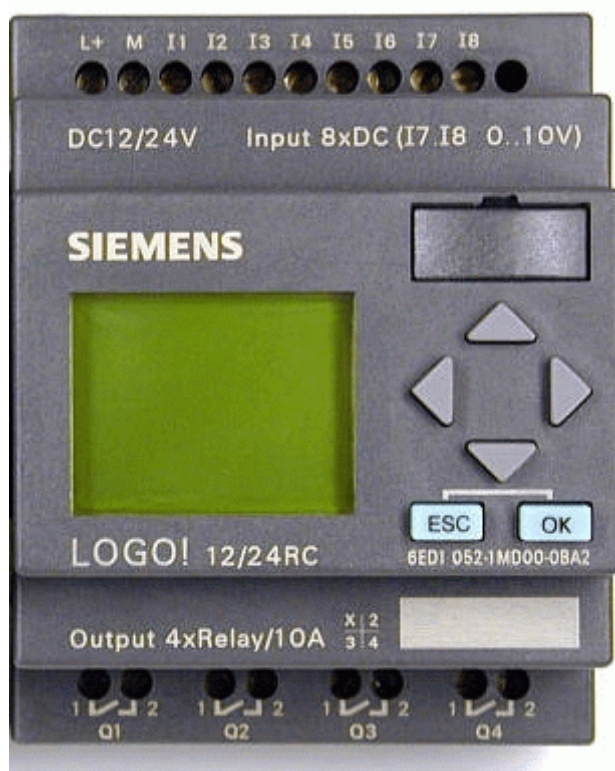




# Základy programování LOGO!

Jiří Skála



Tento projekt je spolufinancován Evropským sociálním fondem a státním rozpočtem České republiky.

# Obsah

1. Úvod.....	6
2. Odkazy.....	7
3. Univerzální logický modul LOGO!.....	8
3.1. Základní myšlenka.....	8
Stručná charakteristika.....	8
3.2. Identifikátory určující model 12/24RC OBA5.....	8
Základní modul LOGO! Basic.....	8
Obrázek a popis základního modulu LOGO! Basic.....	9
3.3. Rozšiřující moduly.....	9
3.4. Napájecí zdroj a jištění zdroje.....	10
Napájení a jištění modulu.....	10
3.5. Montáž.....	10
Bezpečnostní předpisy.....	10
Pravidla pro připojení vodičů.....	11
3.6. Připojení vstupů.....	11
Připojení výstupů.....	12
Maximální proudové zatížení výstupních kontaktů.....	12
Dovolená spínací frekvence výstupních kontaktů.....	13
3.7. Zapnutí LOGO!.....	13
Obrazovky běžícího programu.....	13
4. Programování LOGO!.....	13
Grafické znázornění struktury menu.....	14
Podrobnější popis jednotlivých úrovní menu.....	14
Kurzor, jeho pohyb a tvar.....	17
Nastavení reálného času a data.....	17
4.1. Čtyři seznamy prvků použitelných při programování.....	18
Seznam Co.....	18
Seznam GF.....	19
Seznam SF.....	19
Seznam Bn.....	19
Přiřazení čísla bloku.....	19
Zobrazení bloku na displeji.....	19
4.2. Postup při ručním programování.....	20
Příklad 1. Úvodní.....	20
Příklad 2. Použití seznamu Bn.....	24
Příklad 3. Konektory Q1, Q2 zapojené v sérii.....	25
Příklad 4.....	26
Příklad 5.....	27
Invertování vstupu při ručním programování.....	28
Zrušení invertovaného vstupu.....	28
4.3. Pracovní režimy LOGO!.....	29
Programovací režim.....	29
Režim RUN.....	29
Režim přiřazení parametrů.....	29
4.4. Editace hotového programu.....	30
Vložení nového bloku do stávajícího programu.....	31
Nastavení parametru bloku při programování.....	32

Přepnutí do režimu přiřazení parametrů za běhu programu.....	32
Změna parametru bloku za běhu programu.....	33
Potlačení režimu přiřazení parametrů.....	33
Časová základna.....	33
4.5. Pojmenování a ochrana programu.....	34
Přiřazení názvu programu.....	34
Ochrana programu heslem.....	35
Heslo změna.....	36
Heslo deaktivace.....	36
4.6. Ruční programování podle výkresu.....	37
Příklad 5. Obr. 36.....	37
Příklad 6. Obr. 37.....	38
Příklad 7. Obr. 38.....	39
Příklad 8. Obr. 39.....	40
Příklad 9. Obr. 40.....	41
4.7. Náměty k opakování.....	42
5. Základní funkce 1. část.....	43
5.1. AND, logický součin.....	44
5.2. NAND, negovaný součin.....	46
5.3. OR, logický součet.....	47
5.4. NOR, negovaný součet.....	48
5.5. NOT, negace, invertor.....	49
5.6. XOR, exkluzivní OR.....	50
5.7. Řešené příklady.....	51
Příklad 10.....	51
Příklad 11. Sepnutí výstupu je podmíněno sepnutím jiného výstupu.....	52
Příklad 12. Když je sepnutý jeden výstup, nesmí sepnout druhý.....	52
Příklad 13.....	53
5.8. Příklady na procvičení.....	54
Příklad 14.....	54
Příklad 15.....	54
Příklad 16.....	54
Příklad 17.....	54
Příklad 18.....	54
Příklad 19.....	54
Příklad 20.....	54
5.9. Opakování, základní funkce.....	55
6. Speciální funkce 1. část.....	56
Vstupy speciálních funkcí.....	57
Logické vstupy.....	57
Parametrové vstupy.....	57
Přesnost času .....	58
Chod vnitřních hodin při výpadku napájení.....	58
Chybová hlášení LogoSoftComfort.....	58
Náběžná a sestupná hrana signálu.....	58
6.1. Blok samodržené relé RS.....	59
6.1.1. Řešené příklady.....	61
Příklad 21. Běh motoru jedním směrem.....	61
Příklad 22. Reverzace otáček.....	62

Příklad 23. Hlavní motor stroje a čerpadlo chladící emulze.....	64
6.1.2. Příklad na procvičení.....	65
Příklad 24. Hlavní a pomocný motor.....	65
6.2. Blok zpožděného zapnutí.....	66
6.2.1. Řešený příklad.....	68
Příklad 25. Postupné sepnutí více výstupů a chybová hlášení.....	68
6.2.2. Příklady na procvičení.....	70
Příklad 26. Hlavní motor a kompresor.....	70
Příklad 27. Cyklus.....	70
6.3. Blok zpožděného vypnutí.....	71
6.4. Příklady na procvičení.....	73
Příklad 28. Spínání výstupu se zpožděním při sepnutí a při vypnutí.....	73
Příklad 29. Ovládání dvou dopravníků.....	73
Příklad 30. Garážová vrata.....	73
Příklad 31. Rozběh Y/D.....	74
Příklad 32. Pásová doprava.....	75
Příklad 33. Odstředivka.....	75
Příklad 34. Postupné spouštění.....	76
Příklad 35. Semafor.....	76
Příklad 36. Pohyb stolu brusky.....	77
7. Základní funkce 2. část.....	78
7.1. AND s detekcí náběžné hrany.....	78
7.1.1. Řešený příklad.....	79
Příklad 37. Varianty ovládání výstupu jedním tlačítkem.....	79
7.2. NAND s detekcí sestupné hrany.....	80
7.2.1. Řešené příklady.....	81
Příklad 38. Spouštění YD.....	81
Příklad 38. Reverzace otáček motoru a brzdění protiproudem.....	82
Příklad 39.....	82
8. Speciální funkce 2. část.....	83
8.1. Dopředný a zpětný čítač.....	83
8.1.1. Nastavení parametru jiným, již nakonfigurovaným blokem.....	85
Ruční programování.....	85
Nastavitelná časová základna.....	86
Nastavení pomocí software.....	87
8.1.1. Řešené příklady.....	88
Příklad 40. Postupné spínání více výstupů, vypnutí dosaženým počtem.....	88
Příklad 41. Využití vstupu Dir.....	88
8.1.2. Příklady na procvičení.....	90
Příklad 42. Postupné spínání.....	90
Příklad 43. Cyklus: Sepnutí a vypnutí výstupu po dosažení určeného počtu.....	90
Příklad 44. Zablokování funkce dosaženým počtem.....	90
Příklad 45. Zablokování funkce počtem pracovních cyklů.....	90
Příklad 46. Čítání vzestupných i sestupných hran.....	90
Příklad 47. Jen určené pořadí a počet sepne výstup.....	90
8.2. Textové zprávy.....	91
Ukázka 1. Neobsazený výstup bloku textové zprávy.....	92
Ukázka 2. Vypnutí Q zablokováno do potvrzení textové zprávy.....	92
Ukázka 3. Opětné zapnutí Q zablokováno do potvrzení textové zprávy.....	92

