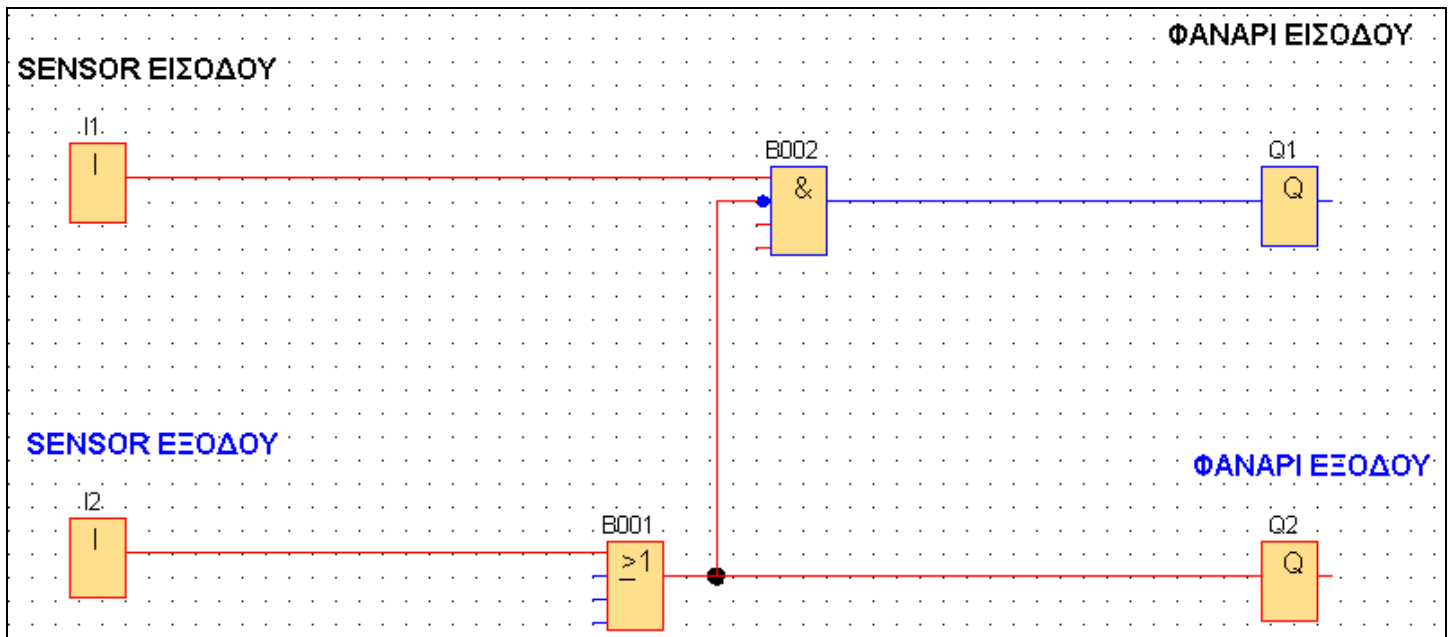
Είσοδοι I1 on και I2 off, Έξοδοι Q1 on και Q2 off



Είσοδοι I1 off και I2 on, Έξοδοι Q1 off και Q2 on



Είσοδοι I1 on και I2 on, Έξοδοι Q1 off και Q2 on (προτεραιότητα στα οχήματα που εξέρχονται)



B. ΧΡΟΝΙΚΑ / TIMERS

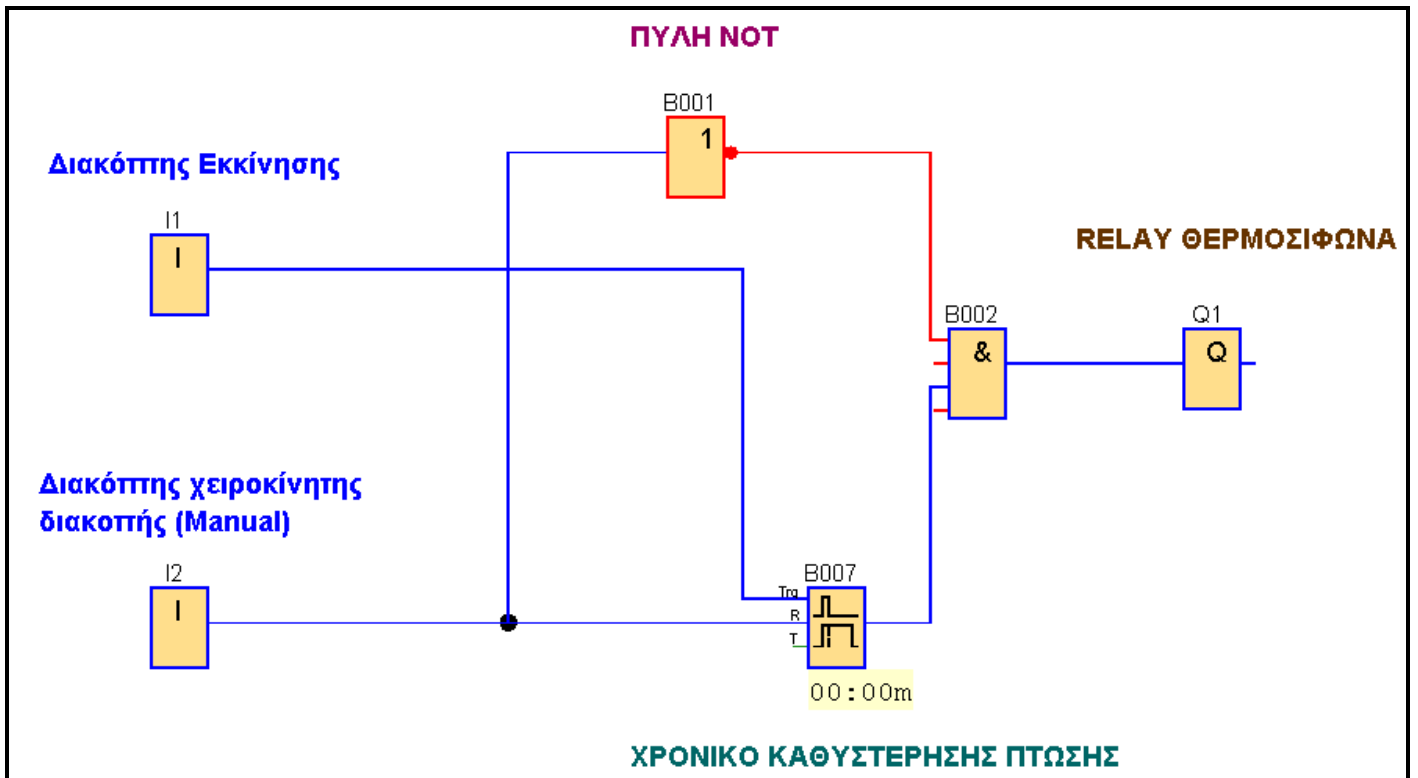
1. Σε έναν Θερμοσίφωνα θέλουμε να εισάγουμε ένα σύστημα κατά το οποίο πατώντας το μπουτόν I1 να λειτουργεί για ένα συγκεκριμένο χρονικό διάστημα (πχ 30 λεπτά) και στη συνέχεια να απενεργοποιείται. Αν ο χρήστης επιθυμεί την διακοπή της λειτουργίας σε οποιοδήποτε χρονικό σημείο ενδιάμεσα να δίνεται η δυνατότητα τερματισμού με ένα μπουτόν διακοπής.

ΛΥΣΗ:

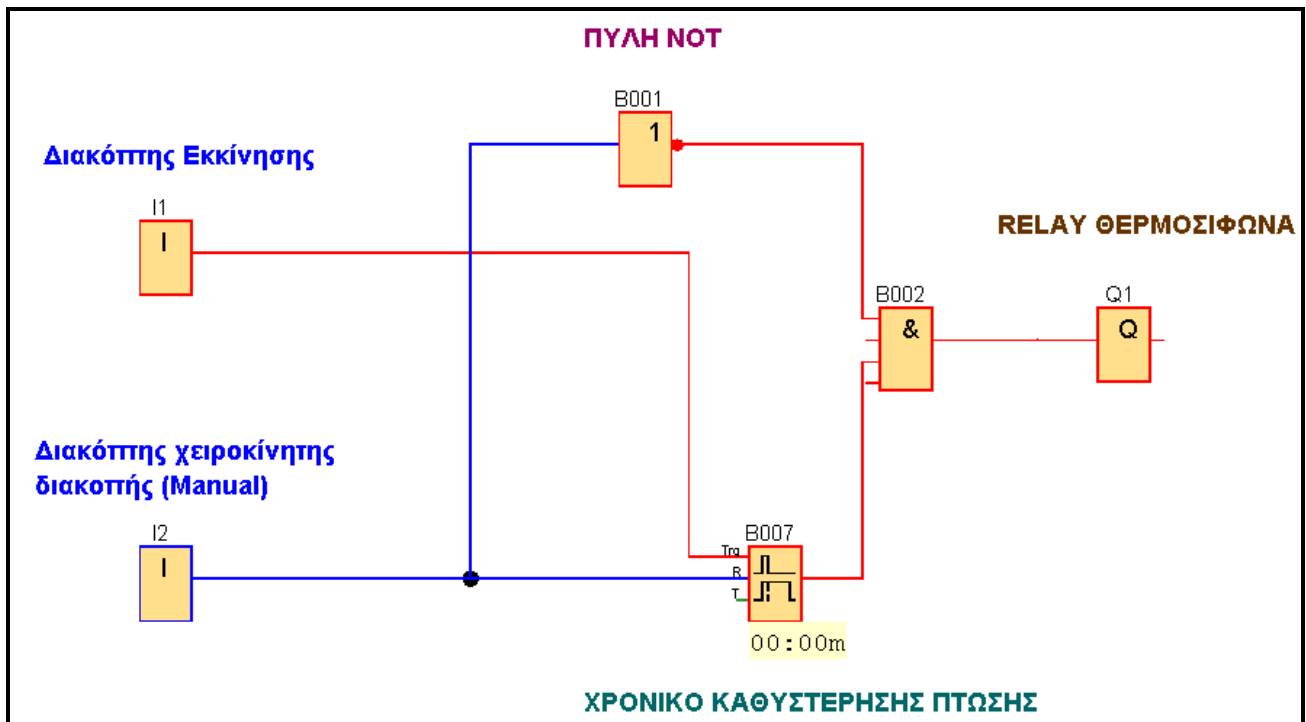
ΕΙΣΟΔΟΙ: Διακόπτες Εκκίνησης/Διακοπής, **I1, I2**

ΕΞΟΔΟΙ: Relay Θερμοσίφωνα, **Q1**

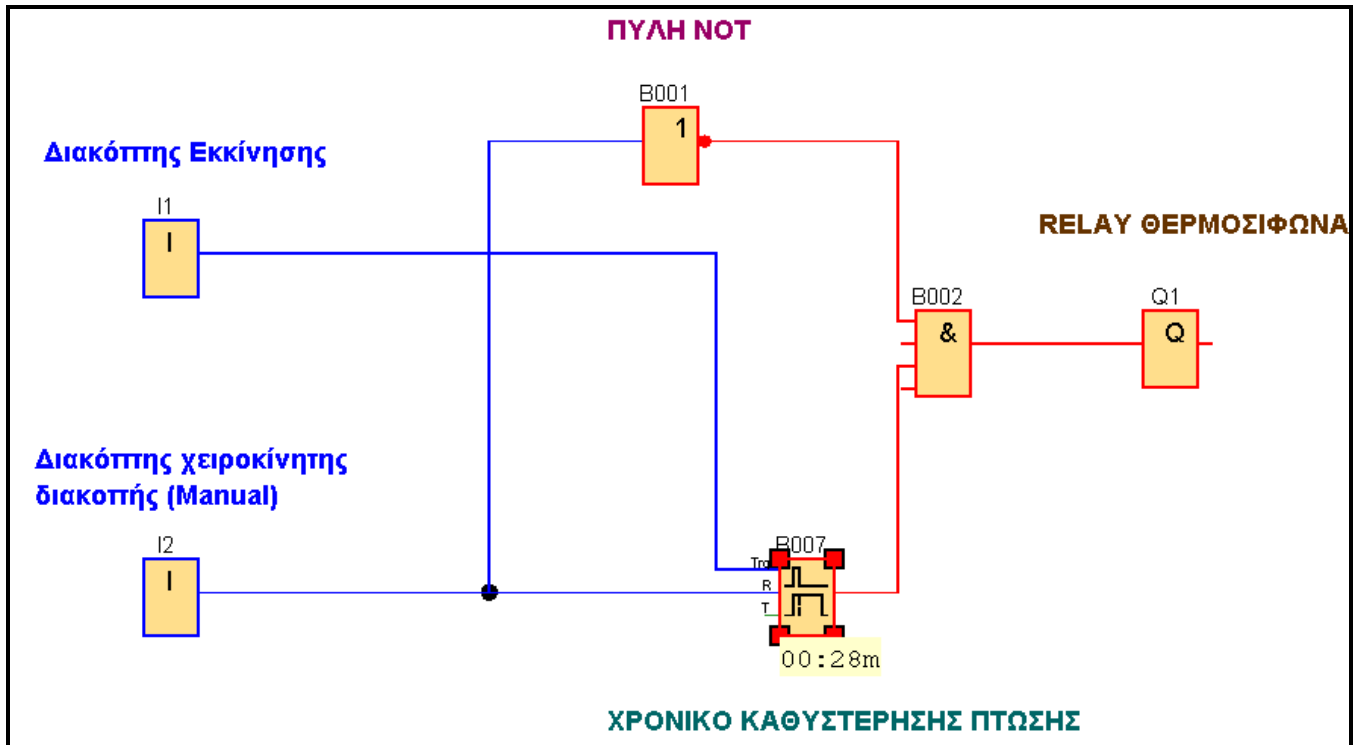
Είσοδοι I1 off και I2 off, **Timer:** 00:00m *Εξοδοι* Q1 off



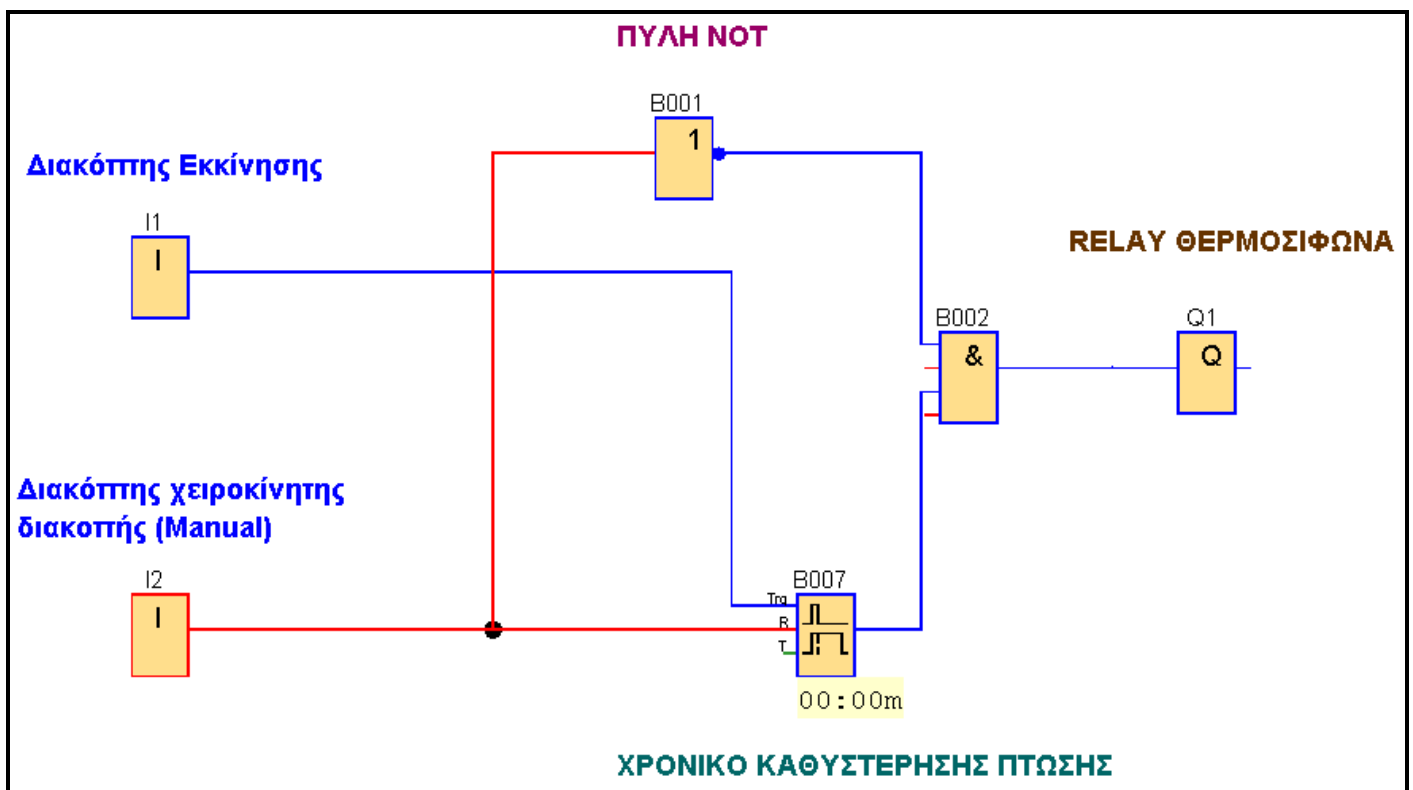
Είσοδοι: I1 on (ενεργοποίηση του Trigger του Χρονικού και ταυτόχρονα της εξόδου Q1) και I2 off, **Timer:** 00:00m **Εξοδοι:** Q1 on



Είσοδοι: I1 off και I2 off, **Timer:** 00:28m **Εξοδοι:** Q1 on



Είσοδοι I1 off και I2 on (απενεργοποίηση του Χρονικού (reset) και ταυτόχρονα της εξόδου Q1,
Timer: 00:00m, *Εξοδοι* Q1 off



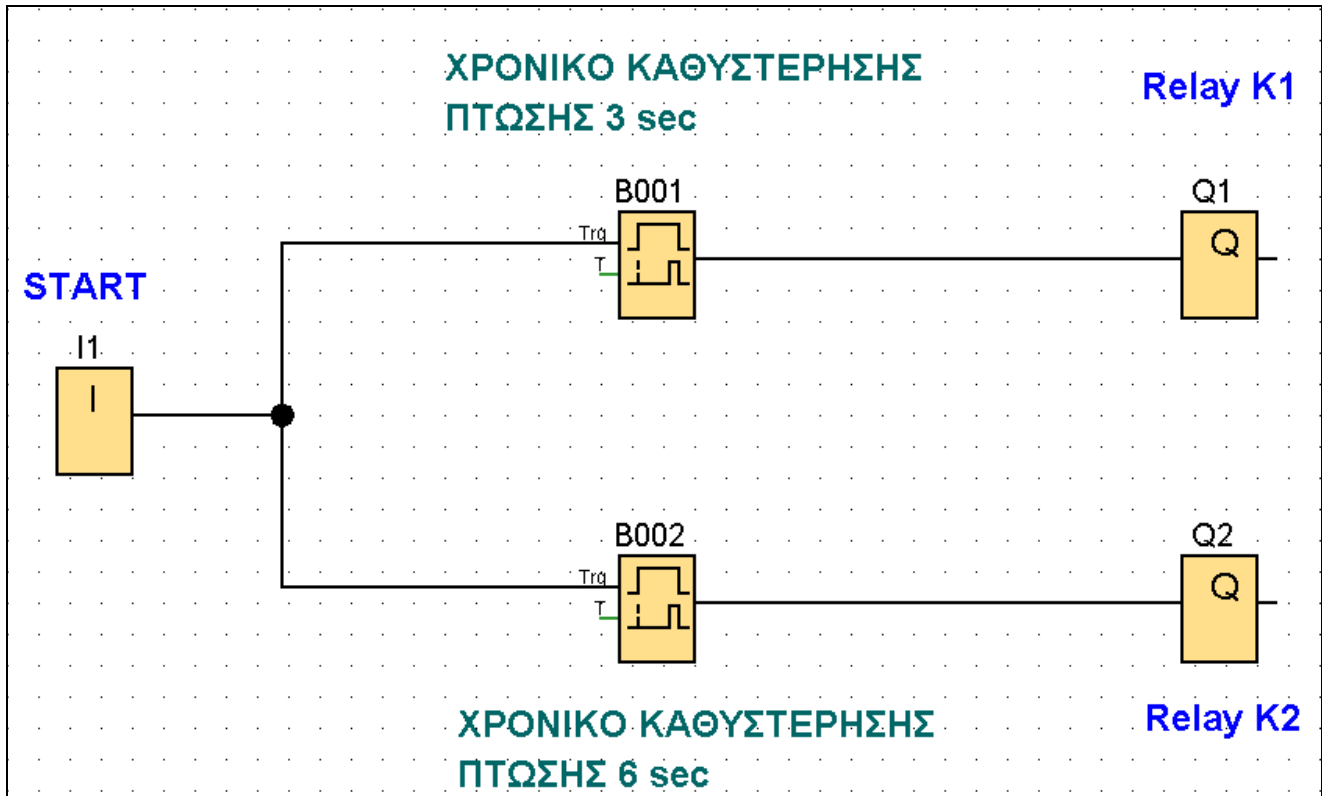
2. Σε μια διαδικασία έλεγχου η οποία χρονομετρείται για 10 sec ζητάμε μετά το πέρας 3 sec να γίνεται ενεργοποίηση του relay K1 και μετά από το πέρας 6 sec να γίνεται ενεργοποίηση και του relay K2. Τα δύο αυτά relay να παραμείνουν ενεργοποιημένα μέχρι την επανάληψη της διαδικασίας αυτής.

ΛΥΣΗ:

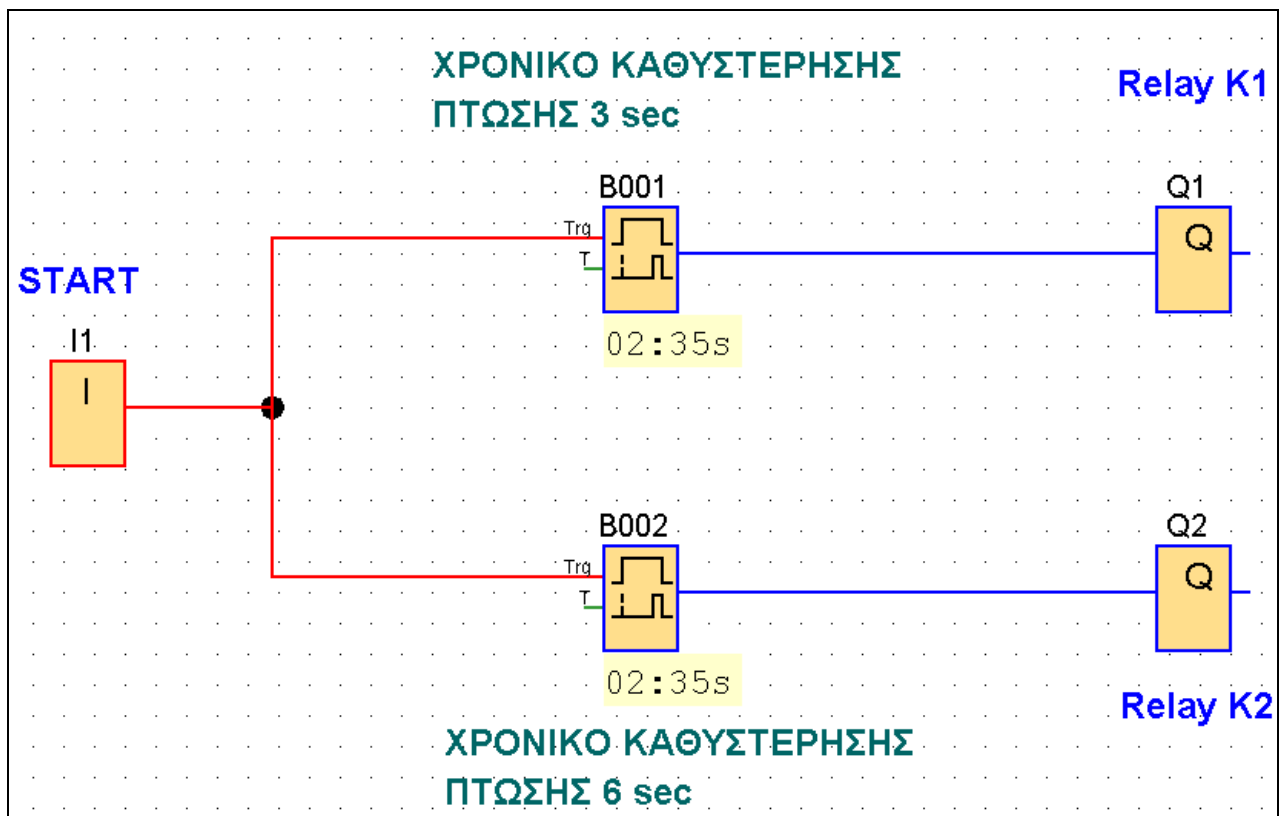
ΕΙΣΟΔΟΙ: Start, **I1**

ΕΞΟΔΟΙ: Relay, **Q1,Q2**

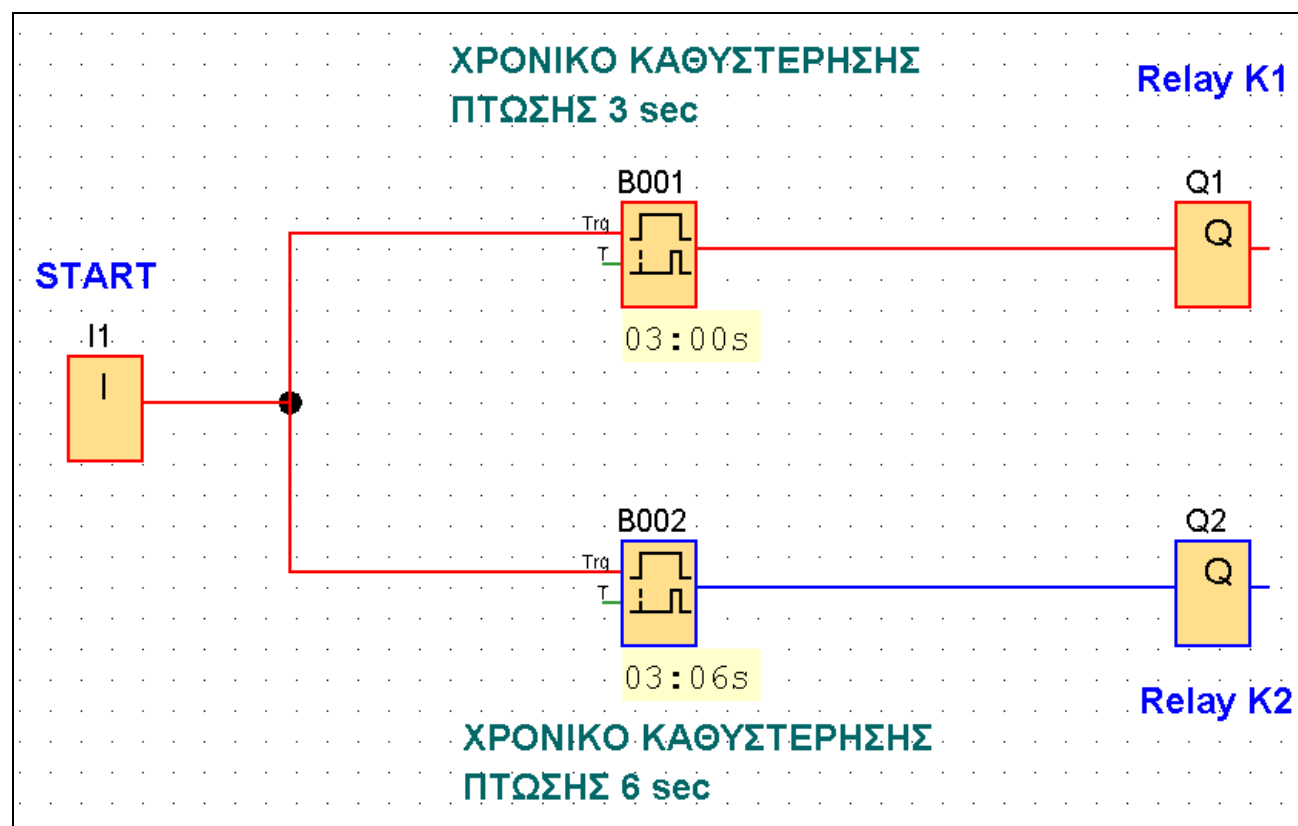
Είσοδοι: I1 off, **Timer 1:** 00:00 sec, **Timer 2:** 00:00 sec, *Έξοδοι:* Q1 off , Q2 off



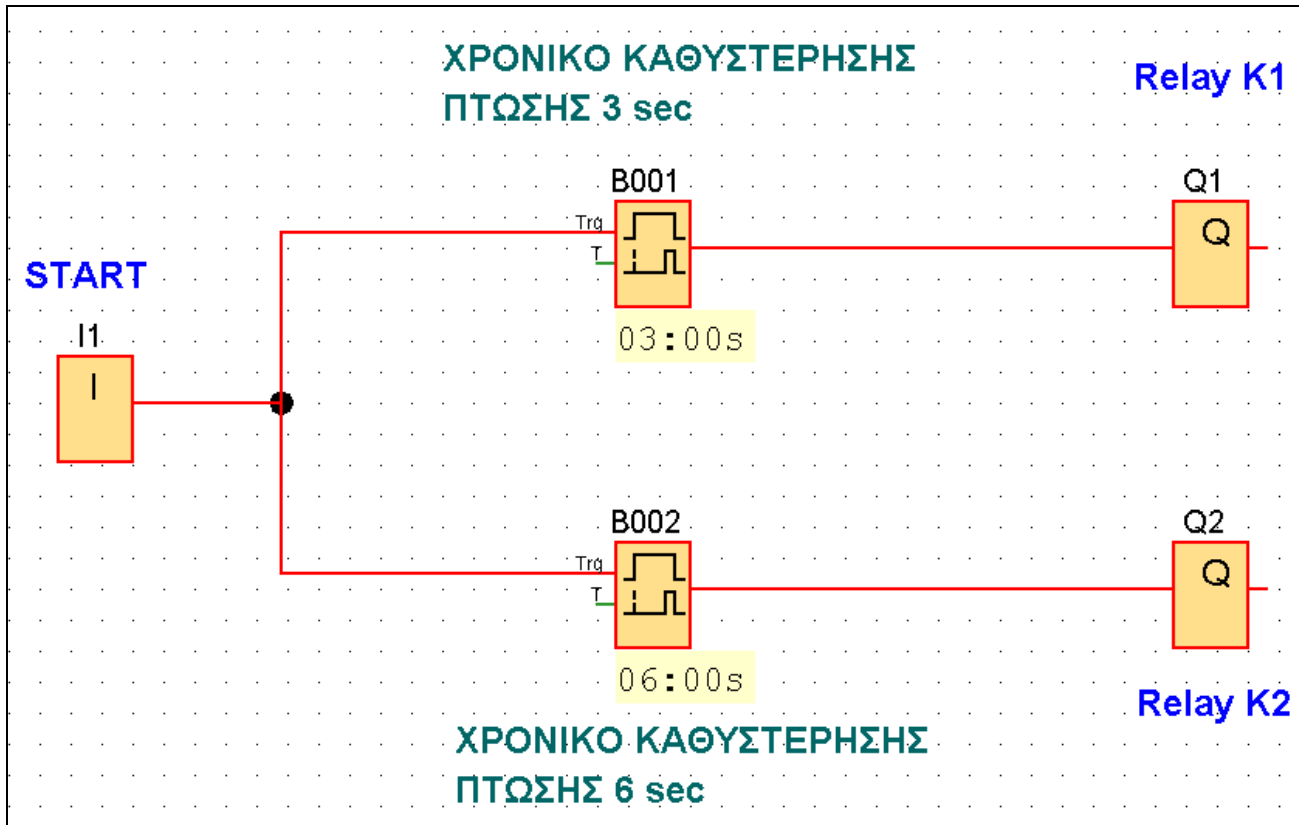
Είσοδοι: I1 on, **Timer 1:** 02:35 sec, **Timer 2:** 02:35 sec, *Έξοδοι:* Q1 off , Q2 off