Ukázka 4. Výstup Q vypne až po potvrzení textové zprávy.....	93
Ukázka 5. Zobrazení přednastaveného textu spolu s parametry jiných bloků.....	93
Konfigurace bloku Textové zprávy softwarově.....	94
Ukázka 6. Priorita.....	95
8.2.1. Příklady na procvičení.....	96
Příklad 48. Aktuální informace o stavu výstupu.....	96
Příklad 49. Aktuální informace, priorita a blokování pomocí nepotvrzené zprávy...	96
Příklad 50. Textová zpráva, aktuální informace, kombinace textu a parametru.....	96
9. Závěrečné příklady.....	97
Příklad 51. Blokování dosaženým počtem nebo časem.....	97
Příklad 52. Cyklus – po opětovném zapnutí program pokračuje z místa vypnutí.....	97
Příklad 53. Výstupy spínají a vypínají střídavě 10 sekund.....	98
Příklad 54. Sepnutí výstupu blokováno textovou zprávou.....	98

## 1. Úvod

*Parkoviště je automaticky osvětleno v pracovní dny po setmění, nebo od 19:00 do 23:00 hodin a potom od 04:00 do 08:00 hod, nebo do rozednění. Od 23:01 hod. do 03:59 hod. se rozsvítí světla na dobu pěti minut po zaznamenaném pohybu v prostoru parkoviště. Zároveň je vyrozuměna ostraha.*

*O víkendech a celozávodní dovolené se od setmění do rozednění světla rozsvěcují na dobu pěti minut jen při zaznamenaném pohybu na parkovišti o zároveň je upozorněna ostraha.*

Řešení je možné jak pomocí klasické reléové techniky, tak pomocí univerzálního programovacího modulu.

V klasické reléové technice jsou vstupní elektrické signály senzorů přenášeny na výstupní akční členy pomocí vodičů. Funkci obvodu realizují různé konkrétní přístroje. Pomocná a časová relé, roční hodiny, atd.

V univerzálním programovacím modulu tytéž signály řídí program. Přístroje jsou nahrazeny předprogramovanými funkcemi. Ušetří se značné množství přístrojů, vodičů, času a tím peněz.

**Výhoda programovacího modulu vynikne při složitějších aplikacích a častých změnách programu.**

### **Univerzální logické moduly, mikrosystémy.**

Jsou to cenově výhodné řídicí prvky pro širokou třídu aplikací spodní části výkonnostního spektra se základní společnou myšlenkou. „Jednoduchost v řešení, ovládní, programování.“

V současné době dodává na trh univerzální programovací moduly více výrobců, patří mezi ně:

ABB: [http://www.vaeprosyst.cz/logicke\\_moduly\\_cz.htm](http://www.vaeprosyst.cz/logicke_moduly_cz.htm)

Lovato : <http://www.lovato.cz/katalog/Lovato/katalog.htm>

Siemens: <http://www1.siemens.cz/ad/current/prezentace/as/micro/>

Z názvu kurzu je zřejmé, že se budeme věnovat programovacímu modulu LOGO! výrobku firmy Siemens.

Předložený text by měl pomoci při ručním i softwarovém programování LOGO! 12/24RC. Byl zpracován v rámci grantového projektu „Centrum pro další vzdělávání v elektrotechnice a energetice“. Projekt byl zahájený 22. února 2006 a potrvá 24 měsíců. Hlavními partnery projektu jsou emz Hanauer s.r.o. a Elektrotechnický cech Plzeňského regionu. Cílem projektu je proškolit zaměstnance firem a přispět ke zvýšení jejich kvalifikace.

### **Co text obsahuje**

Učební text je rozdělený do tří částí. Od části 3.2. se zabývá konkrétním přístrojem LOGO! 12/24RC. Při použití jiného typu je nutné se v [1] seznámit s rozdíly jednotlivých verzí LOGO!.

- První část (kapitola druhá až čtvrtá) popisuje přístroj, jeho parametry, montáž, připojení vstupů a výstupů a ruční programování. Závěr tvoří kontrolní otázky a příklady pro ruční programování.
- Druhá část (kapitola pátá až osmá) obsahuje popis základních bloků (GF) a vybraných bloků speciálních funkcí (SF). Ke každé funkci jsou uvedené jednoduché příklady. Od druhé části se programuje pomocí LogoSoftComfort.
- Třetí část (kapitola devátá) nabízí příklady na procvičení celé probrané látky.

### **Co text není a co neobsahuje.**

Tato příručka nenahrazuje manuál k přístroji ani nepopisuje ovládní programovacího software. Až na výjimky nejsou uváděny odlišnosti starších verzí LOGO!.

### **Pro koho je text určený.**

Text je určený posluchačům, kteří se poprvé setkávají s programovatelným logickým modulem.

## **Kde kurs probíhá**

V SOUE Vejprnická 56, Plzeň.

## **Použitá literatura.**

[1] LOGO! manuál 8.vydání 7/2005

## **Závěrečné slovo účastníkům kurzu.**

Každý problém se dá řešit mnoha různými způsoby a samozřejmě i jinými, než je uvedeno v této příručce. Vezměte to jako výzvu, povolte uzdu své fantazii, hravosti a nebojte se experimentovat.

## **2. Odkazy**

V případě nefunkční adresy na stránky automatizační techniky Siemens zadejte do libovolného vyhledávače slova: mikrosystémy siemens.

### **Mikrosystémy pro automatizační techniku Siemens**

<http://www1.siemens.cz/ad/current/prezentace/as/micro/>

### **Software LOGO!.**

<http://www1.siemens.cz/ad/current/prezentace/as/micro/logo/index.php?mi=20>

### **Dokumentace k LOGO!.**

<http://www1.siemens.cz/ad/current/prezentace/as/micro/logo/index.php?mi=5>

## **Další užitečné odkazy.**

Novinky a rady ověřené praxí:

<http://automatizace.hw.cz/>

PLC, automatizace, měření, regulace:

<http://www.volny.cz/blaja/>

Odborný časopis pro automatizaci:

<http://www.automa.cz/>

Odborný časopis pro automatizaci:

<http://www.automatizace.cz/>

## 3. Univerzální logický modul LOGO!

### 3.1. Základní myšlenka

Základní myšlenkou je spojit logické, spínací, časové a ostatní speciální funkce do jednoho modulu a ulehčit tak technikům práci při projektování i realizaci celé aplikace.

S jedním kompaktním modulem je pak možné realizovat všechny malé automatizační úlohy. LOGO! může ovládat čerpadla, kompresory, otvírat závory a vrata, zavlažovat rostliny ve skleníku, řídit osvětlení, nebo výtahy v budovách a podobné jednoduché úkoly. Toto zařízení je však schopné ovládat i relativně komplikovaná zařízení, včetně sledování teplot, tlaků a ostatních analogových veličin.

### Stručná charakteristika

LOGO! je obchodní název pro univerzální logický modul firmy Siemens, vyvíjený a vyráběný od roku 1996. První model měl typové číslo OBA0 a jeho paměť obsáhla 30 bloků programu. Od roku 2005 se vyrábí šestá generace s typovým označením OBA5. Program může mít 130 bloků. LOGO! je spínací a řídicí modul, používaný jako náhrada klasických reléových zapojení.

Za pomoci vhodných čidel dokáže vyhodnotit teplotu, tlak, vzdálenost, rychlost proudění atd. Vyrábí se pro různé druhy napájecího napětí i proudovou zátěž výstupů.

Obsahuje množství naprogramovaných funkcí. Modulární struktura umožňuje přizpůsobit LOGO! požadavkům aplikace.

### 3.2. Identifikátory určující model 12/24RC OBA5

12: Napájecí napětí 12V DC (stejnoseměrné napětí).

24: Napájecí napětí 24V DC.

R: Na výstupech jsou relé.

C: Modul obsahuje hodiny reálného času.

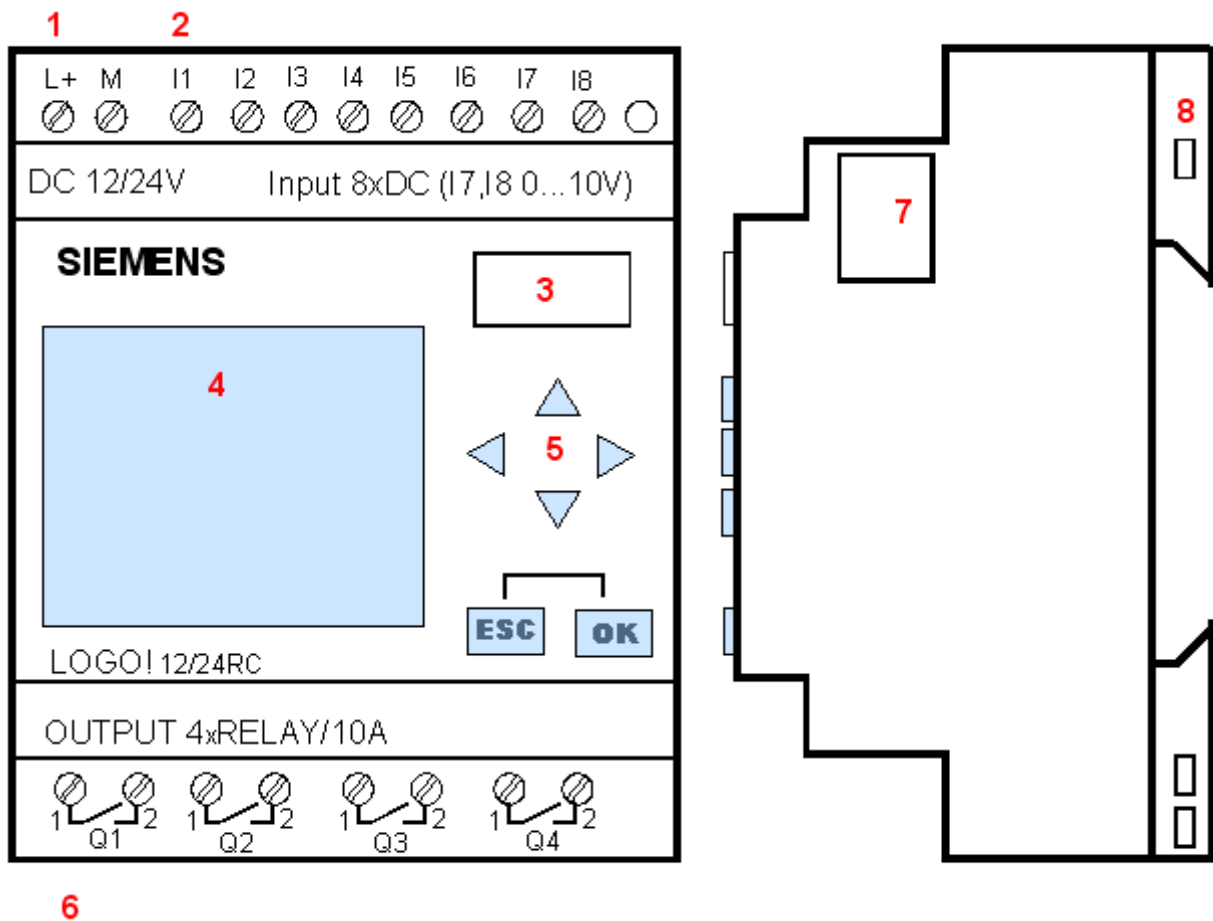
OBA5: Typové číslo modelu.

### Základní modul LOGO! Basic

- Základní modul se vyrábí s displejem a klávesnicí, (lacinější verze *Pure* je nemá).
- Má 8 vstupů a 4 výstupy.
- Šířku 72mm tj. 4 jističové jednotky.
- Hloubku 55mm.
- Je vybaven rozhraním pro připojení rozšiřující modulů.



## Obrázek a popis základního modulu LOGO! Basic



*Obr. 1. Základní modul*

1. napájení (L+, M)
2. vstupy (I1 – I8)
3. kryt objímky pro paměťový modul a propojovací šňůru PC-LOGO!
4. displej
5. klávesové pole
6. výstupy Q1 – Q4
7. kryt rozšiřujícího rozhraní pro připojení rozšiřujícího modulu
8. mechanické kódování zabráňující připojení rozšiřujícího modulu jiné napěťové třídy

### 3.3. Rozšiřující moduly

Zvyšují počet digitálních (analogových) vstupů, výstupů, nebo umožňují komunikaci. Každý modul má rozšiřující rozhraní pro připojení dalšího modulu.

Podle napájecího napětí se moduly dělí do dvou tříd:

Třída 1 do 24V: (12V DC, 24V AC / DC).

Třída 2 nad 24V: (115 až 240V AC / DC).

- DM: Digitální modul je možné připojit k základnímu modulu se stejnou třídou napětí.
- AM: Analogový modul je možné připojit k zařízení o libovolné třídě napětí.
- CM: Komunikační modul je možné připojit k zařízení o libovolné třídě napětí.

Levé rozhraní analogového, nebo komunikačního modulu je galvanicky odděleno, proto je můžeme připojit k zařízením jiné napěťové třídy, než na jaké je sám zkonstruovaný.

#### **Pořadí modulů ovlivňuje rychlost komunikace mezi nimi.**

Pro dosažení optimálního a vysokorychlostního komunikačního výkonu mezi LOGO! Basic a různými moduly doporučuje výrobce instalovat nejdříve digitální a pak analogové moduly.

Před vyjmutím a vložením rozšiřovacího modulu musíme vypnout napájení. Rozšiřující rozhraní posledního rozšiřovacího modulu musí být zakryto.

Více o rozšiřujících modulech v [1] část 2.1.

### **3.4. Napájecí zdroj a jištění zdroje**

- Vstupní napětí zdroje:  
100 až 240V AC (dovolená tolerance 85 až 264V AC), vstupní proud 0,7 až 0,85A  
Doporučené jištění vstupu zdroje: 16A char.B, 10A char.C
- Výstupní napětí zdroje:  
2x24V DC / 1,3A

Krátkodobá porucha napájení může způsobit signál pro některé funkce spouštěné hranou. Překlenutí výpadku napájení je u LOGO! 12/24RC napájené 24V DC 5ms, (12V DC 2ms). Data posledního nepřerušeného cyklu jsou uložena do LOGO! Basic.

### **Napájení a jištění modulu**

- K napájení je použito 12 nebo 24V DC, přívod je na svorky modulu L+, M.
- Doporučené jištění napájení LOGO! je 0,8A. Jistí se svorka L+.
- Modul je chráněn před přepólováním.
- 

### **3.5. Montáž**

LOGO! můžeme montovat:

- Na stěnu, pomocí dvou výsuvných šoupátek. Pro zavrtání je v dodávce k dispozici šablona.
- Na DIN lištu  $s = 35$  mm.
- Do dveří rozváděče pomocí speciálního rámečku.

### **Bezpečnostní předpisy**

Konektor paměťového modulu a rozšiřující rozhraní posledního rozšiřujícího modulu musí být při provozu chráněn krytkou.

Při montáži musíme dodržovat BP vyplývající z charakteru prováděných operací.

#### **Ochrana před úrazem elektrickým proudem:**

Modul je s dvojitou izolací, u modulů napěťové třídy 2 není nutné připojovat ochranný vodič.

**Před uvedením zařízení do provozu musíme odzkoušet všechny předvídatelné závady, ke kterým může při provozu dojít.**

## Pravidla pro připojení vodičů

Do vstupních a výstupních svorek se smí připojit maximálně:

- Jeden vodič o průřezu  $2,5\text{mm}^2$ .
- Nebo dva vodiče o průřezu max.  $1,5\text{mm}^2$ .

Svorky musíme utahovat přiměřenou silou, vhodným šroubovákem o šířce čepele 3mm, max. krouticí moment  $0,4$  až  $0,5\text{N/m}$ , nebo 3 až 4 libry/palec ☺.

Vodiče musí mít přiměřenou rezervu a musí být odlehčeny před namáháním. Delší přívody na vstupy je doporučeno provést stíněným vodičem. Délka nestíněného vedení může být maximálně 100 m.

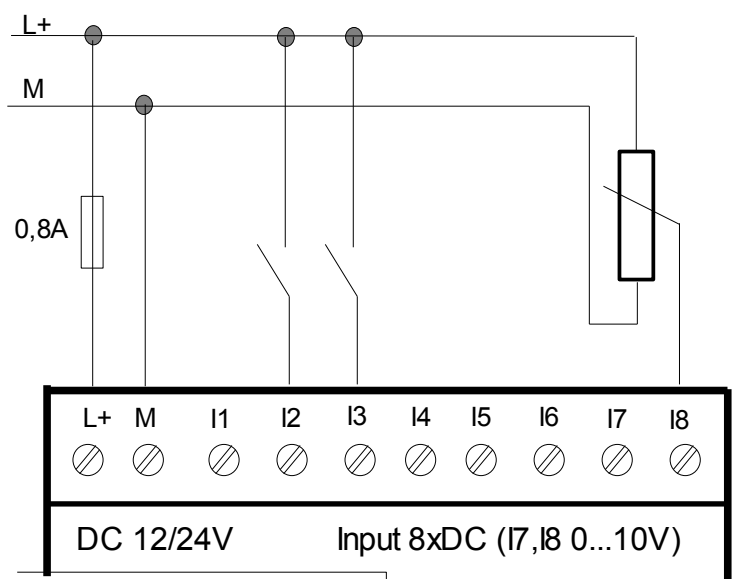
Vždycky musíme oddělit:

- Obvody AC.
- Obvody s rychlými vypínacími cykly.
- Obvody nn signálové.

Pro analogové vstupy se musí použít kroucený a stíněný kabel jen o nezbytné délce.

Pro ochranu před zasažením bleskem nutno použít vhodnou přepětíovou ochranu.

### 3.6. Připojení vstupů



*Obr. 2. Připojení vstupů do LOGO!*

#### VSTUP I - INPUT

Vstupní svorky jsou označeny I1 – I8. Jsou potenciálově závislé, jejich napětí je vztaženo ke stejnému potenciálu jako je napájení modulu.