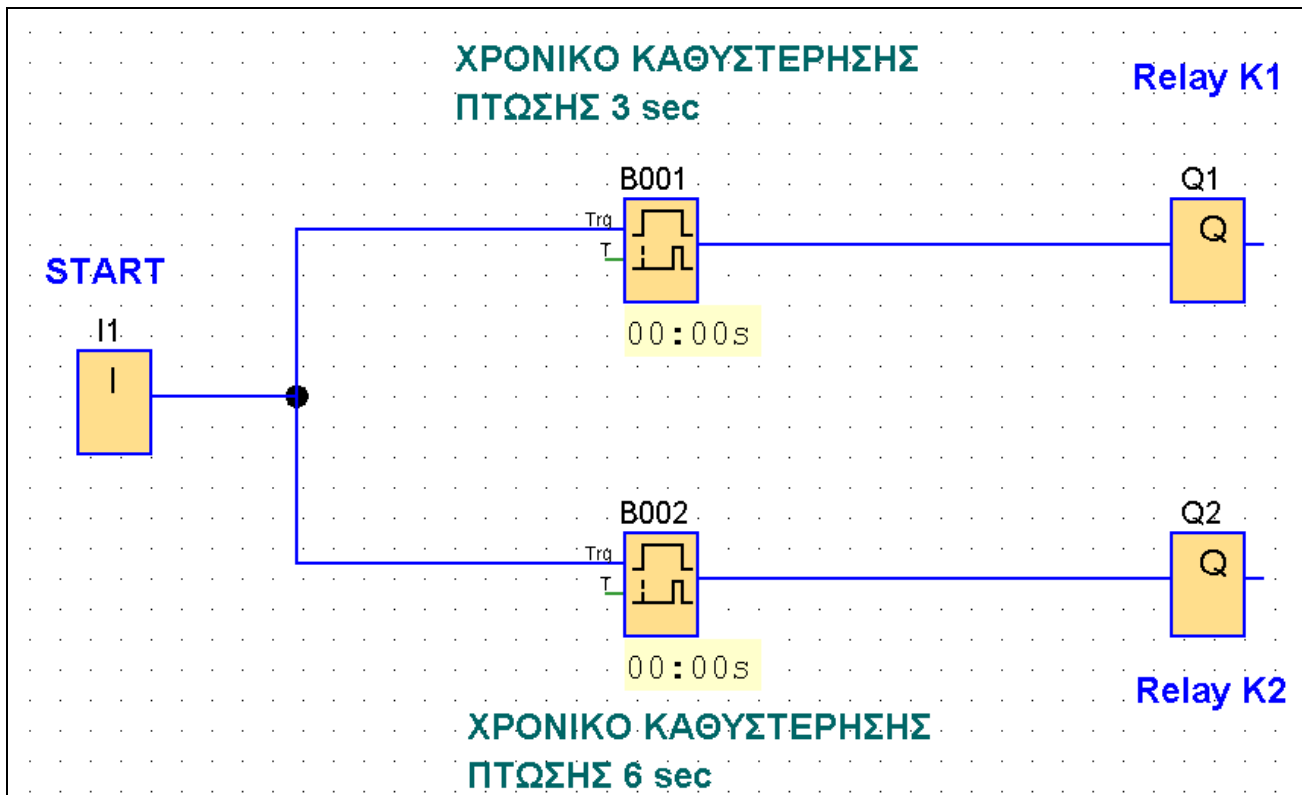
Είσοδοι: I1 on, **Timer 1:** 03:00 sec, **Timer 2:** 03:06 sec, *Έξοδοι:* Q1 on , Q2 off



Είσοδοι: I1 on, **Timer 1:** 03:00 sec, **Timer 2:** 06:00 sec, *Έξοδοι:* Q1 on , Q2 on



Είσοδοι: I1 off, **Timer 1:** 00:00 sec (reset), **Timer 2:** 00:00 sec (reset), *Έξοδοι:* Q1 off , Q2 off



Γ.

ΑΠΑΡΙΘΜΗΤΕΣ / COUNTERS

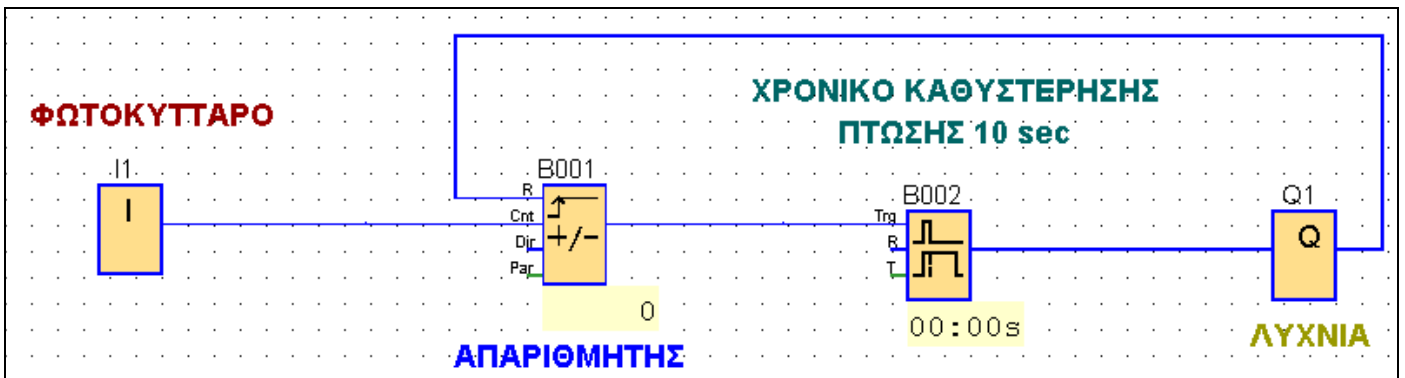
1. Σε μια διαδικασία έλεγχου θέλουμε να γίνονται τα εξής: Όταν περάσουν 5 μπουκάλια από το φωτοκύτταρο μιας γραμμής παραγωγής να ανάβει μια λυχνία η οποία να φωτοβολεί για 10 sec. Η διαδικασία να επαναλαμβάνεται συνέχεια.

ΛΥΣΗ:

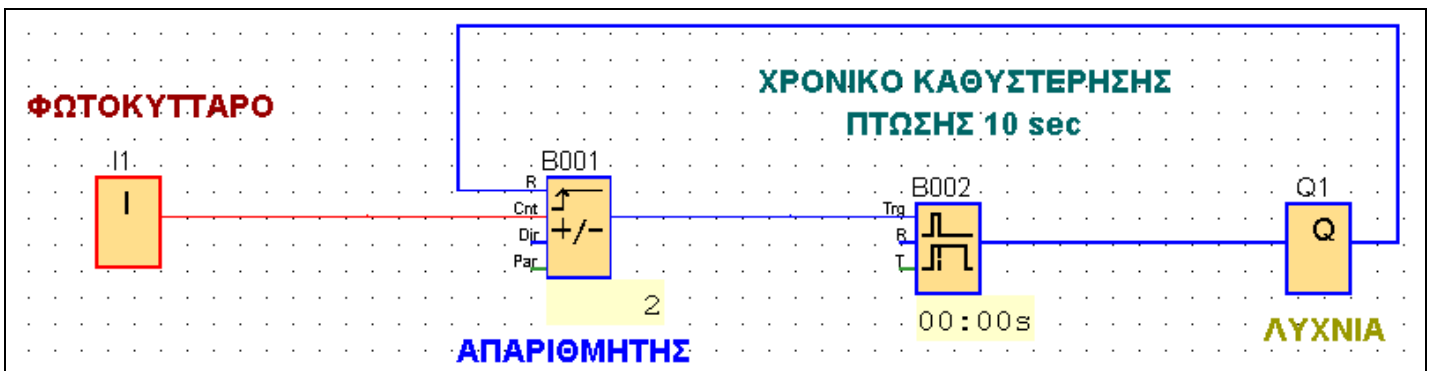
ΕΙΣΟΔΟΙ: Φωτοκύτταρο, I1

ΕΞΟΔΟΙ: Λυχνία, Q1

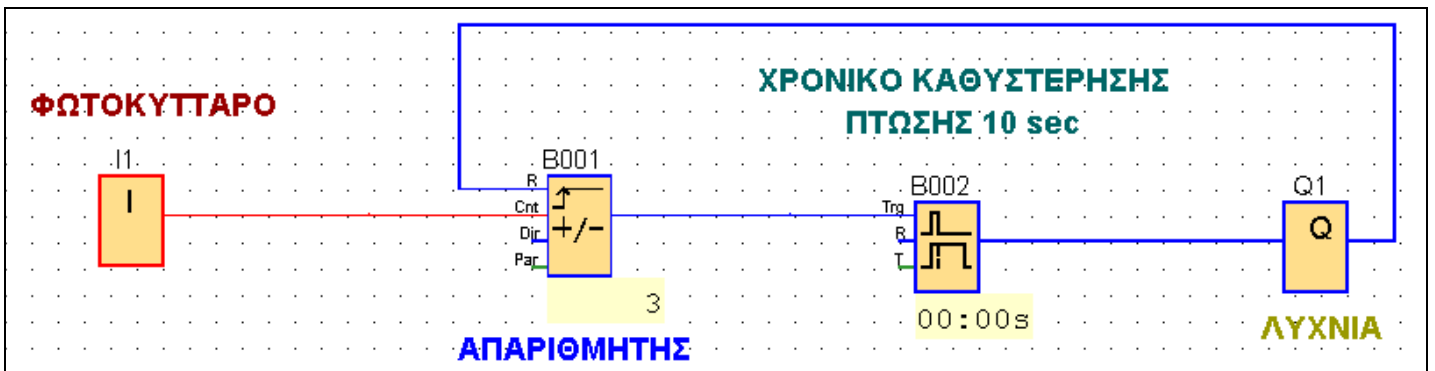
Είσοδοι: I1 off, **Counter:** 0, **Timer:** 00:00 sec, **Έξοδοι:** Q1 off



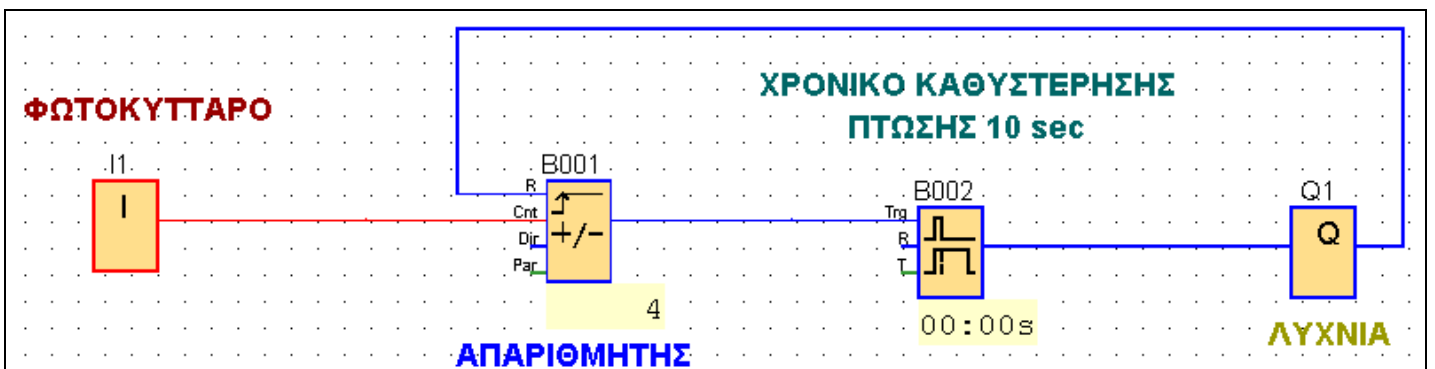
Είσοδοι: I1 on, **Counter:** 2, **Timer:** 00:00 sec, **Έξοδοι:** Q1 off



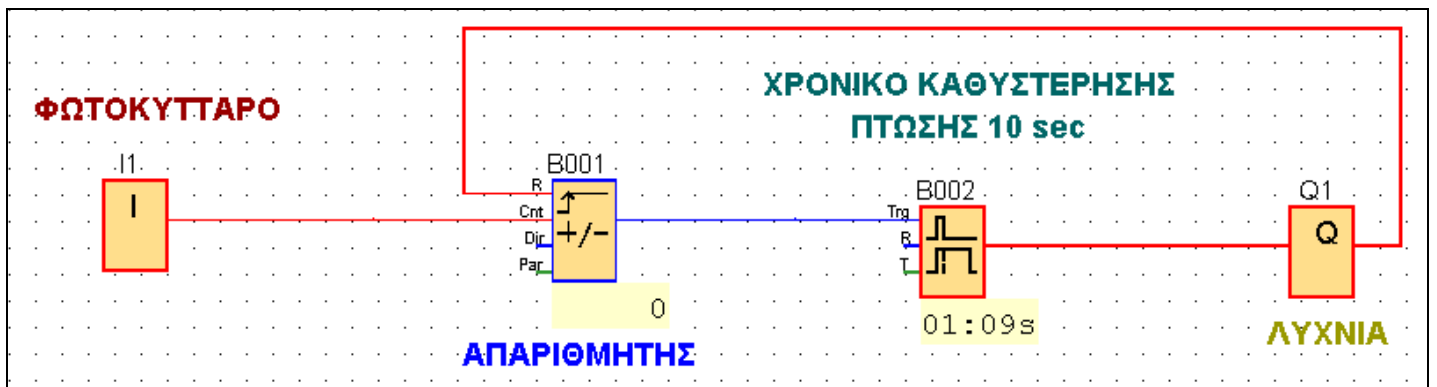
Είσοδοι: I1 on, Counter: 3, Timer: 00:00 sec, Έξοδοι: Q1 off



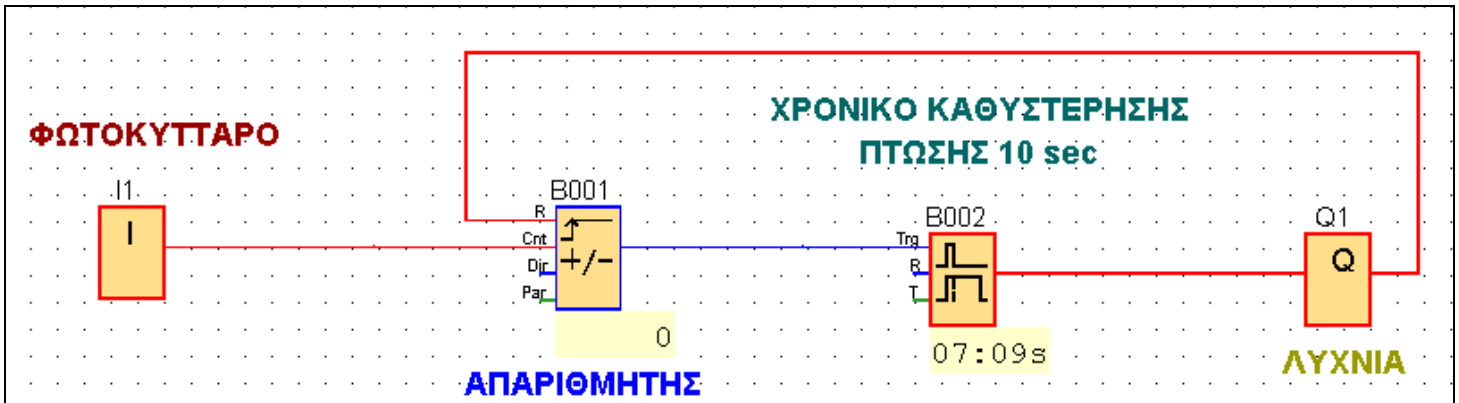
Είσοδοι: I1 on, Counter: 4, Timer: 00:00 sec, Έξοδοι: Q1 off



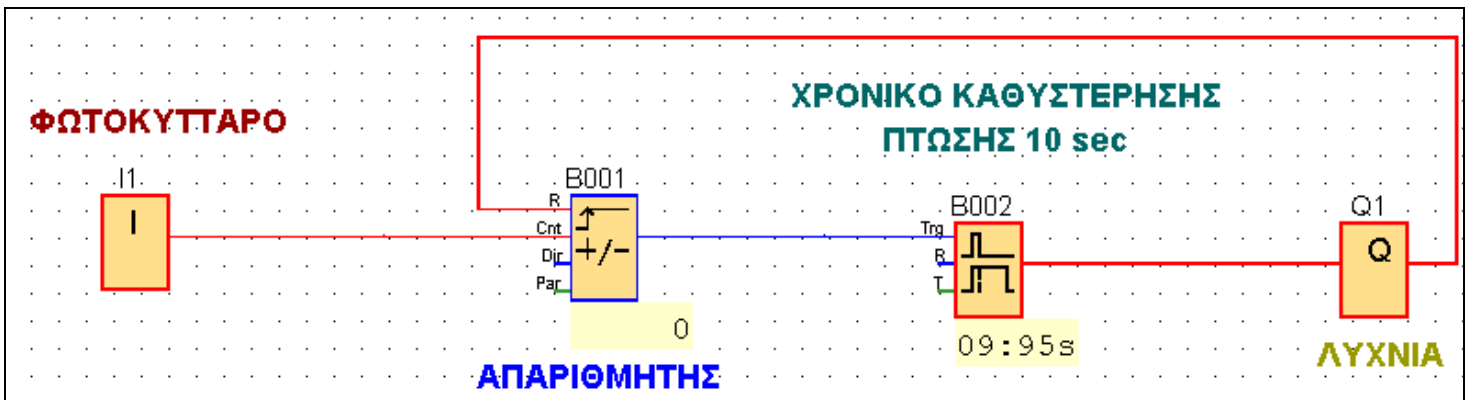
Είσοδοι: I1 on, Counter: 0 (5) [reset Counter], Timer: 01:09 sec, Έξοδοι: Q1 on



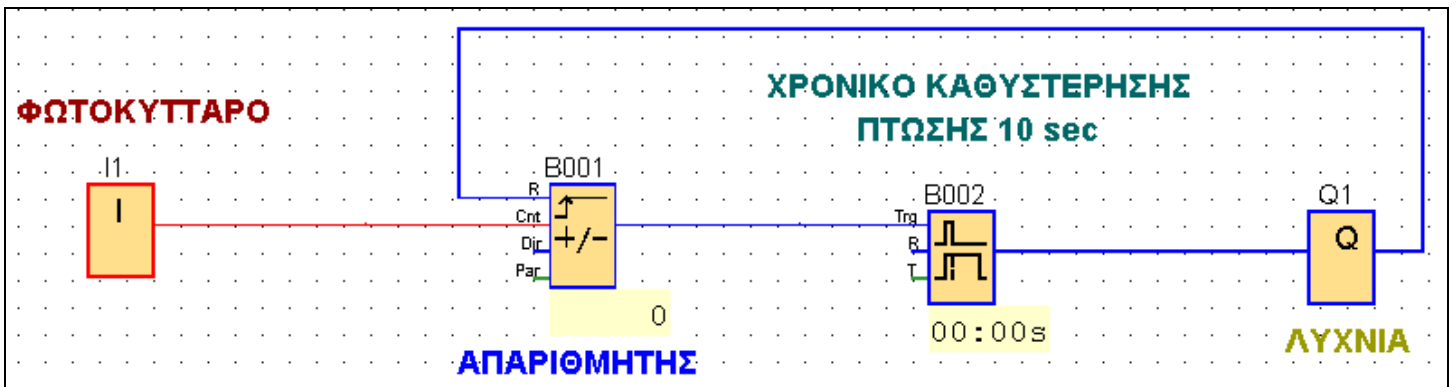
Είσοδοι: I1 on, Counter: 0, Timer: 07:09 sec, Έξοδοι: Q1 on



Είσοδοι: I1 on, Counter: 0, Timer: 09:95 sec, Έξοδοι: Q1 off



Είσοδοι: I1 on, Counter: 0, Timer: 00:00 (10:00) sec, Έξοδοι: Q1 off



2. Ζητείται να φτιαχτεί ένα πρόγραμμα που να ελέγχει τον κινητήρα ενός πλυντηρίου ρούχων.

Συμφώνα με αυτό το πρόγραμμα ο κινητήρας θα λειτουργεί για προκαθορισμένο χρονικό διάστημα (πχ 30 sec) δεξιόστροφα και μετά αριστερόστροφα για το ίδιο χρονικό διάστημα.

Η διαδικασία αυτή να επαναλαμβάνεται έως ότου τελειώσει ο ολικός χρόνος πλύσης (πχ 90 min) και να διακόπτεται χειροκίνητα από τον χρήστη σε οποιοδήποτε χρονικό σημείο.

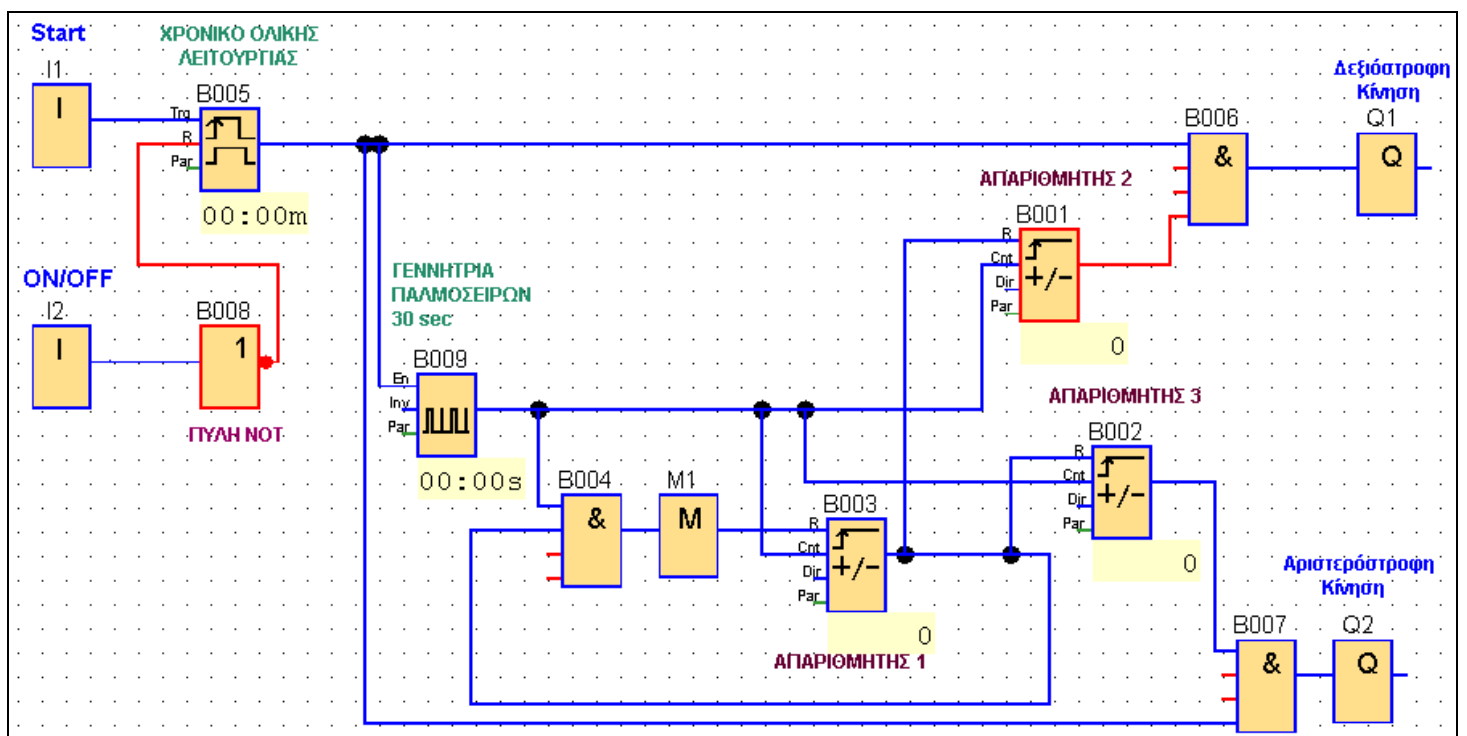
Σημείωση: Να υλοποιηθεί με χρήση του block Asynchronous Pulse Generator.

ΛΥΣΗ:

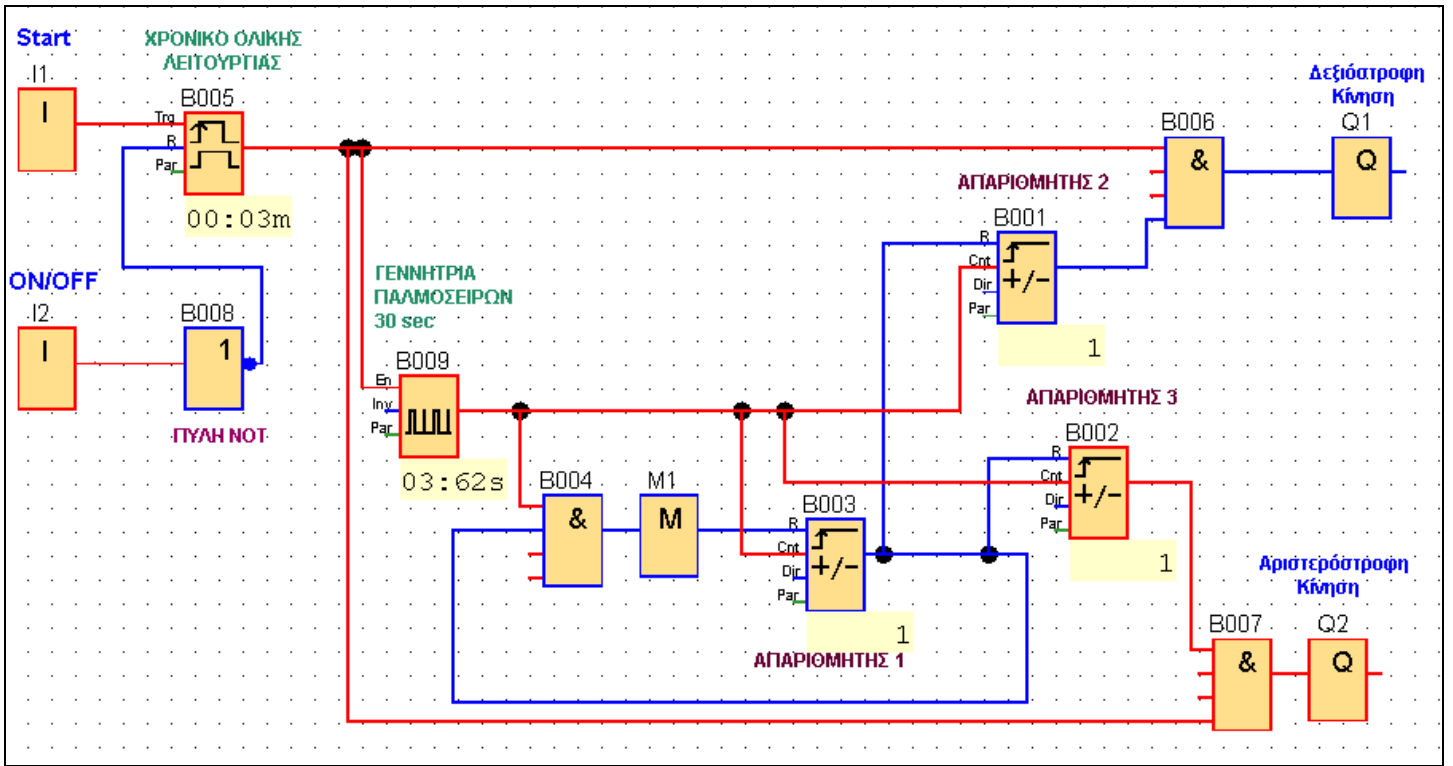
ΕΙΣΟΔΟΙ: Start, ON/OFF, I1, I2

ΕΞΟΔΟΙ: Δεξιόστροφη /Αριστερόστροφη Κίνηση, Q1,Q2

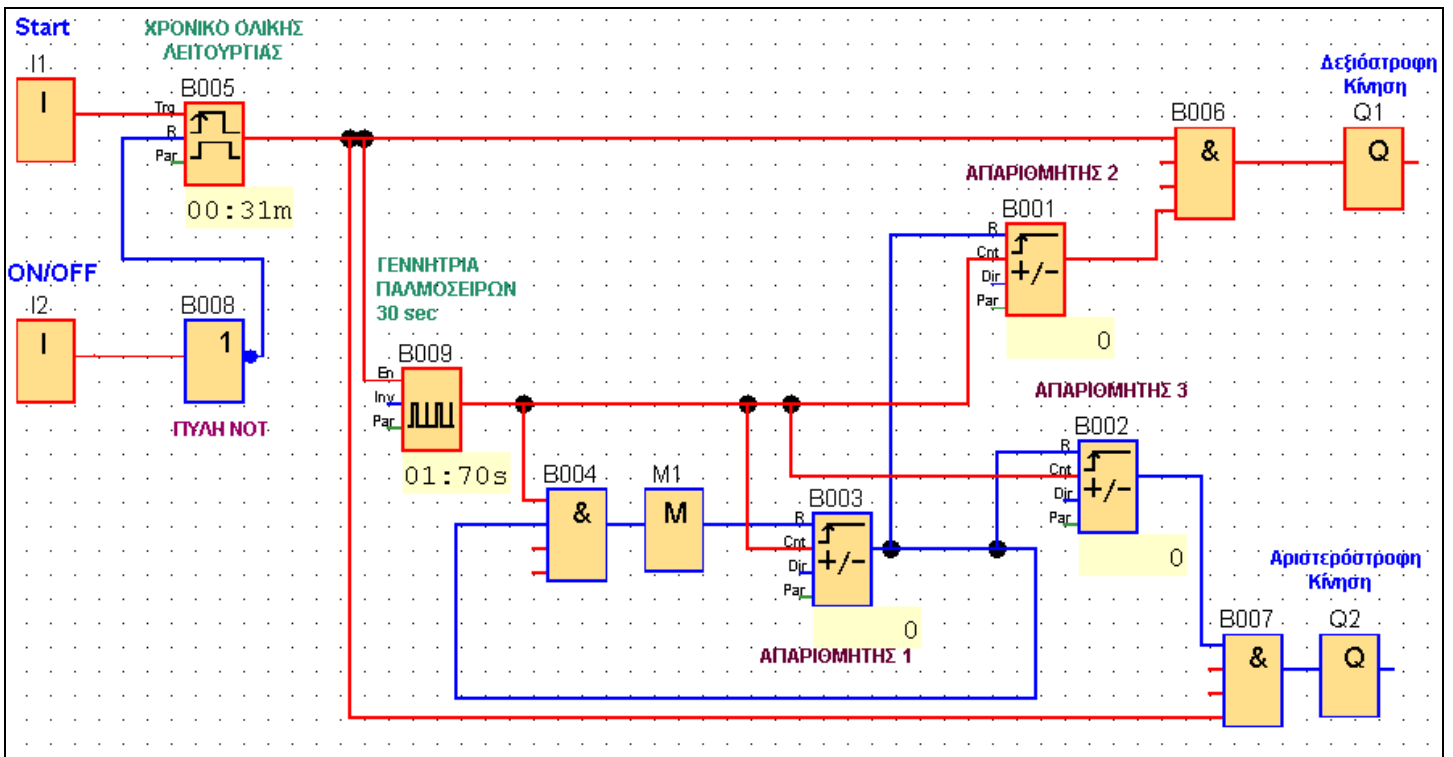
Είσοδοι: I1 off, I2 off, **Timer:** 00:00 min, **Pulse Generator:** 00:00 sec, **Εξοδοι:** Q1 off, Q2 off
Counters: C1, C2, C3: 0



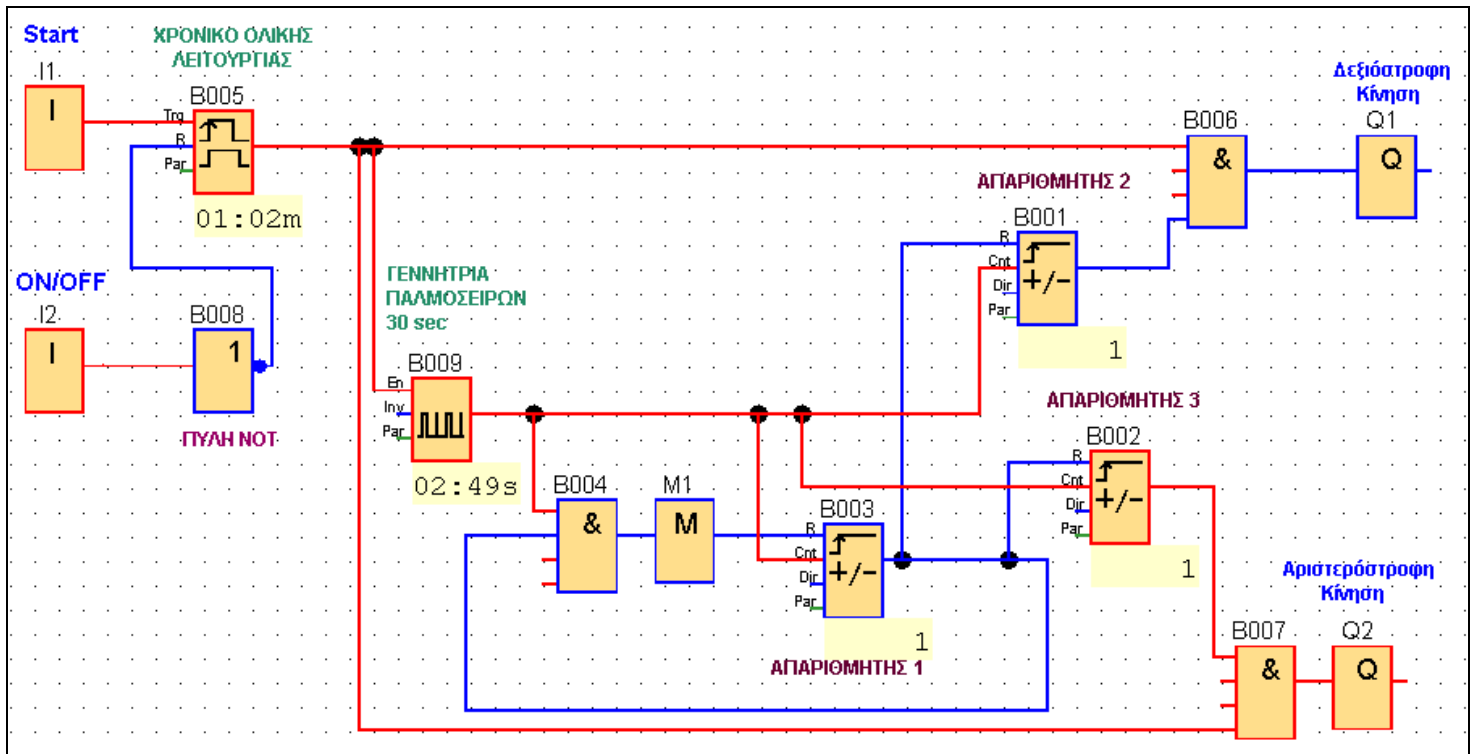
Είσοδοι: I1 on, I2 on, **Timer:** 00:03 min, **Pulse Generator:** 03:62 sec, **Εξοδοι:** Q1 off, Q2 on
Counters: C1, C2, C3: 1