**Na vstupy připojujeme:**

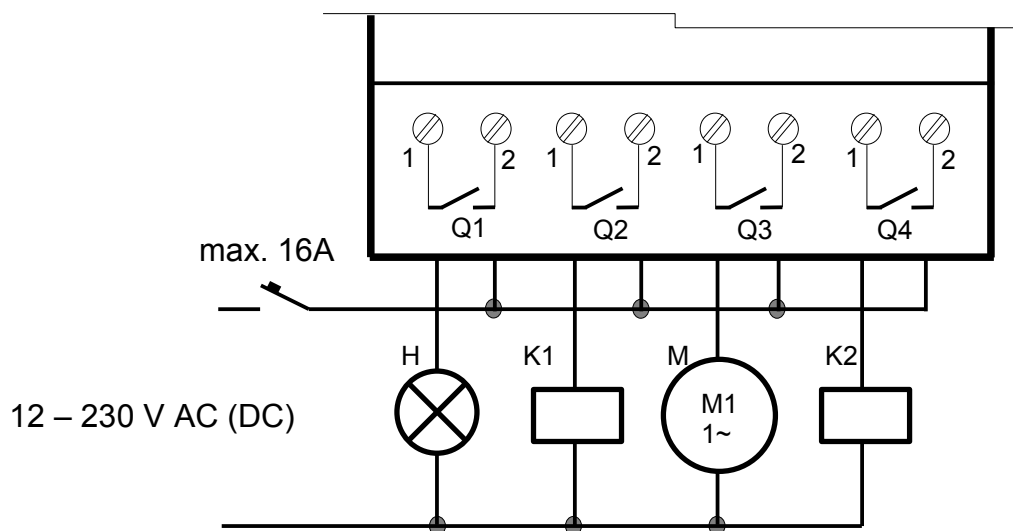
- Kontakty (př. koncové spínače, tlačítka, kontakty tepelných ochran, atd.).
- Snímače (př. snímače tlaku, teploty, vlhkosti, vzdálenosti, atd.).

### Na vstupy přivádíme:

- a) **Digitální signály.** Jestliže jsou změny signálu do 4Hz, použijeme vstupy I1-I8. V souvislosti s digitálním signálem mluvíme o logické nule a logické jedničce.
  - Stav log.0 (VYPNUTO) je napětí v rozmezí 0 až 5V DC (max. proud 1mA).
  - Stav log.1 (ZAPNUTO) je napětí v rozmezí 8 až 24V DC (min. proud 1,5mA).
- b) **Digitální signály s vysokou frekvencí.**  
K jejich zpracování použijeme pouze vstupy I5, I6.
- c) **Analogové signály.**  
Vodiče s tímto signálem připojíme na fyzické vstupy I7, I8. V programu musí být použité konektory AI1, AI2. Povolený rozsah napětí je 0 – 10V DC.

Aby LOGO! signál nebo jeho změnu zaznamenalo, musí trvat alespoň jeden programový cyklus.

### Připojení výstupů



*Obr. 3. Připojení výstupů do LOGO!*

### VÝSTUP Q - QUIT

Jako výstupy má základní modul LOGO! 12/24RC čtyři reléové kontakty, které jsou označeny písmeny Q1 – Q4. Kontakty jsou fyzicky oddělené od vstupů. Mají svoje vlastní napájení, které může být libovolné, ale maximálně 230V AC/DC.

### Maximální proudové zatížení výstupních kontaktů

$I_{\max} = 10A$  DC na jedno relé.

Kontakty výstupních relé nesmíme proudově přetížít. Není dovoleno spojovat paralelně výstupní kontakty pro zvýšení výkonu. Maximální proudové zatížení kontaktů má vliv na životnost přístroje. Pro spínání vyšších proudů použijeme stykač. Doporučené maximální jištění výstupů pokud je vyžadováno je 16A pro jistič char.B.

## Dovolená spínací frekvence výstupních kontaktů

Při ohmické zátěži: 2 Hz.

Při induktivní zátěži: 0,5 Hz.

V [1] str.143 jsou uvedeny další příklady dovoleného zatížení kontaktů.

### 3.7. Zapnutí LOGO!

Modul se zapíná připojením napětí. Při zapnutí mohou nastat dva případy.

- **Program byl natažený už před vypnutím** a tehdy se program spustí. Zobrazí se jedna ze startovacích obrazovek. Buď datum a čas, nebo obrazovka vstupů. Startovací obrazovku nastavíme<sup>1</sup> pomocí volby menu StartScreen.
- **Program v LOGO! není**, tehdy se zobrazí text „NO PROGRAM PRESS ESC”.

### Obrazovky běžícího programu

Při běhu programu je možné přepínat kurzorovými klávesami vpravo / vlevo mezi obrazovkami datum a čas, logický stav digitálních i analogových vstupů a výstupů, paměťových bitů, programovacích kláves. Volbou [StartScreen] můžeme nastavit jako výchozí obrazovku datum /čas, nebo obrazovku vstupů.

Při běhu programu je možné stisknutím klávesy ESC vstoupit do podmenu přiřazení parametrů.

## 4. Programování LOGO!

Program je schéma vyjádřené jinak. Programování je vkládání schémat do LOGO!

LOGO! můžeme programovat:

- **Ručně**

Pomocí kláves na přístroji vkládáme program od výstupu ke vstupu, "od motoru k tlačítku". Tento způsob programování je vhodný pro kratší programy, případně pro změnu parametrů běžícího programu.

Displej je nezastupitelný při ručním programování, editaci programů a změně parametru běžícího programu.

- **Softwarově**

Pomocí programu LogoSoftComfort.

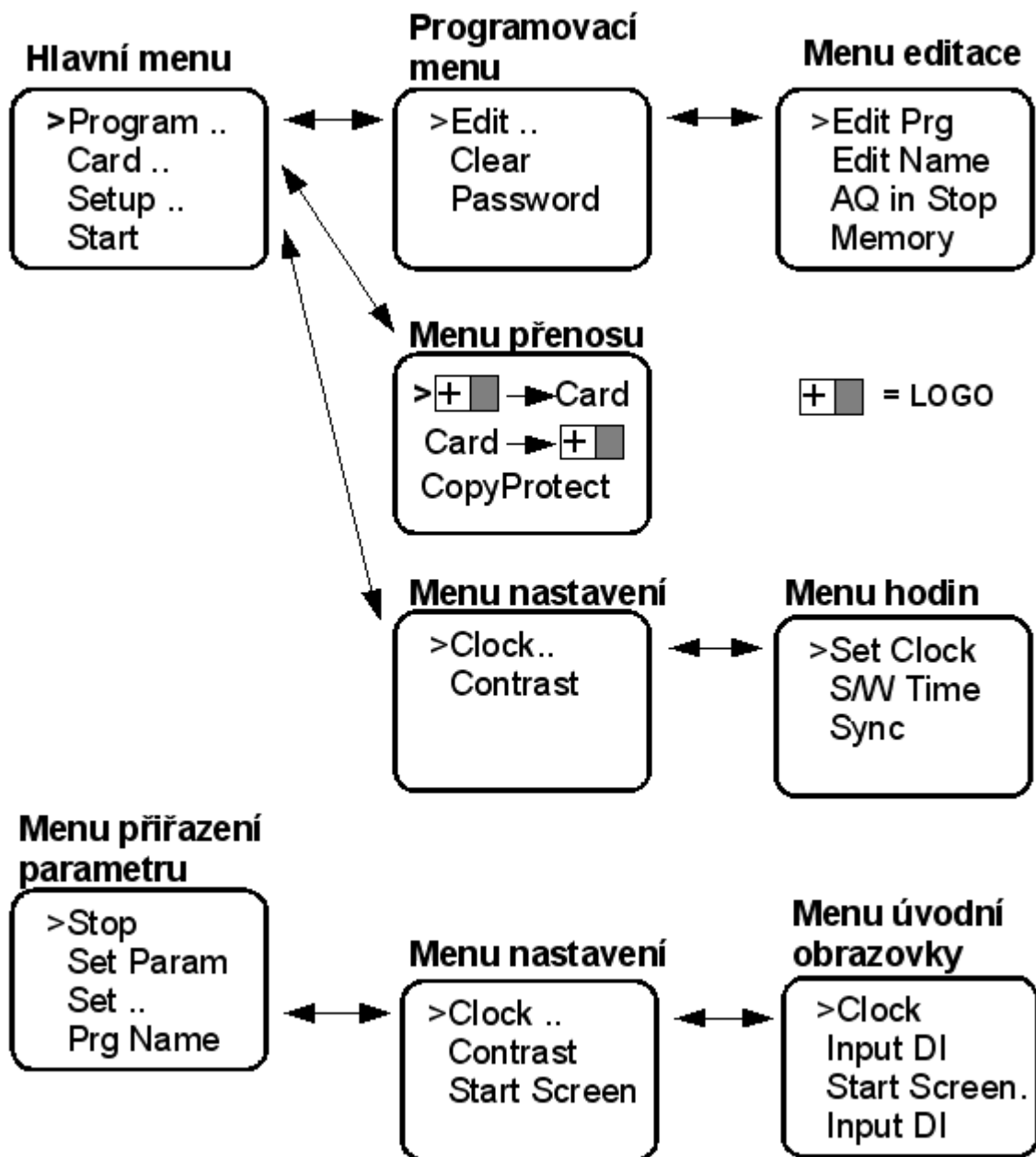
Navržený a odladěný program přeneseme z PC do LOGO! pomocí šňůrové propojky.