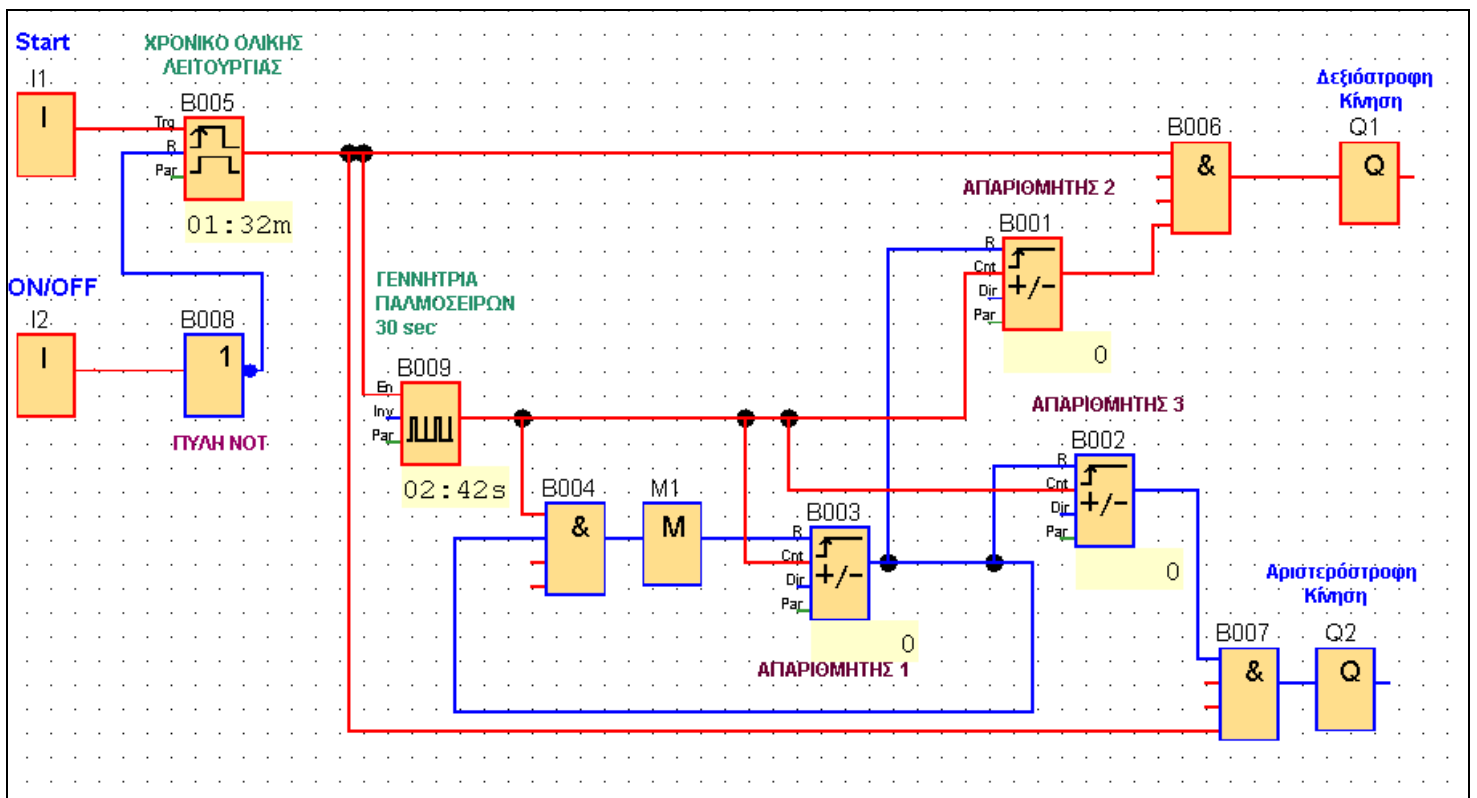
Είσοδοι: I1 on, I2 on, **Timer:** 00:31 min, **Pulse Generator:** 01:70 sec, **Εξοδοι:** Q1 on, Q2 off
Counters: C1, C2, C3: 0



Είσοδοι: I1 on, I2 on, **Timer:** 01:49 min, **Pulse Generator:** 02:49 sec, **Έξοδοι:** Q1 off, Q2 on
Counters: C1, C2, C3: 1

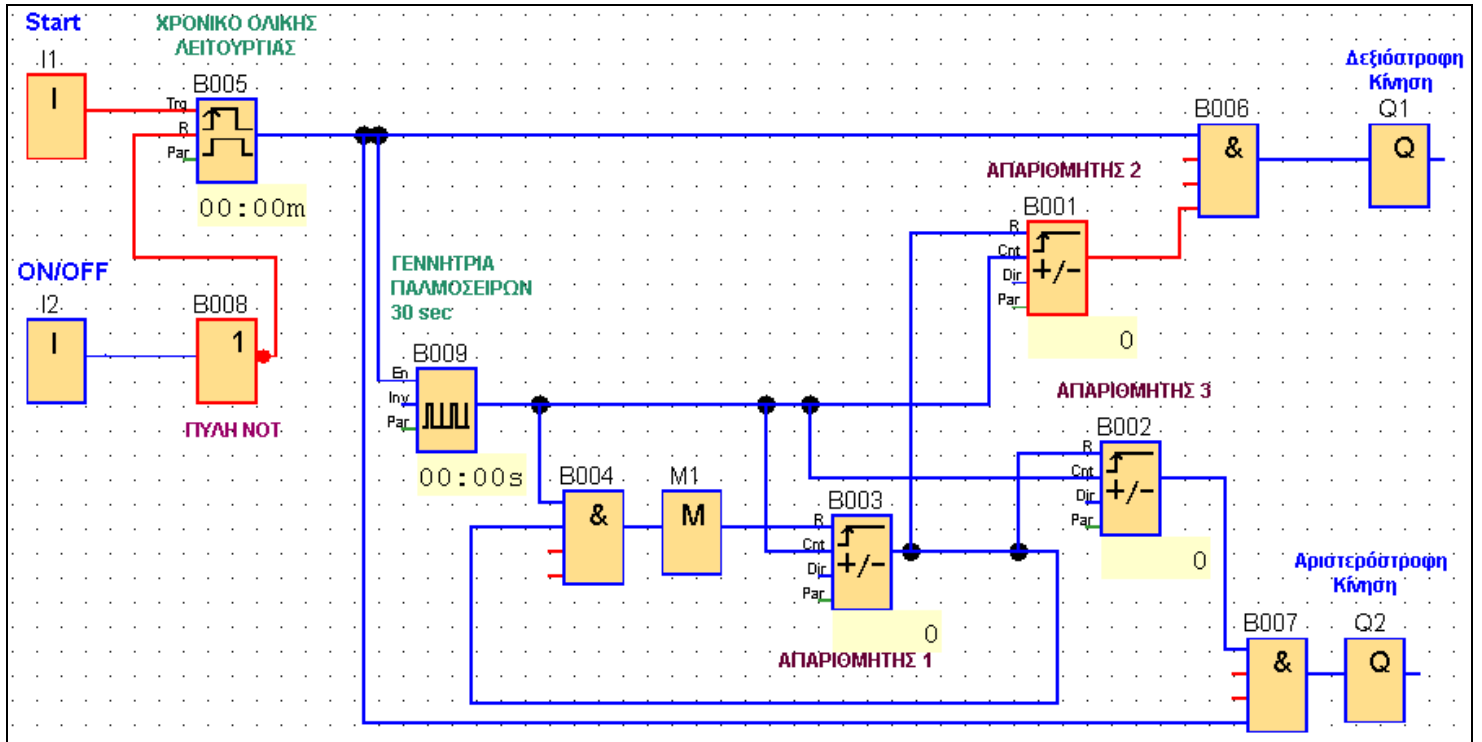


Είσοδοι: I1 on, I2 on, **Timer:** 01:32 min, **Pulse Generator:** 02:42 sec, **Έξοδοι:** Q1 on, Q2 off
Counters: C1, C2, C3: 0



Είσοδοι: I1 on, I2 off, **Timer:** 00:00 min (reset & off), **Pulse Generator:** 00:00 sec (reset & off)

Έξοδοι: Q1 off, Q2 off, **Counters:** C1, C2, C3: 0



Δ. ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ / SET - RESET

1. Ζητείται να φτιαχτεί ένα πρόγραμμα που να ελέγχει την λειτουργία 4 Βανών ενός αυτόματου συστήματος **ποτίσματος**.

Συμφώνα με αυτό το πρόγραμμα πατώντας το μπουτόν **start** θα λειτουργεί η πρώτη βάνα για προκαθορισμένο χρονικό διάστημα (πχ 20 sec) και μετά θα λειτουργεί η δεύτερη για το ίδιο χρονικό διάστημα, μετά η τρίτη **κ.ο.κ** .

Η διαδικασία αυτή να επαναλαμβάνεται έως ότου τελειώσει ο ολικός χρόνος ποτίσματος (πχ 80 sec = 1 cycle) και να διακόπτεται χειροκίνητα από τον χρήστη σε οποιοδήποτε χρονικό σημείο.

Σημείωση: Να υλοποιηθεί με χρήση του block Asynchronous Pulse Generator καθώς και των latching relay και pulse relay.

ΛΥΣΗ:

ΕΙΣΟΔΟΙ: ON/OFF, **I1**

Start, **I2**

ΕΞΟΔΟΙ: BANA 1, **Q1**

BANA 2, **Q2**

BANA 3, **Q3**

BANA 4, **Q4**

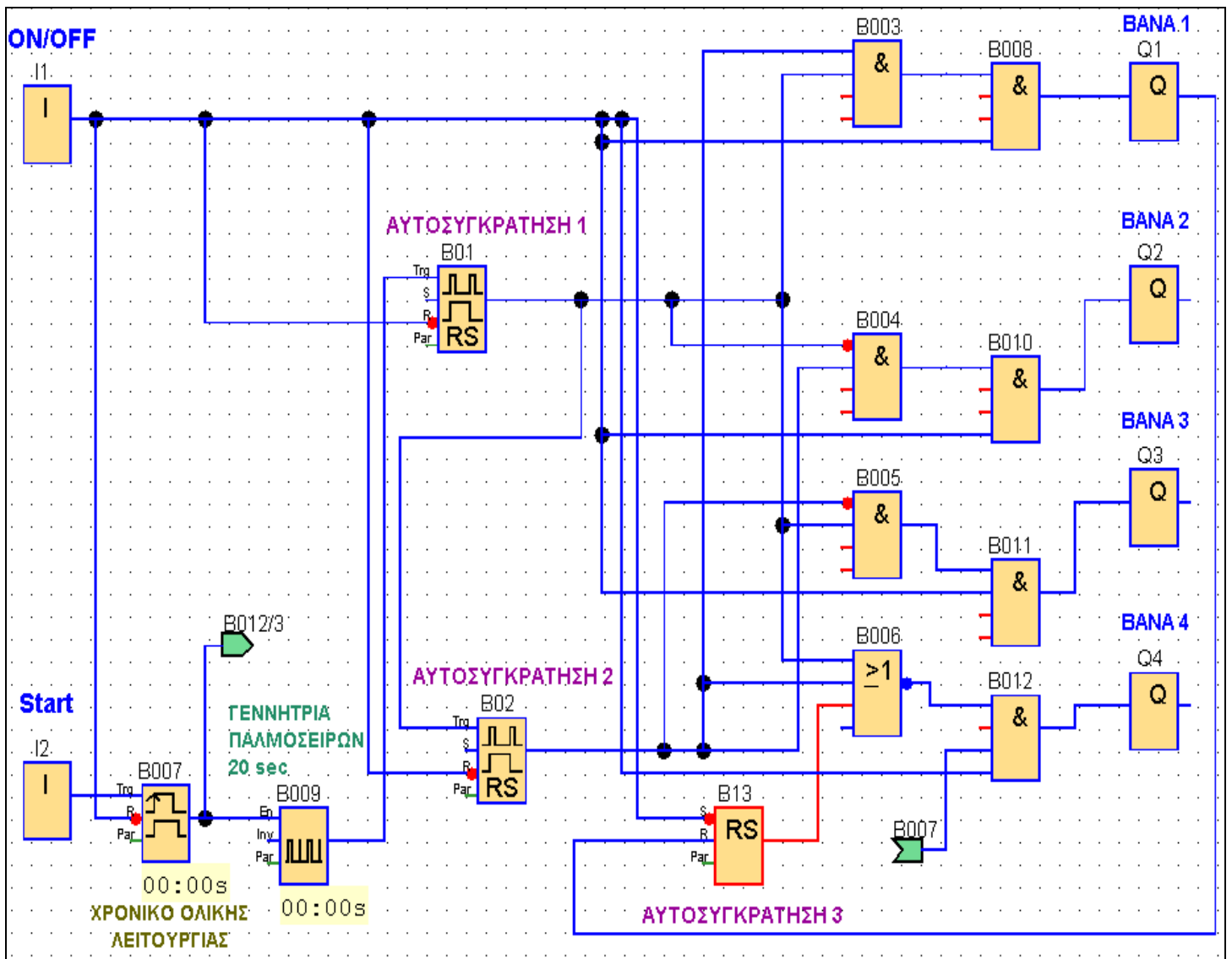
Είσοδοι: I1 off, I2 off

Timer: 00:00 sec

Pulse Generator: 00:00 sec

ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ: 1, 2 RESET, 3 SET

Έξοδοι: Q1 off, Q2 off, Q3 off, Q4 off



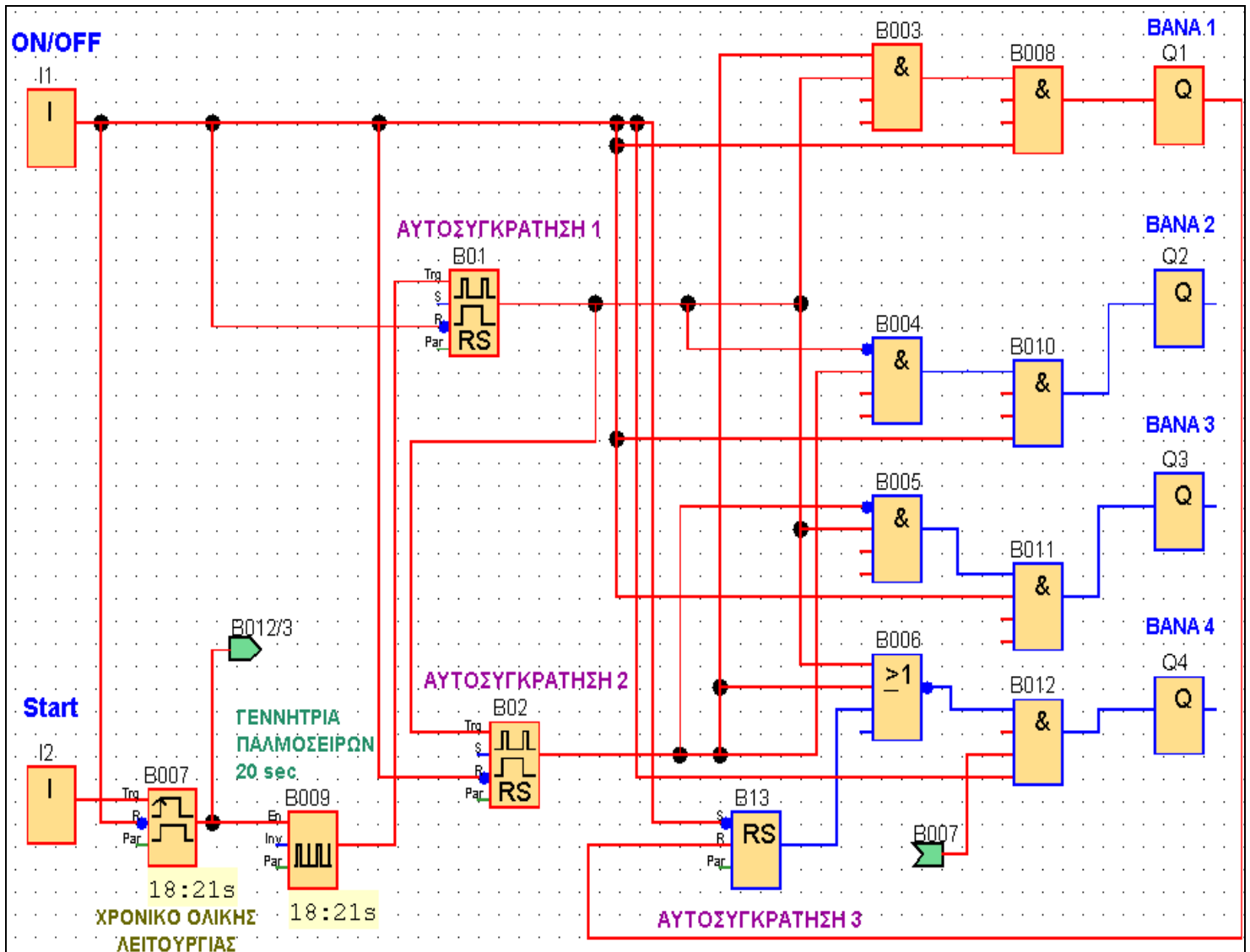
Είσοδοι: I1 on, I2 on (instant function)

Timer: 18:21 sec

Pulse Generator: 18:21 sec

ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ: 1, 2 SET, 3 RESET

Έξοδοι: Q1 on, Q2 off, Q3 off, Q4 off



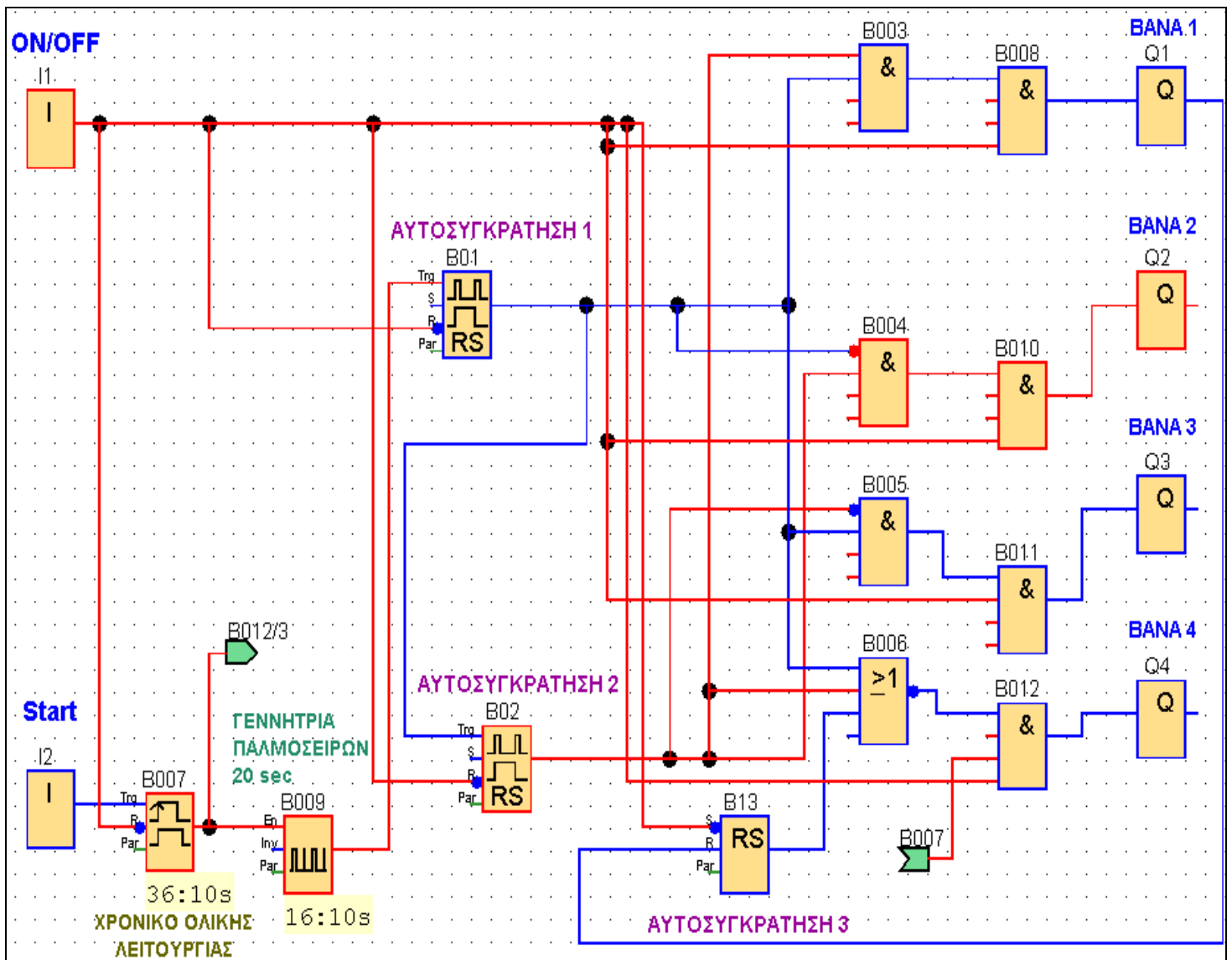
Είσοδοι: I1 on, I2 off

Timer: 36:10 sec

Pulse Generator: 16:10 sec

ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ: 1, 3 RESET, 2 SET

Έξοδοι: Q1 off, Q2 on, Q3 off, Q4 off



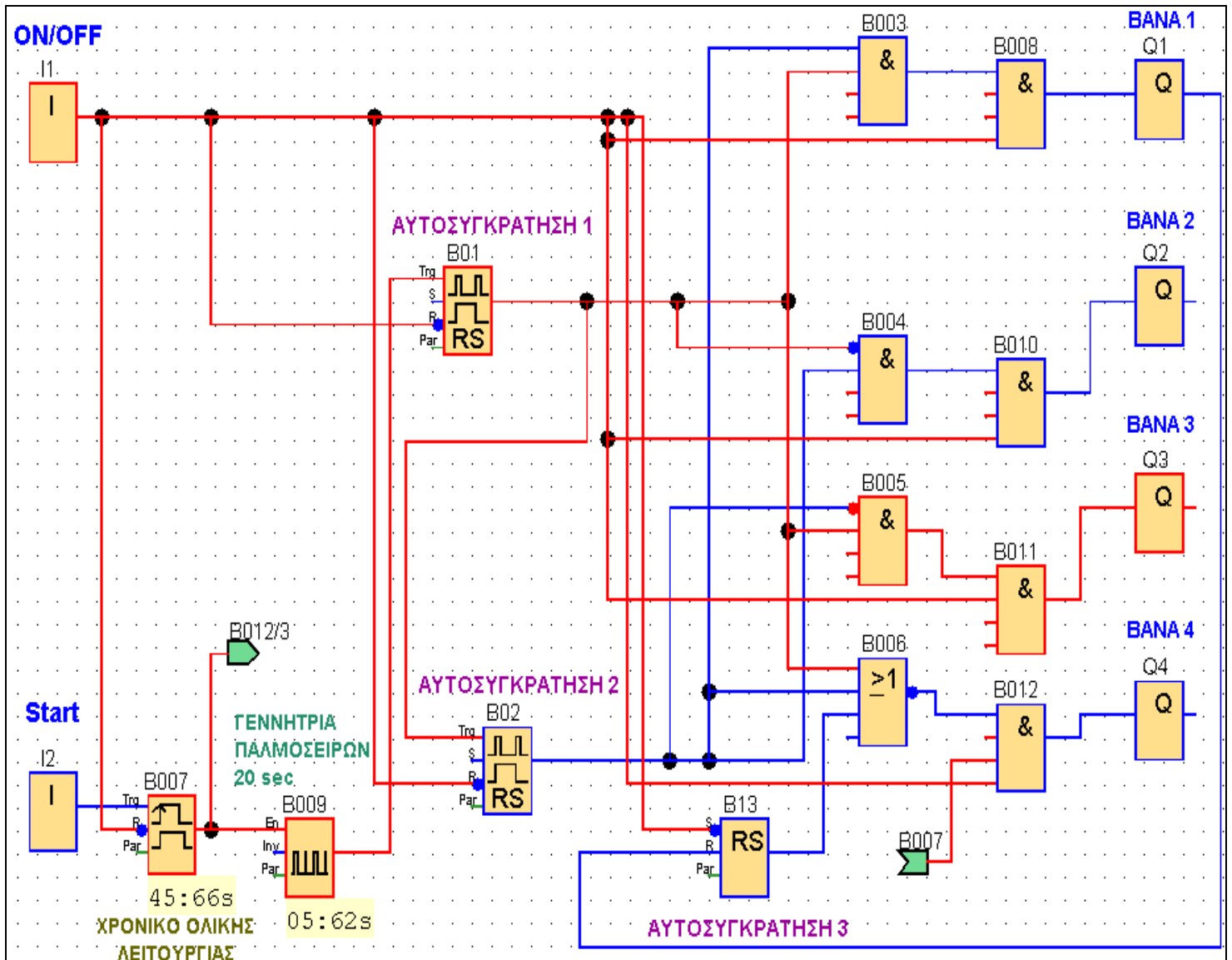
Είσοδοι: I1 on, I2 off

Timer: 45:66 sec

Pulse Generator: 05:62 sec

ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ: 1 SET, 2, 3 RESET

Έξοδοι: Q1 off, Q2 off, Q3 on, Q4 off



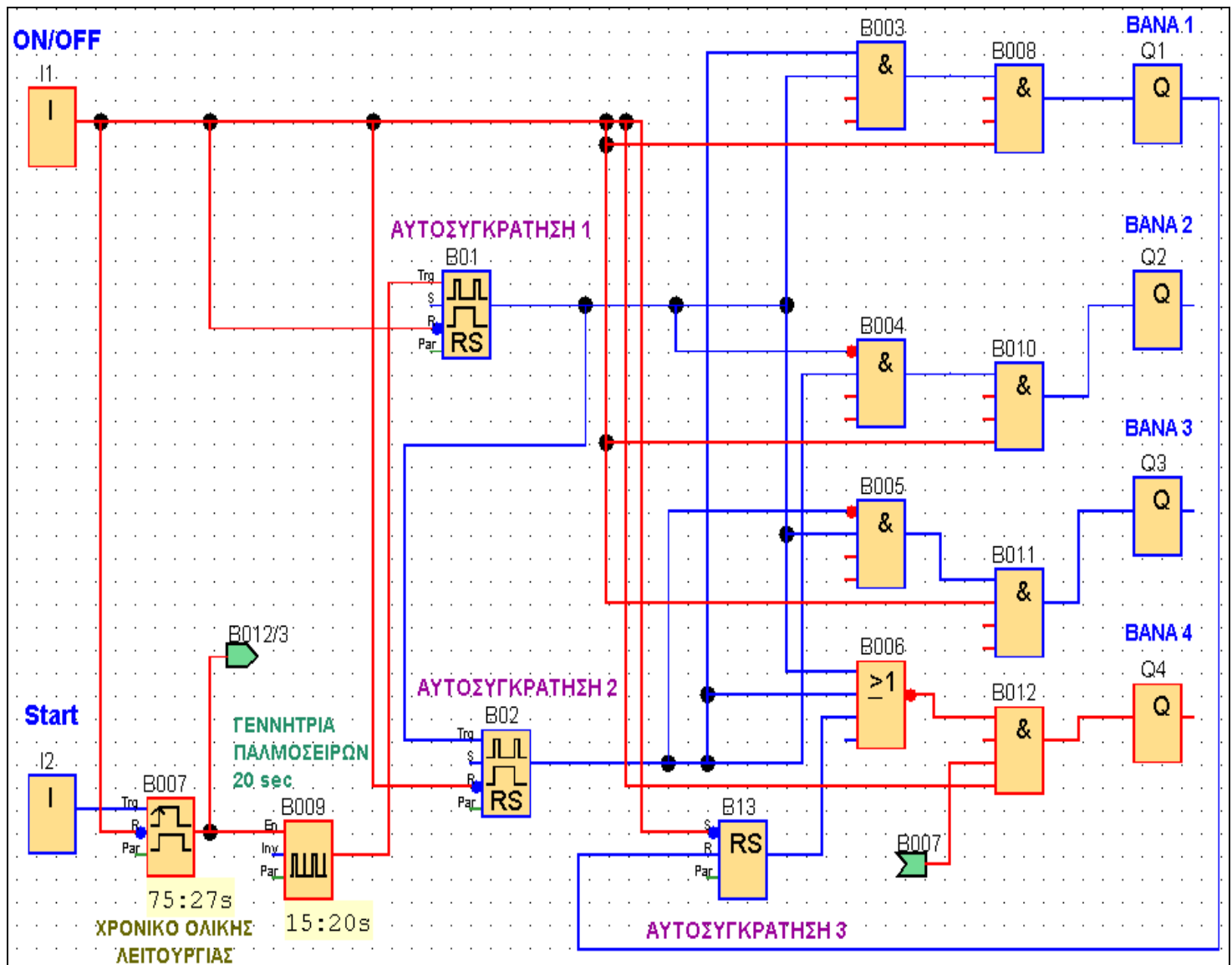
Είσοδοι: I1 on, I2 off

Timer: 75:27 sec

Pulse Generator: 15:20 sec

ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ: 1, 2, 3 RESET

Έξοδοι: Q1 off, Q2 off, Q3 off, Q4 on



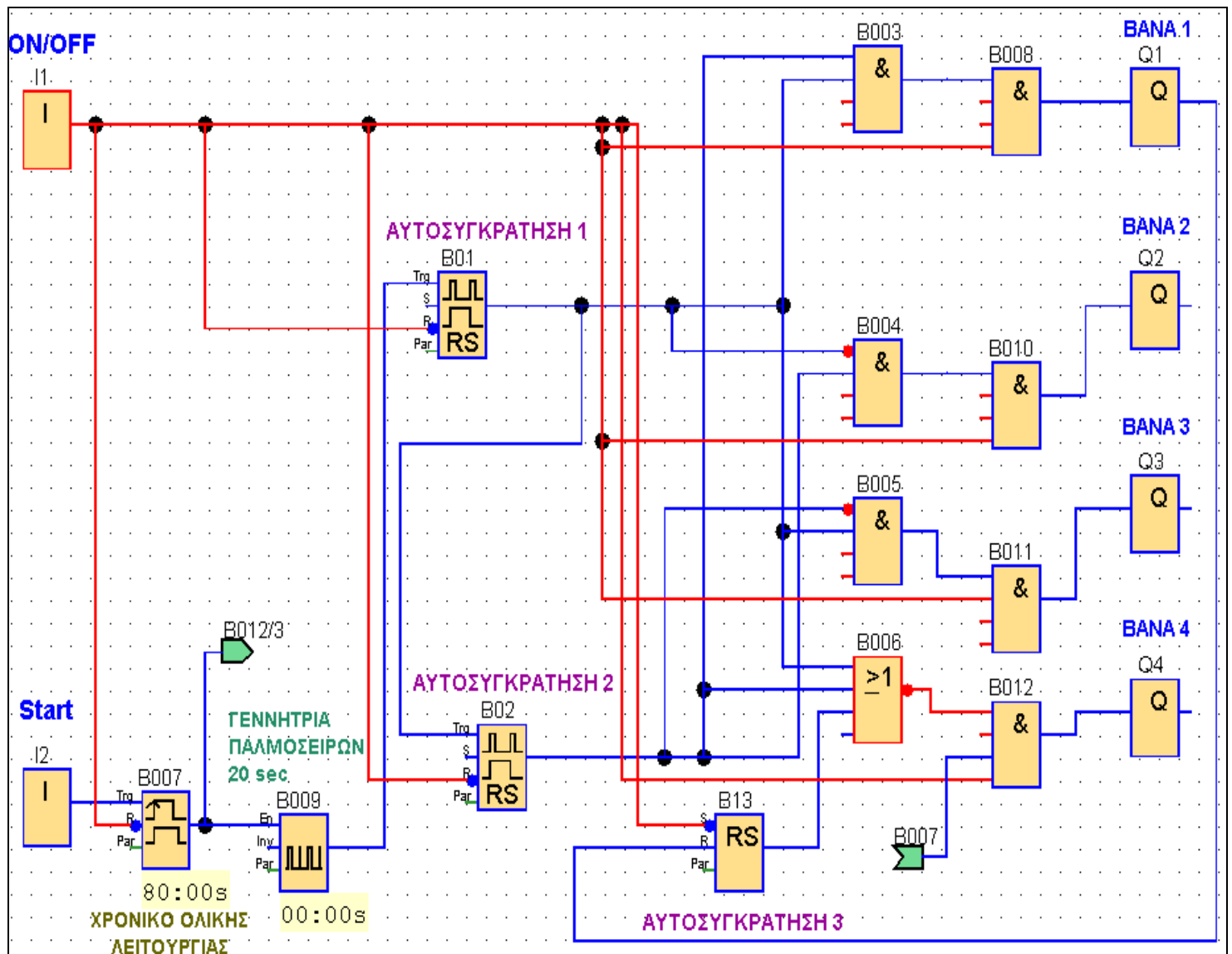
Είσοδοι: I1 on, I2 off

Timer: 80:00 sec

Pulse Generator: 00:00 sec

ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ: 1, 2, 3 RESET

Έξοδοι: Q1 off, Q2 off, Q3 off, Q4 off



2. Ζητείται να φτιαχτεί ένα πρόγραμμα που να ελέγχει την λειτουργία 2 αντλιών ενός αυτόματου συστήματος άντλησης νερού 3 επιπέδων στάθμης.