Přenos programu je možný i z LOGO! do PC.

---

<sup>1</sup> Program musí být v LOGO!.

## Grafické znázornění struktury menu



*Obr. 4. Menu základního modulu*

## Podrobnější popis jednotlivých úrovní menu

**Hlavní menu => podmenu 1.řádu => podmenu 2.řádu**  
 [Program] => Programovací menu => Menu editace  
 [Card... ] => Menu přenosu  
 [Setup] => Menu nastavení => Menu hodin  
 [Start ] Po spuštění programu se objeví výchozí obrazovka. Viz menu nastavení [StartScreen].

### Programovací menu.

[ Edit..] =>

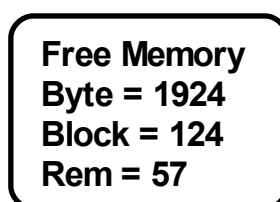
### Menu editace.

[ Edit Prg ] Tvorba nového, editování hotového programu.

[ Edit Name ] Pojmenování, editace názvu programu.

[ AQ in Stop ] Výběr analogových výstupních hodnot pro přechod ze stavu RUN do Stop.

[ Memory ] Zobrazení zbývajících místa v paměti viz obrázek 5.



**Free Memory**  
**Byte = 1924**  
**Block = 124**  
**Rem = 57**

*Obr. 5. Volná paměť*

### Poznámka k obr. 5

**Byte:** Celková velikost paměti vyhrazené pro bloky programu je 2000 Bytů (v ukázce je využito 76 Byte, volných je 1924).

*Paměťové nároky bloků jsou různé.*

*Nejméně požaduje blok NOT = 4 Byte, XOR 8 Byte, ostatní GF bloky 12 Byte.*

*Výběr z bloků SF: Blok zpožděné zapnutí a RS 8 Byte, zpožděné vypnutí 12 Byte, čítač 24 Byte, regulátor 40 Byte.*


**Block:** Maximální počet bloků použitých v programu je 130. Zde použito 6, volných je 124.


**Rem:** Při povolené remanenci jednoho bloku, je obsazena velikost 3 Byte. Celková velikost paměti vyhrazené pro remanenci bloků je 60 Byte. V ukázce jsou použity 3 Byte, protože volných je ještě 57 Byte.

[ Clear prg ] Po potvrzení otázky dojde k vymazání programu.

[ Password ] Zadání, změna, nebo zrušení hesla.

### Menu přenosu.

[  => Card ] Kopírování z LOGO! do paměťové karty.

[ Card =>  ] Kopírování z paměťové karty do LOGO!.

[ CopyProtect ] Nastavení ochrany proti kopírování programu.

**Menu nastavení.**

[ Clock.. ] (hodiny) =>

**Menu hodin.**

[ Set Clock ] Seřízení hodin.

[ S/W Time ] Letní /zimní čas.

[ Sync] Povolit/zakázat synchronizaci časů mezi LOGO! Basic a komunikačním modulem.

Poznámka k synchronizaci:

Při použití základního modulu LOGO! společně s digitálními, nebo analogovými rozšiřovacími moduly bez komunikačního modulu (od verze 0AA1 výše) EIB/KNX, **nesmíme** aktivovat časovou synchronizaci! Časová synchronizace musí být deaktivována („Sync“ nastaveno na „Off“).

Více o synchronizaci v [1] str. 59.

[ Contrast ] Nastavení podsvícení (kontrastu) displeje.

**Menu přiřazení parametru.** Otevřeme stisknutím klávesy ESC když je LOGO! v RUN.

[ Stop ] Po potvrzení volby je možné zastavit běh programu.

[ Set Param ] Po potvrzení volby je možné nastavit parametr bloku.

[ Set ..]

=> **Menu nastavení.**

[ Clock .. ] => Nastavení hodin. Viz výše.

[ Contrast] Změna kontrastu displeje. Viz výše.

[ StartScreen ] => **Menu úvodní obrazovky.**

Clock

Input DI

StartScreen

Input DI

[ Prg Name ] Na této úrovni můžeme přečíst jméno programu, ale ne ho změnit.



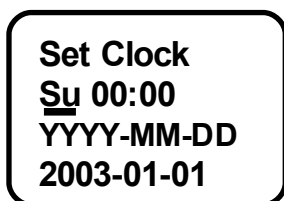
## Kurzor, jeho pohyb a tvar

Kurzor je grafická značka, označující aktivní místo na displeji. Jeho pohyb ovládáme čtyřmi kurzorovými klávesami.

Tvar kurzoru se mění podle možností daného místa programu.

- Má - li kurzor tvar  $\>$  *větší než*, můžeme provádět operace:  
Kurzorovými šipkami nahoru / dolů můžeme volit položky v menu.  
Pomocí klávesy OK můžeme potvrdit volbu.  
Pomocí klávesy ESC se můžeme vrátit o krok zpět.
- Má - li kurzor tvar:    *podtržítka*, můžeme provádět operace:  
Kurzorovými šipkami vlevo / vpravo / nahoru / dolů můžeme pohybovat kurzorem v daném směru.  
Pomocí klávesy OK můžeme ve zvoleném místě přejít na výběr konektoru / bloku..  
Pomocí klávesy ESC můžeme opustit programování.
- Má - li kurzor tvar:  *plného obdélníku*, můžeme provádět operace:  
Kurzorovými šipkami nahoru / dolů můžeme z příslušného seznamu volit konektor, blok, písmeno.  
Pomocí klávesy OK můžeme potvrdit volbu.  
Pomocí klávesy ESC se můžeme vrátit o krok zpět.

## Nastavení reálného času a data



Datum je ve tvaru rok, měsíc, den.  
YYYY ... rok.  
MM ... měsíc.  
DD ... den.

*Obr. 6. Kurzor na pozici dne v týdnu*

Čas a datum můžeme nastavit dvěma způsoby.

### 1. V programovacím režimu.

- Přejdeme do hlavního menu.  
Je-li program v RUN, nejprve ho pomocí nabídky menu [STOP] zastavíme.
- Vybereme nabídku [Setup].
- Vybereme nabídku [Clock...].
- Vybereme nabídku [Set Clock].
- Zvolíme den v týdnu (kurzorovými klávesami nahoru / dolů).  
Posuneme kurzor na další pozici (kurzorovými klávesami vpravo / vlevo).
- Postupně změníme hodnoty čísel (kurzorovými klávesami nahoru / dolů).
- Nastavíme správné datum (postup je stejný jako při nastavení času).
- Potvrdíme zadané údaje klávesou OK.

## 2. V režimu přiřazení parametrů

- Stiskneme klávesu ESC.
- Vybereme položku [Set ...] v nabídnutém podmenu.
- Vybereme nabídku [Clock...].
- Vybereme nabídku [Set Clock].
- Ostatní postup je stejný jako při nastavení v programovacím režimu.

den	zkratky	název	přibližná výslovnost
pondělí	Mo (M)	Monday	<i>mandej</i>
úterý	Tu (T)	Tuesday	<i>tjúzdej</i>
středa	We (W)	Wednesday	<i>wenzdej</i>
čtvrtek	Th (T)	Thursday	<i>thezdej</i>
pátek	Fr (F)	Friday	<i>frajdej</i>
sobota	Sa (S)	Saturday	<i>satrdej</i>
neděle	Su (S)	Sunday	<i>sandej</i>

**Tab. 1.** Dny v týdnu

## 4.1. Čtyři seznamy prvků použitelných při programování

Při programování máme k dispozici různé prvky, které jsou pro snazší orientaci rozdělené do čtyř seznamů.

- Co: seznam konektorů.
- GF: seznam bloků základních funkcí.
- SF: seznam bloků speciálních funkcí.
- Bn: seznam čísel použitých bloků.

Seznamy vždy ukazují bloky a konektory, které je možné v daném místě programu použít. Za normálních podmínek to jsou všechny konektory, GF a SF bloky, které LOGO! zná. Jestliže tomu tak není, žádaný blok by potřeboval více paměťových jednotek, než je momentálně k dispozici<sup>2</sup>.

Blok je výrobcem naprogramovaná funkce převádějící přesně definovaným způsobem informaci ze vstupu na výstup. V klasických rozváděcích se převod vstupních informací na výstupy děje pomocí propojení příslušných jednotlivých součástek. V LOGO! se totéž děje bez drátů a fyzických přístrojů.

### Seznam Co

Connector, konektor

Výraz konektor je používán pro vstupy, výstupy, příznaky a pevné hodnoty napětí (konstanty).

Kompletní seznam konektorů obsahuje:

- 24 vstupů I1 – I24.
- 16 výstupů Q1 – Q16.
- 16 prázdných (virtuálních) výstupů. X1 – X16.
- Osm analogových vstupů AI1 – AI8.

---

<sup>2</sup> Maximální počet bloků použitých v programu je 130.

- 24 digitálních příznaků M1 – M24.
- Šest analogových příznaků AM1 - AM6.
- Kurzorové klávesy. (Kurzorové kl. modulu mohou být použité k ovládání programu).
- Dva analogové výstupy AQ1 a AQ2.  
Úrovně napětí hi, lo. (Signál 1, signál 0).  
Konektor x.
- Bity posuvného registru S1 – S8.

Pokud nelze pro vyřešení rekurze použít konektor Q, použijeme příznak s přihlédnutím na vlastnost příznaku M8 v prvním programovém cyklu.

- M8 Iniciační paměťový bit.

Příznak M8 je nastaven v prvním cyklu programu na 1, můžeme ho proto použít jako spouštěcí (iniciační) bit. Po provedení prvního cyklu programu je M8 automaticky resetován a ve všech následujících cyklech se může použít jako každý jiný příznak.

## Seznam GF

General funkcion, základní funkce.

Obsahuje 8 funkcí (bloků) pracujících na základě Booleovy algebry logiky.

## Seznam SF

Special funkcion, speciální funkce.

Obsahuje 29 funkcí (bloků) naprogramovaných a použitelných v programu LOGO! podobně jako jsou použité klasické přístroje v reléové technice.

## Seznam Bn

Block number, bloková čísla.

Je seznam čísel v průběhu programování použitých a dokončených funkcí (bloků), jejichž výstupy můžeme na daném místě programu použít. Seznam Bn se tvoří postupně, tak jak LOGO! přiřazuje použitým blokům pořadová čísla.

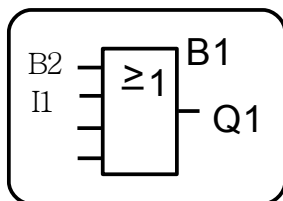
### Přiřazení čísla bloku

Blok vložený do programu je ihned automaticky očíslován jedinečným číslem.

Čísla bloku slouží pro zobrazení odkazů propojení bloků – viz obrázek 7.

Bloková čísla umožňují orientaci v programu, protože v jednom okamžiku je na displeji modulu vidět jen jeden blok. Můžeme je také použít i při popisování funkce programu.

### Zobrazení bloku na displeji



- B1 – číslo bloku, který je právě zobrazený na displeji
- B2 – číslo bloku, jehož výstup je spojený s prvním vstupem B1
- I1 – číslo konektoru vstupu připojeného na druhý vstup B1
- Q1 – číslo konektoru, který ovládá fyzický výstup Q1

**Obr. 7.** Zobrazení bloku

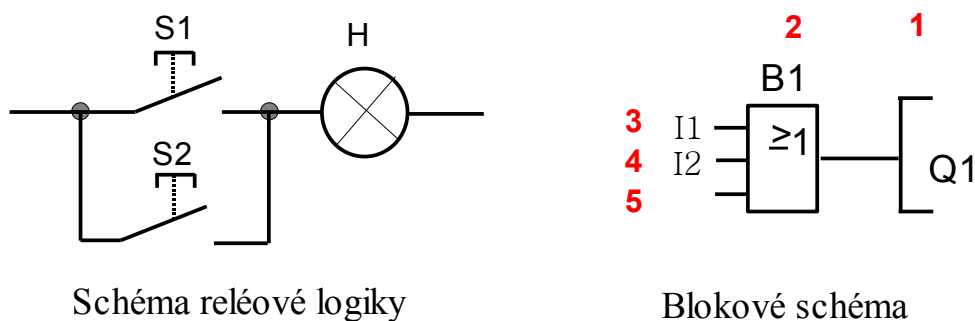
*K obr.7*

*Protože nevíme jakou funkci má B2 můžeme s určitostí tvrdit jen to, že signál na I1 sepne Q1.*

## 4.2. Postup při ručním programování

### Příklad 1. Úvodní

Uvedené schéma reléové logiky je převedeno na blokové a krok za krokem ho naprogramujeme ručně do LOGO!

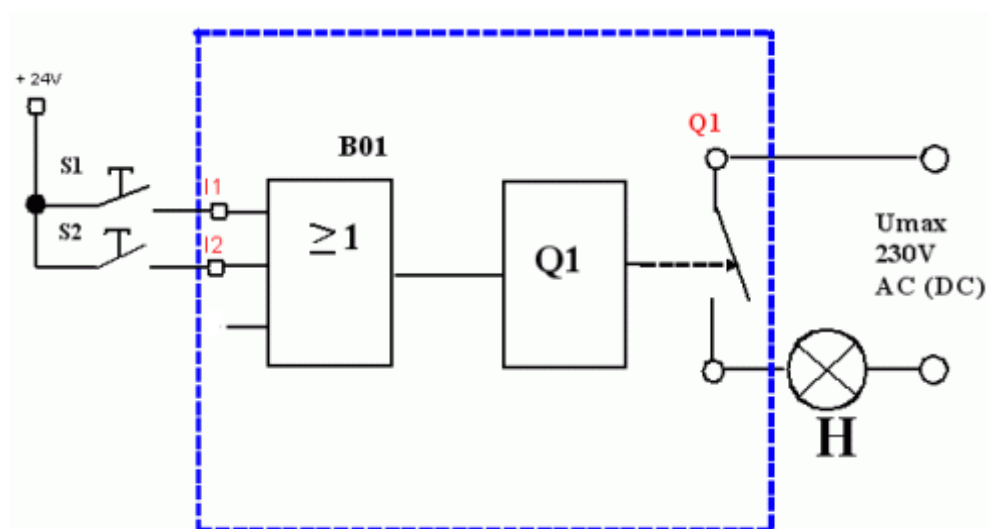


Obr. 8. K příkladu 1

Jednotlivé kroky programování jsou označené u blokového schéma červenými číslicemi. **Blok B1 je v praxi čtyřvstupový. Pokud to nebude nutné, budou pro lepší čitelnost nadále všechny čtyřvstupové bloky zobrazovány jako třívstupové.**

### Vnější spoje modulu LOGO! a blokové schéma programu

Spínače S1, S2 jsou připojené na příslušné vstupy LOGO!. Při stisknutí spínačů je na vstupy I1, I2 přiveden signál 1. Přes výstupní kontakt Q1 je připojena signální žárovka, která má svoje vlastní napájení.



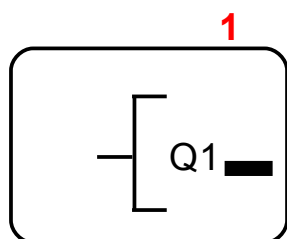
Obr. 9. Vnější spoje a blokové schéma

I1, I2 a Q1 jsou vstupní a výstupní svorky modulu a zároveň i konektory v programu.

**Výchozí situace:** LOGO! je bez programu, datum a čas jsou nastaveny.  
Po zapnutí modulu se zobrazí na displeji text : NO PROGRAM PRESS ESC.

**Postup:**

- Stiskneme klávesu ESC.  
Zobrazí se hl. menu, [Program], [PC/Card], [Clock..], [Start].
- Zvolíme nabídku [Program] a potvrdíme volbu klávesou OK.  
Zobrazí se programovací menu [Edit Prg], [Prg Name], [Clear Prg], [Password].
- Zvolíme nabídku [Edit Prg] a potvrdíme volbu klávesou OK.  
Zobrazí se obrazovka obr. 10.



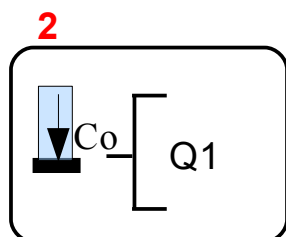
*Obr. 10. První krok*

Kurzor je ve tvaru podtržítka.

Jako první se nabídne k použití konektor Q1.

Pomocí kurzorové klávesy nahoru / dolů je možné zvolit i jiný výstup.

- Potvrdíme nabídku konektoru Q1 klávesou OK a kurzor skočí na pozici druhého kroku, obr. 11.

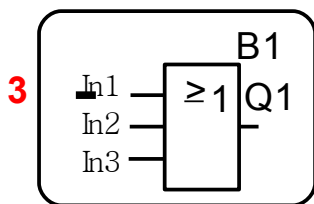


*Obr. 11. Druhý krok*

**2. krok, obr. 11**

Tvar kurzoru se změní na blikající obdélník a zobrazí se seznam Co. Kurzorovou klávesou nahoru / dolů můžeme listovat a volit mezi seznamy Co, GF, SF, které jsou v tomto okamžiku k dispozici. Seznam Bn není přístupný, protože ještě není žádný blok pro použití v programu připravený.

- Vybereme seznam GF a potvrdíme volbu klávesou OK.  
Nabídne se 1. blok v seznamu GF, blok AND.
- Kurzorovou klávesou nahoru / dolů listujeme v seznamu GF.
- Vybereme blok OR a potvrdíme volbu klávesou OK.  
Kurzor skočí na pozici třetího kroku a bloku OR je přiřazeno číslo B1, obr. 12.