Σε κάθε επίπεδο στάθμης έχουμε τοποθετήσει απο ένα αισθητήριο πίεσης. Έτσι έχουμε 3 επίπεδα:

Επίπεδο 1: ΣΗΜΕΙΟ ΑΝΑΦΟΡΑΣ
Επίπεδο 2: ΣΤΑΘΜΗ 1
Επίπεδο 3: ΣΤΑΘΜΗ 2

Συμφώνα με το πρόγραμμα όταν ανιχνευτεί νερό στο επίπεδο 1 **και** στο επίπεδο 2 θα λειτουργεί η πρώτη αντλία (αντλία 1) και θα συνεχίζει να λειτουργεί μέχρις ότου στο επίπεδο 1 σταματήσει να ανιχνεύεται νερό.

Αν τώρα ανιχνευτεί νερό και στο επίπεδο 3 τότε θα λειτουργήσει και η δεύτερη αντλία.

Τέλος και οι δύο αντλίες θα σταματήσουν να λειτουργούν όταν στο επίπεδο 1 δεν θα ανιχνεύεται νερό.

Αν για οποιαδήποτε λόγο ανιχνευτεί ξανά νερό στο επίπεδο 1 **και** στο επίπεδο 2 (2nd cycle) τότε θα λειτουργήσει σαν ‘πρώτη’ η αντλία που χρησιμοποιήθηκε σαν ‘δεύτερη’ στον προηγούμενο κύκλο λειτουργίας **και εναλλάξ.**

ΛΥΣΗ:

ΕΙΣΟΔΟΙ: ΑΙΣΘΗΤΗΡΙΑ // ΣΗΜΕΙΟ ΑΝΑΦΟΡΑΣ, ΣΤΑΘΜΗ 1, ΣΤΑΘΜΗ 2: **I1, I2, I3**

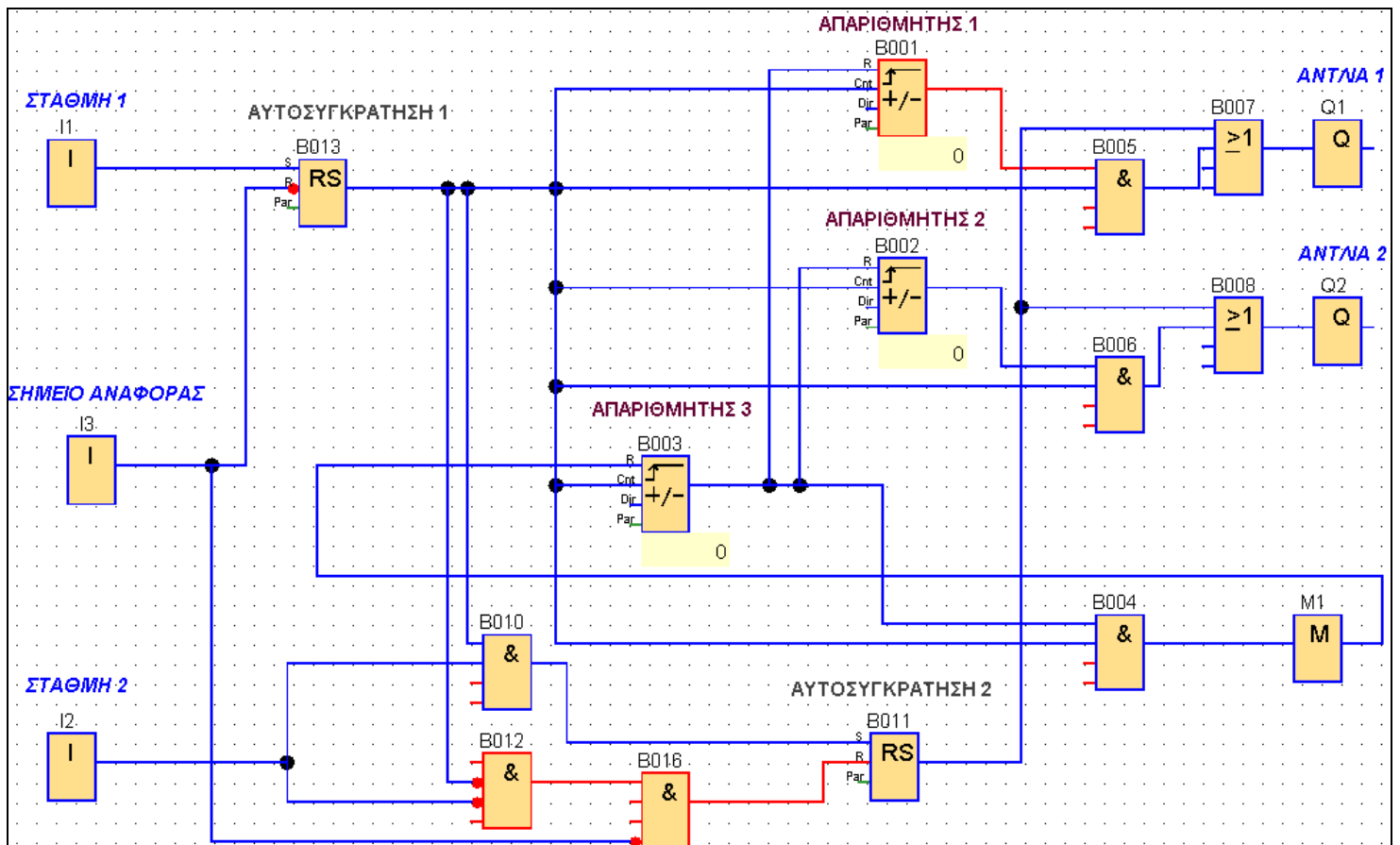
ΕΞΟΔΟΙ: ΑΝΤΛΙΕΣ, **Q1, Q2**

Είσοδοι: I1 off, I2 off, I3 off,

Counters: C1, C2, C3: 0, 0, 0

ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ: 1 RESET, 2 RESET

Έξοδοι: Q1 off, Q2 off

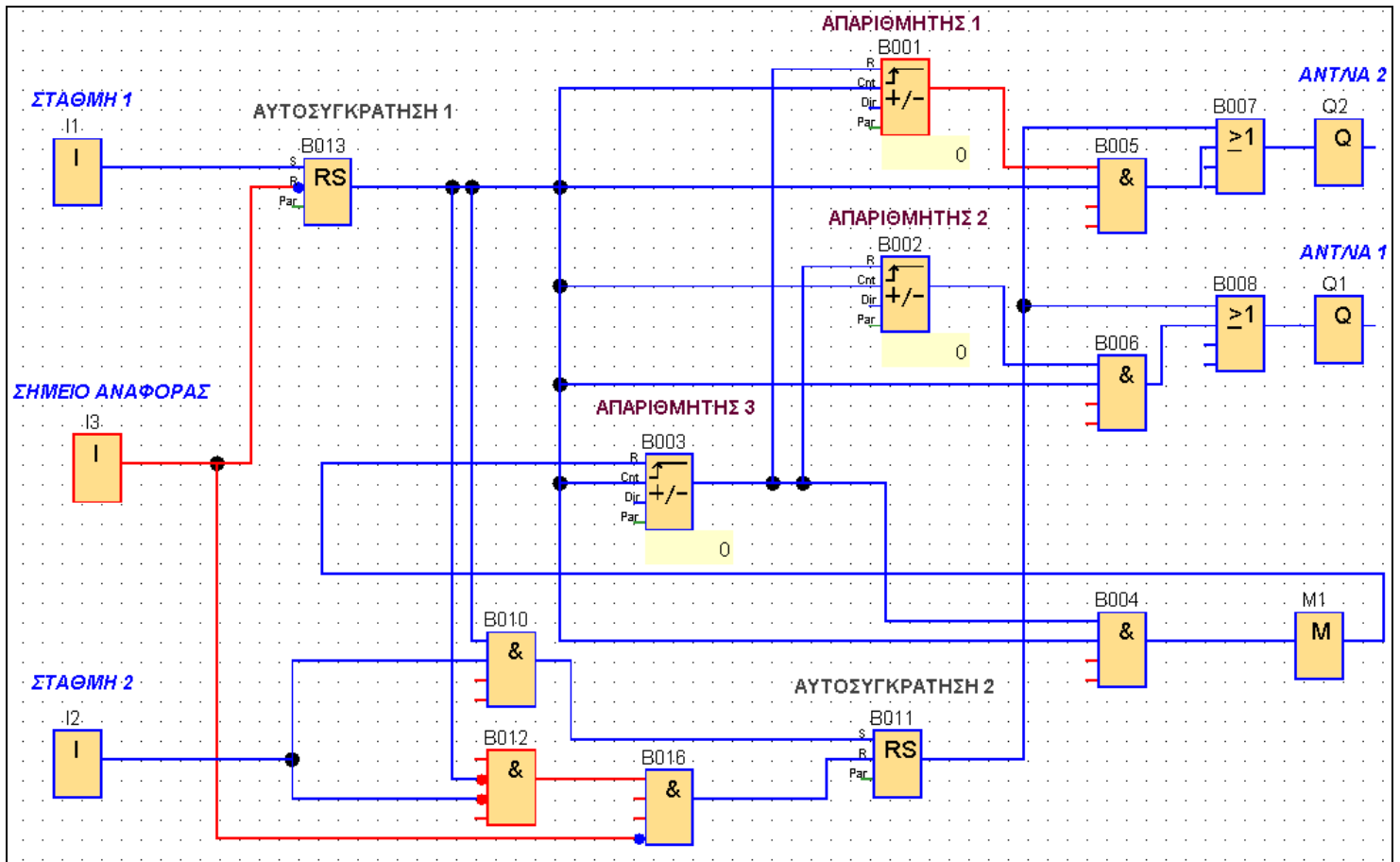


Είσοδοι: I1 off, I2 off, I3 on,

Counters: C1, C2, C3: 0, 0, 0

ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ: 1 RESET, 2 RESET

Έξοδοι: Q1 off, Q2 off

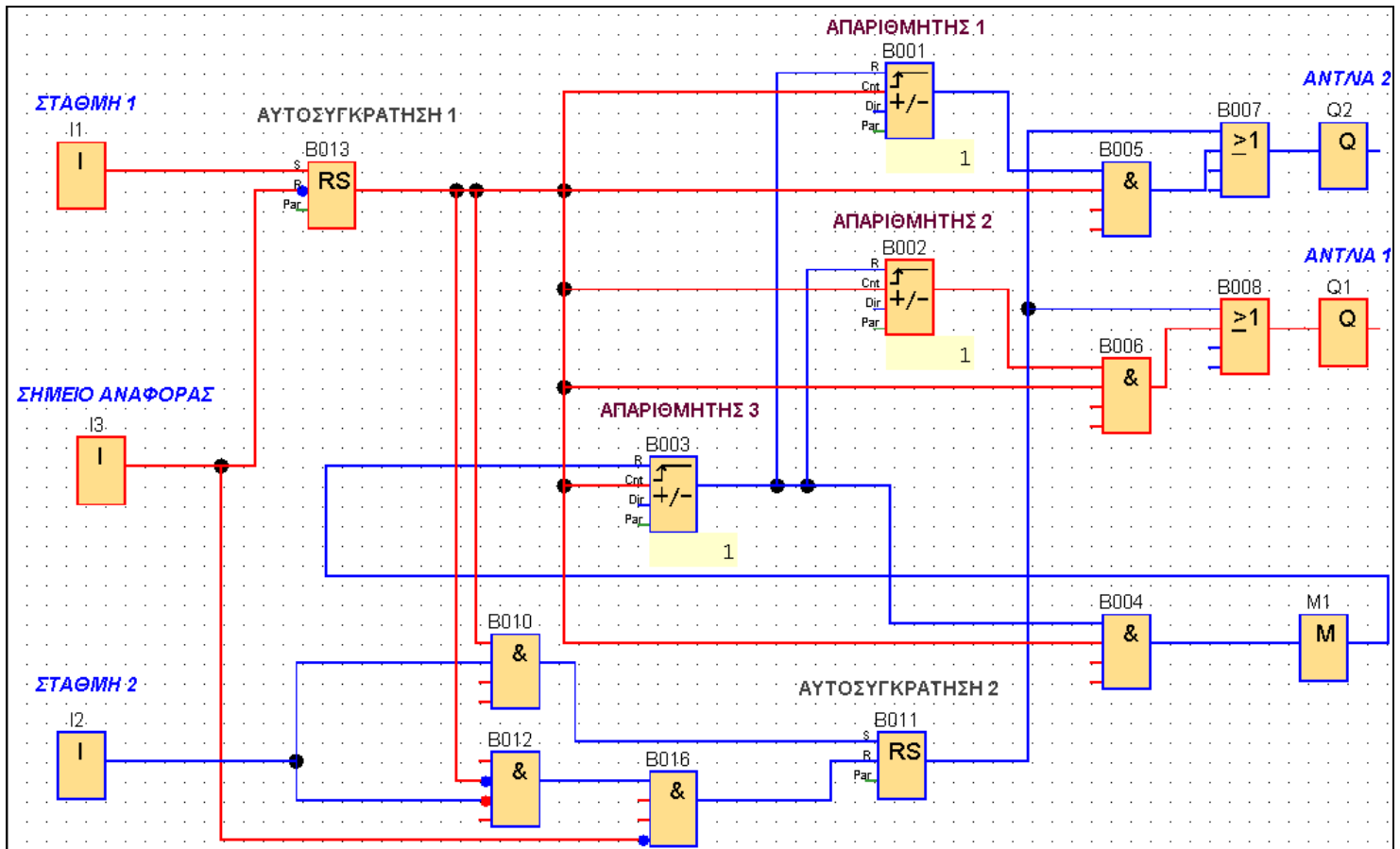


Είσοδοι: I1 on, I2 off, I3 on,

Counters: C1, C2, C3: 1, 1, 1

ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ: 1 SET, 2 RESET

Έξοδοι: Q1 on, Q2 off

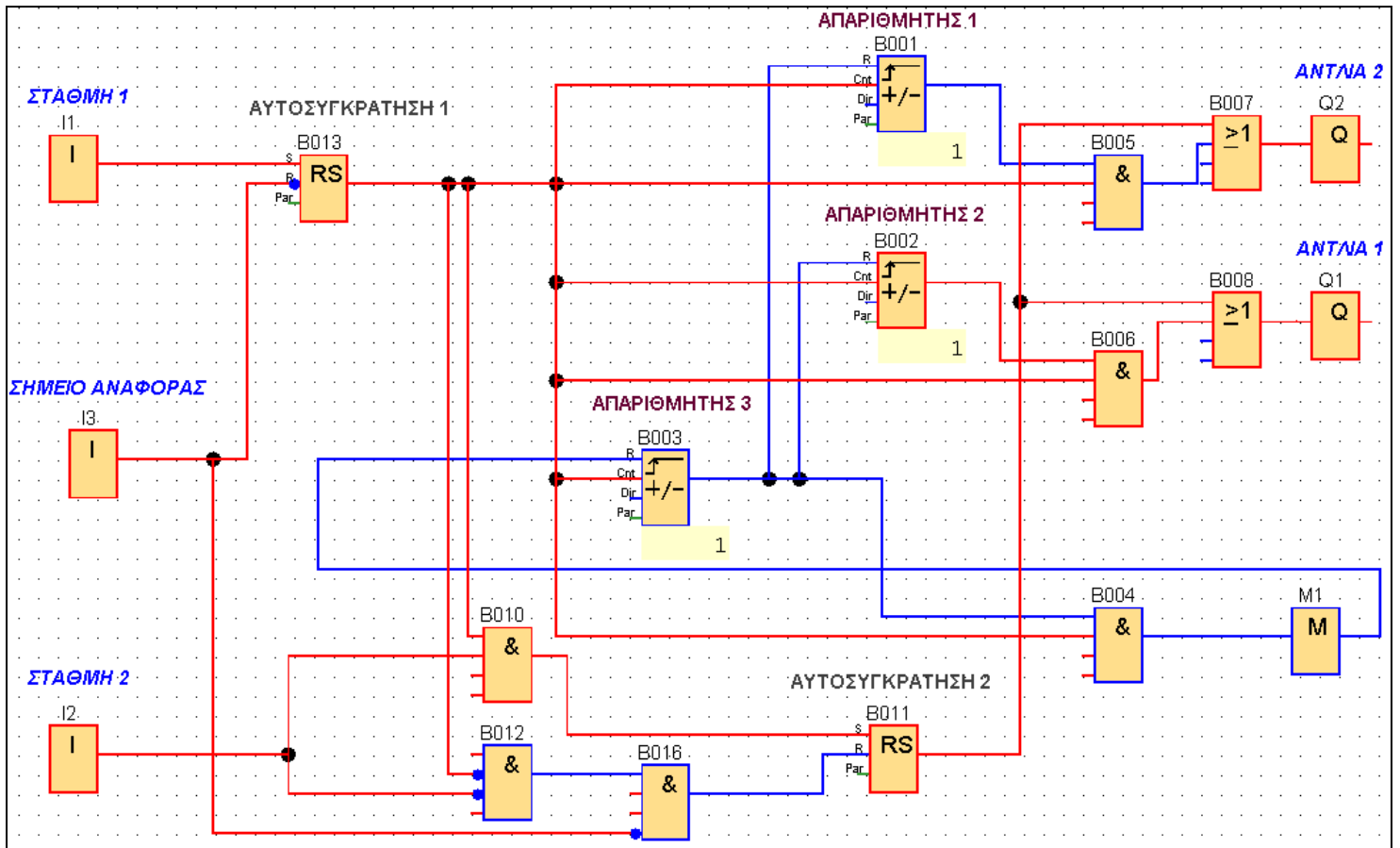


Είσοδοι: I1 on, I2 on, I3 on,

Counters: C1, C2, C3: 1, 1, 1

ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ: 1 SET, 2 SET

Έξοδοι: Q1 on, Q2 on

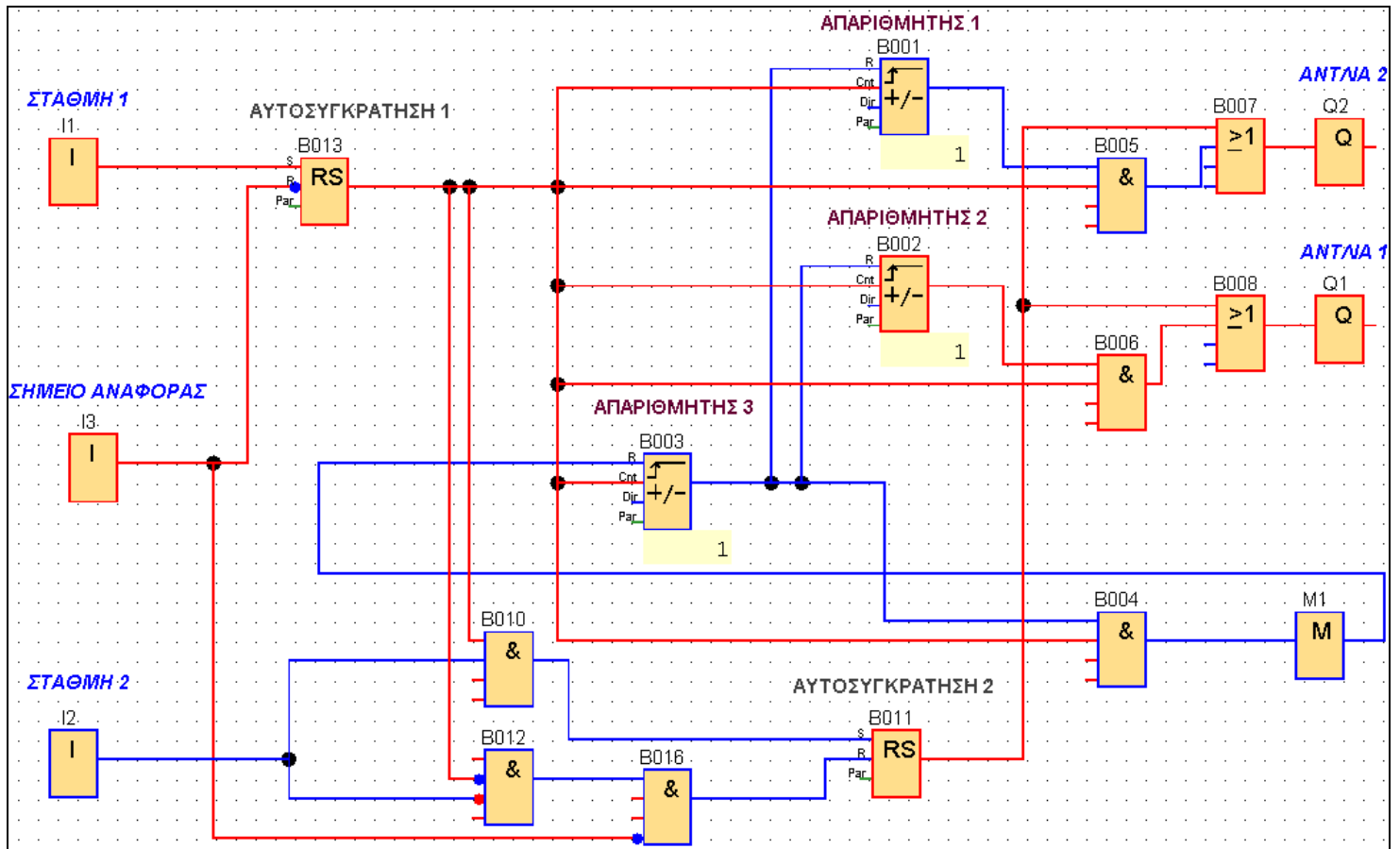


Είσοδοι: I1 on, I2 off, I3 on,

Counters: C1, C2, C3: 1, 1, 1

ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ: 1 SET, 2 SET

Έξοδοι: Q1 on, Q2 on

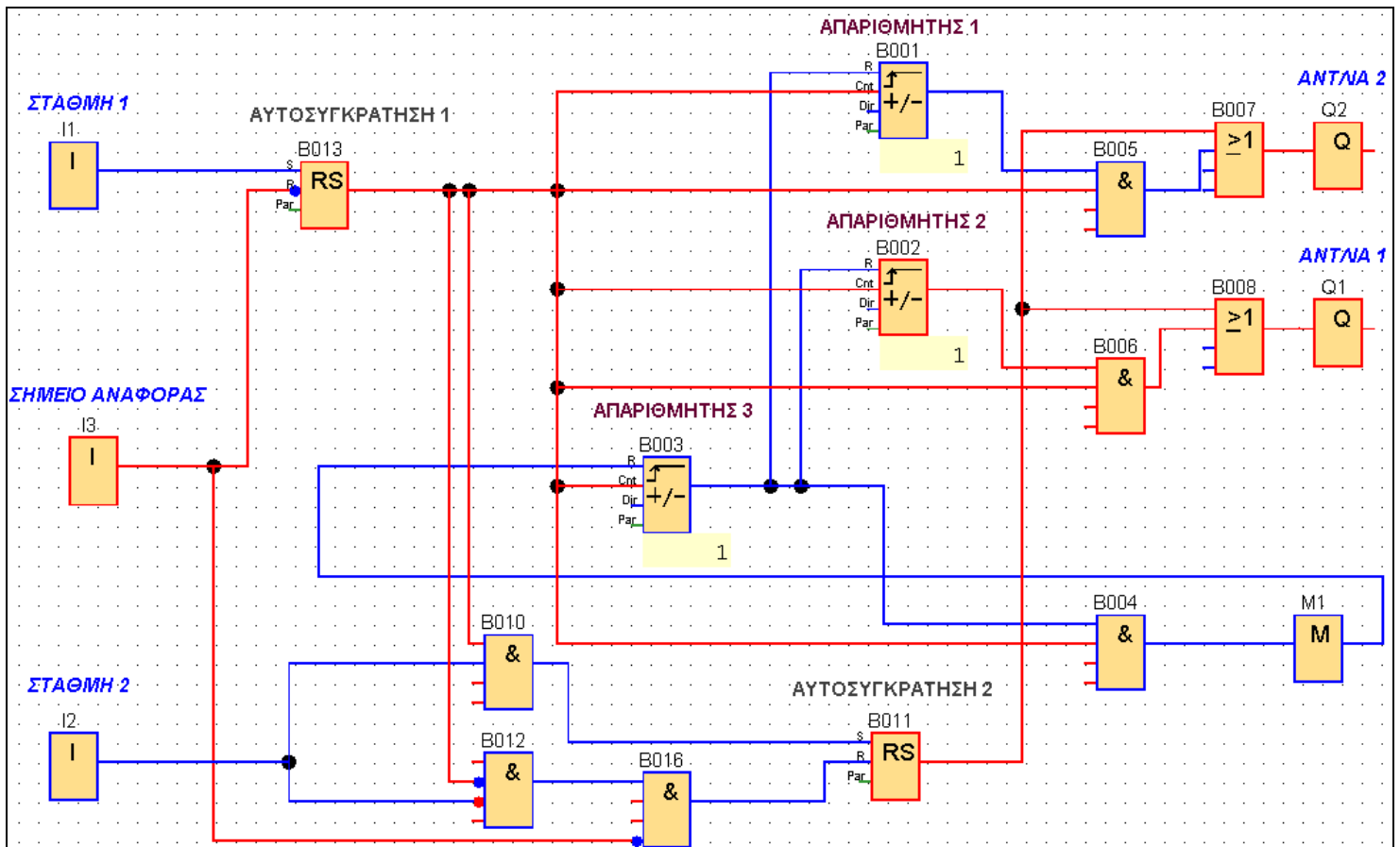


Είσοδοι: I1 off, I2 off, I3 on,

Counters: C1, C2, C3: 1, 1, 1

ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ: 1 SET, 2 SET

Έξοδοι: Q1 on, Q2 on

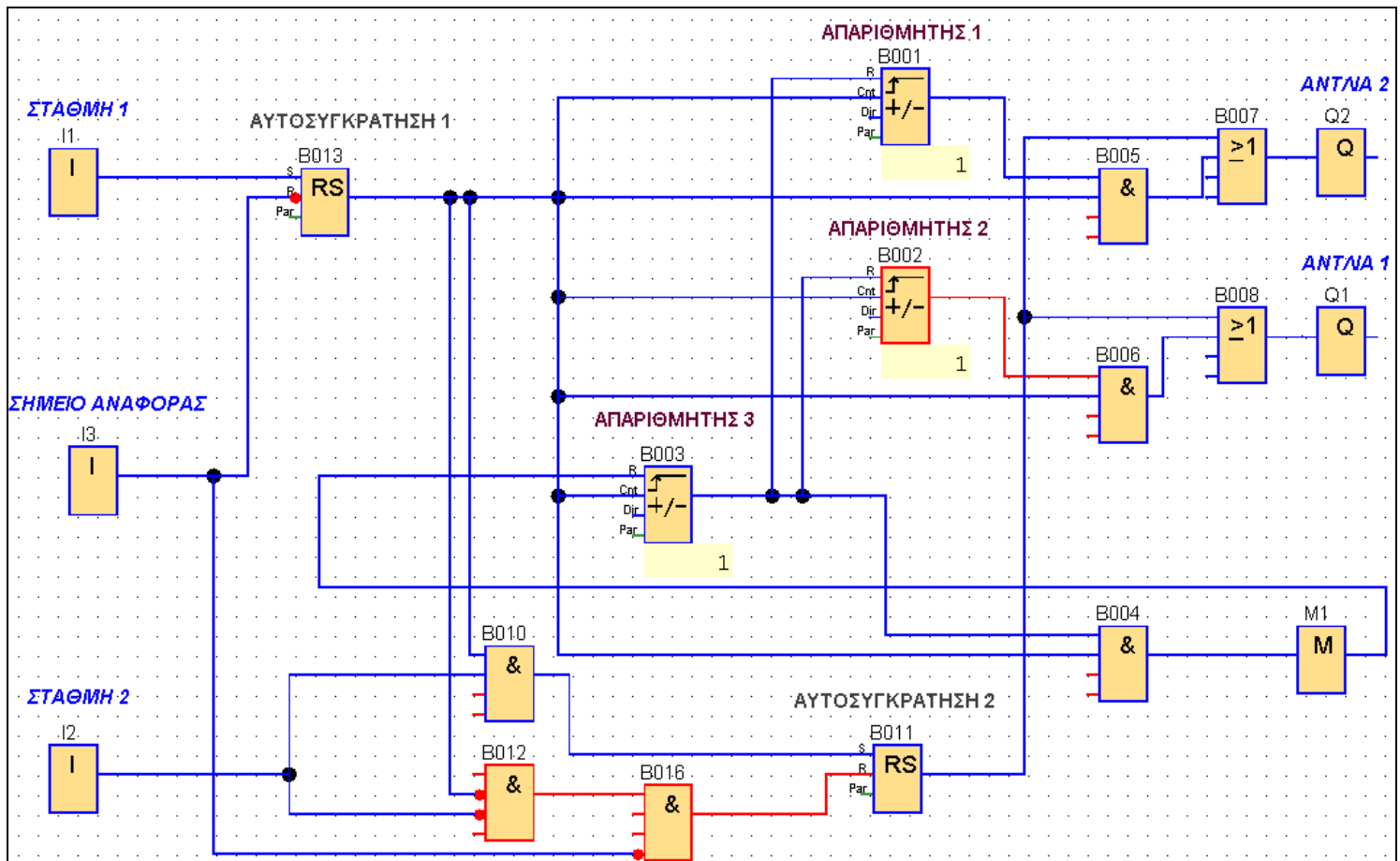


Είσοδοι: I1 off, I2 off, I3 off,

Counters: C1, C2, C3: 1, 1, 1

ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ: 1 RESET, 2 RESET

Έξοδοι: Q1 off, Q2 off

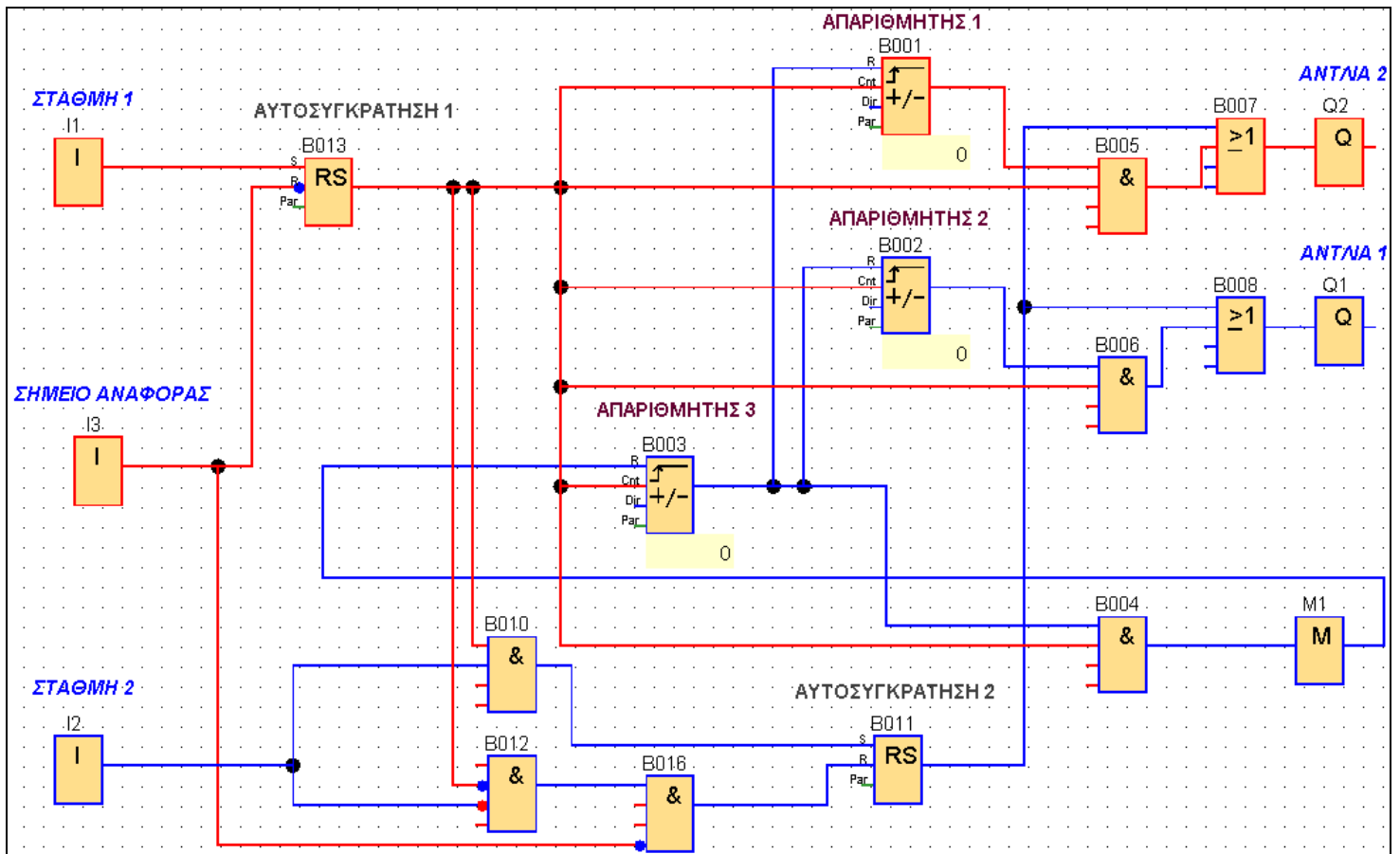


Είσοδοι: I1 on, I2 off, I3 on,

Counters: C1, C2, C3: 0, 0, 0

ΑΥΤΟΣΥΓΚΡΑΤΗΣΗ: 1 SET, 2 RESET

Έξοδοι: Q1 off, Q2 on



...