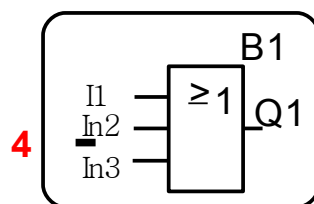


*Obr. 12. Třetí krok*

Kurzor je ve tvaru podtržítka na prvním vstupu bloku OR.

Pohyb kurzoru mezi jednotlivými vstupy je možný kurzorovými klávesami nahoru / dolů.

- Klávesou OK potvrdíme volbu umístění kurzoru na prvním vstupu. Kurzor se změní na plný obdélník a nabídne se seznam Co. Kurzorovými klávesami nahoru / dolů je možné listovat mezi seznamy Co, GF, SF, (Bn stále není k dispozici).
- Potvrdíme nabídnutý seznam Co klávesou OK. Nabídne se první položka v seznamu Co, konektor I1. Kurzorovými klávesami nahoru / dolů je možné listovat v seznamu konektorů.
- Potvrdíme volbu konektoru I1 klávesou OK. Kurzor skočí na pozici čtvrtého kroku, obr.13.

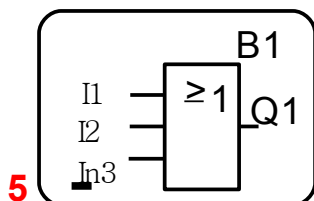


*Obr. 13. Čtvrtý krok*

#### 4. krok, obr. 13

Kurzor je ve tvaru podtržítka u druhého vstupu bloku OR.

- Klávesou OK potvrdíme nabízený vstup.
- Postupem dle kroku 3 zadáme na vstup I2 a potvrdíme volbu klávesou OK. Kurzor skočí na pozici pátého kroku, obr. 14.



*Obr. 14. Pátý krok*

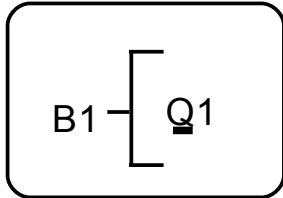
#### 5. krok, obr. 14

Kurzor je ve tvaru podtržítka u třetího vstupu bloku OR.

- Program nyní můžeme spustit, nebo pokračovat v další činnosti z pozice kurzoru pod konektorem Q1, obr.15.

### Spuštění programu.

Třikrát stiskneme klávesu ESC a tak přejdeme postupně do hlavního menu. Tam vybereme a potvrdíme položku [START].



Obr. 15. Další možnosti

### Jaké možnosti máme, je-li kurzor v pozici na obr. 15?

- **Můžeme pokračovat v programování.**  
V poloze označené na obr. 15 vybereme kurzorovými klávesami další konektor a programujeme další část (další linii) programu.
- **Můžeme zkontrolovat hotový program.**  
Pomocí kurzorových kláves nahoru / dolů / vpravo / vlevo se budeme pohybovat v programu příslušným směrem a kontrolujeme umístění a druh jednotlivých bloků.
- **Můžeme opravit program.**  
Posuneme kurzor na místo které chceme opravit.  
Zvolené místo potvrdíme klávesou OK.  
Zadáme jiný blok, nebo konektor a potvrdíme novou volbu klávesou OK.

### Poznámky k editaci programu:

- Zrušení celé větve programu.  
Pokud je potřebujeme zrušit část programu, který je napojen na určitý vstup, na tento vstup bloku stačí zadat konektor x (vstup neobsazen). Všechny bloky i konektory od tohoto místa doleva tím zrušíme a můžeme je zadat znova.
- Pořadí vstupů GF bloků.  
Pořadí vstupů u GF bloků je libovolné, na funkci bloku nemá pořadí vstupů vliv.
- Neobsazený vstup.  
Když je nějaký vstup bloku úmyslně neobsazený, program mu automaticky přiřadí logický stav zajišťující správnou funkci bloku. Při ručním programování můžeme nepovinně tento vstup označit konektorem x.

## Příklad 2. Použití seznamu Bn

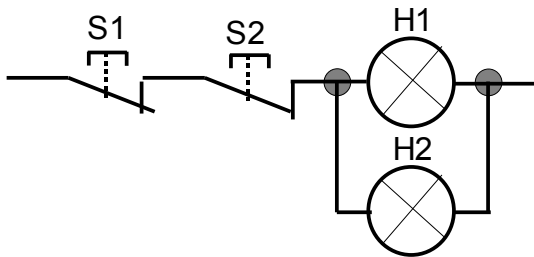
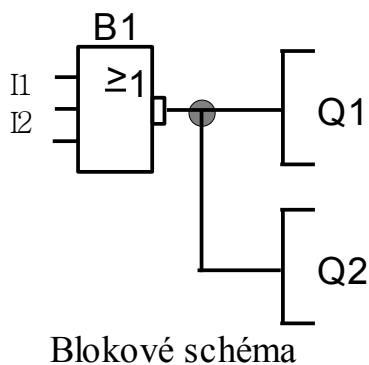
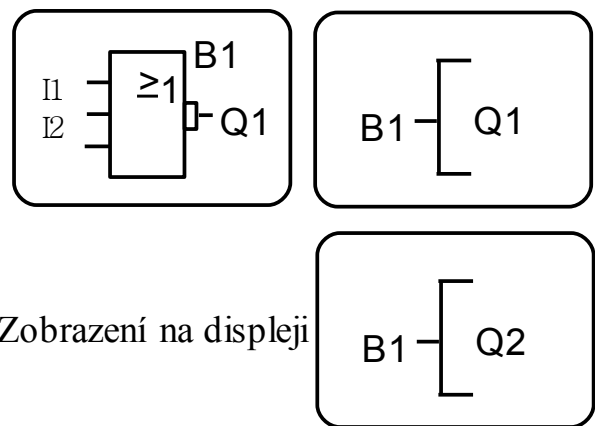


Schéma reléové logiky



Blokové schéma

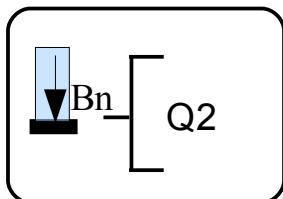


Zobrazení na displeji

Obr.16. K příkladu 2

### Postup:

- Podle postupu uvedeném v příkladu 1 naprogramujeme první řádku programu (linii Q1) a po zadání všech konektorů na vstupy B1 vrátíme kurzor pod Q1.
- Zde nalistujeme konektor Q2 a potvrdíme volbu klávesou OK.
- Začínáme programovat druhou řádku programu, linii Q2.**
- Z nabídky seznamů Co, GF, SF, Bn vybereme seznam Bn (obr 17) a potvrdíme volbu klávesou OK.



Obr. 17. Seznam Bn

- V seznamu Bn je v tomto okamžiku k dispozici pouze B1. Potvrdíme nabídku B1 klávesou OK a tím jsme napojili konektor Q2 na výstup bloku B1. Program je kompletní.
- Stiskneme ESC tolikrát až se dostaneme do hlavního menu, potvrdíme [START] klávesou OK a program běží.



**Příklad 3.** Konektory Q1, Q2 zapojené v sérii

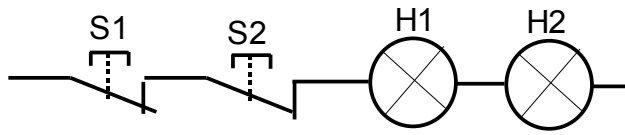
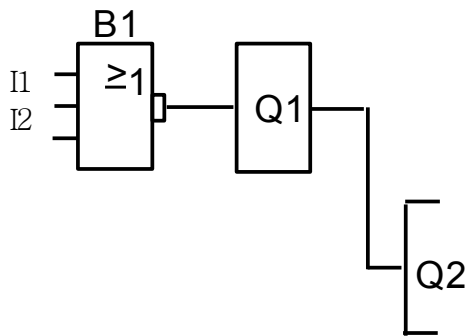
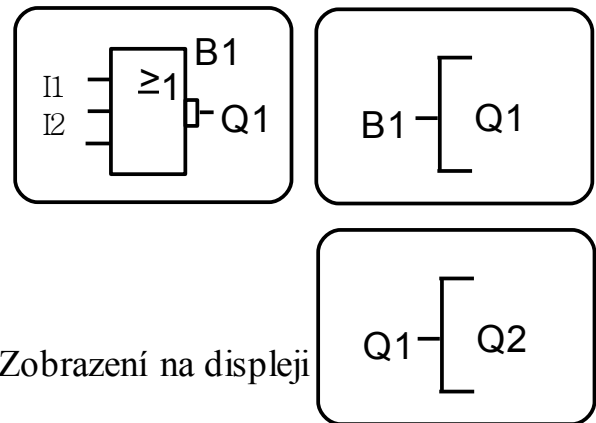


Schéma reléové logiky



Blokové schéma



Zobrazení na displeji

*Obr. 18. K příkladu 3*

**Postup:**

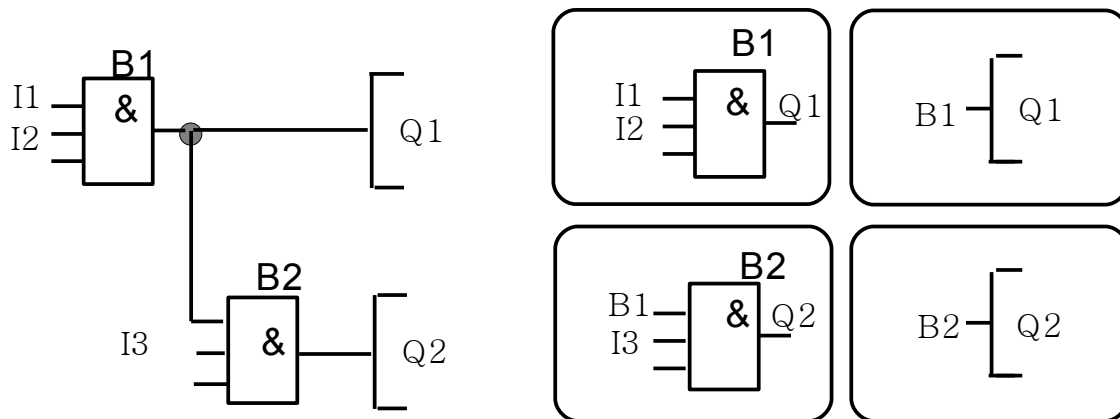
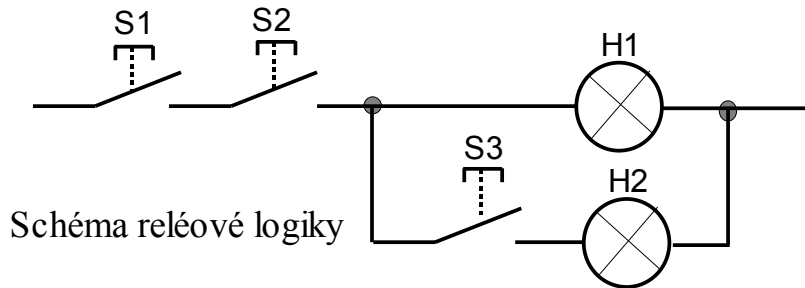
Programujeme stejně jako v příkladu 1 až do bodu C.

- V bodě D vybereme a potvrdíme Co.
- Vybereme a potvrdíme konektor Q1 a program je kompletní.

### Příklad 4.

Jestliže I1 a I2 = 1 potom Q1 = 1

Jestliže I1 a I2 a I3 = 1 potom Q1 = 1 a Q2 = 1



Blokové schéma

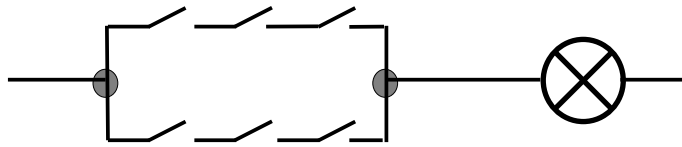
Náhled jednotlivých  
obrazovek LOGO!

*Obr. 19. K příkladu 4*

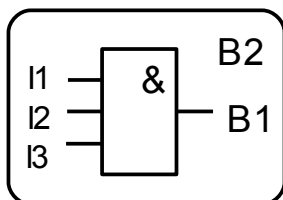
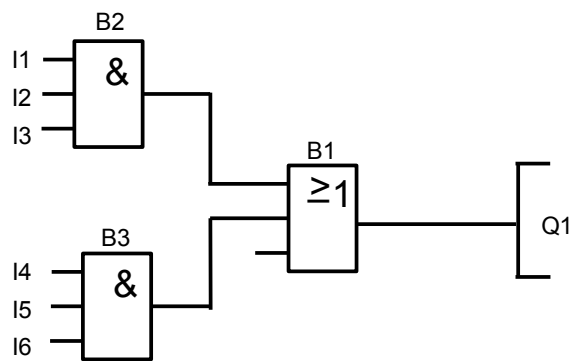
## Příklad 5.

Pokud  $I_1$  a  $I_2$  a  $I_3 = 1$  nebo  $I_4$  a  $I_5$  a  $I_6 = 1$  potom  $Q_1 = 1$

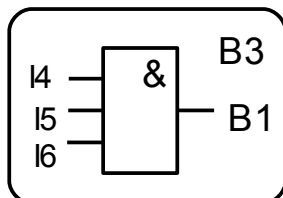
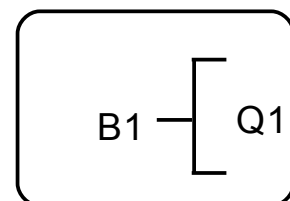
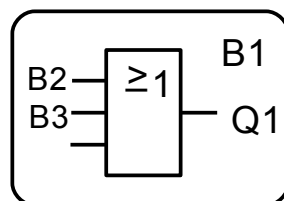
Reléové schéma



Blokové schéma



Jednotlivé obrazovky při kontrole programu.



Obr. 20. K příkladu 5

## Invertování vstupu při ručním programování

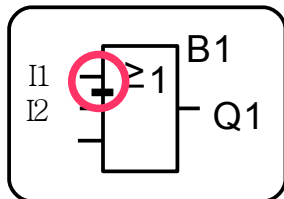
Jednotlivé vstupy základních i speciálních funkcí můžeme invertovat.

Pokud na takto upravený vstup přivádíme signál 1, do bloku půjde 0 a naopak. 0 na vstupu je invertována na 1.

### Postup invertování vstupu.

Chceme-li vstup bloku invertovat, musí už být vstupu obsazený. Buď konektorem jako na obr.21, nebo výstupem z jiného bloku.

Posuneme potom kurzor pomocí kurzorových kláves na příslušné místo bloku, viz obr. 21.



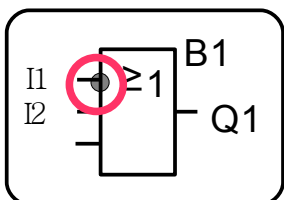
**Obr. 21.** Určení vstupu

Potvrdíme zvolené místo klávesou OK.

Pro inverzi tohoto vstupu stiskneme kurzorovou klávesu nahoru / dolů. Je jedno který směr zvolíme, výsledek bude v obou případech stejný.

Změnu (obr. 22) potvrdíme klávesou OK.

*Symbol negování vstupu není na displeji příliš zřetelný.*



**Obr. 22.** Negování vstupu

### Zrušení invertovaného vstupu

Posuneme kurzor pomocí kurzorových kláves na symbol negování vstupu.

Potvrdíme zvolené místo klávesou OK.

Pro zrušení inverze stiskneme kurzorovou klávesu nahoru / dolů.

Změnu potvrdíme klávesou OK.

## 4.3. Pracovní režimy LOGO!

### Programovací režim

Položka menu [Edit Prg]

Použije se pro vkládání nového programu do modulu, nebo pro úpravu programu stávajícího. Postup při ručním programování (editování) je popsán v části 4.2. (4.4.) této příručky.

### Režim RUN<sup>3</sup>

Položka menu [Start]

#### Program běží a pracuje.

Na displeji je podle nastavení [StartScreen] zobrazený datum, čas, nebo obrazovka vstupů. Jako výchozí je nastavena obrazovka datum a čas.

Je-li číslo vstupu / výstupu zobrazeno inverzně, je ve stavu 1.

Takto můžeme kontrolovat stavy vstupů a výstupů, aniž by kontakty modulu spínaly zátěž.

Kurzorovými šipkami vpravo / vlevo můžeme přepínat mezi dalšími obrazovkami.

#### LOGO! v RUN:

- 1) Program běhá ve smyčce<sup>4</sup> v [1] příloze B je uveden způsob měření délky smyčky.
- 2) Načítá stavy vstupů.
- 3) Vypočítává podle zadaného programu stav výstupů.
- 4) Spíná nebo vypíná výstupy.

Stisknutím klávesy ESC vyvoláme menu přiřazení parametrů a můžeme volit ze čtyř nabídek.

- [ Stop ] můžeme zastavit program.
- [ Set Param ] můžeme sledovat a měnit parametry programu, viz dál.
- [ Set .. ] v dalším podmenu můžeme nastavit čas a datum.
- [ Prg Name ] můžeme si přečíst jméno programu.

### Režim přiřazení parametrů

Položka menu [ Set Param ]

Režim je přístupný jen v RUN, stisknutím klávesy ESC. Můžeme potom prohlížet a měnit parametry určitých bloků, aniž bychom přitom měnili program. Změna parametru se ihned provádí. Tento režim využijeme při ladění programu nebo poučená obsluha může měnit nastavení určitých hodnot.

#### Podmínky pro spuštění režimu přiřazení parametrů

- Program obsahuje blok(y) s parametrem.
- Parametrický režim není zakázaný. Pro každý blok se zakazuje tento režim samostatně.

Když nejsou splněny předchozí podmínky, objeví se po volbě [Set Param] hlášení: NO PARAM PRESS ESC. Stisknutím klávesy ESC opustíme parametrický režim.

---

<sup>3</sup> Vysl. ran

<sup>4</sup> Délka programového cyklu <10ms.

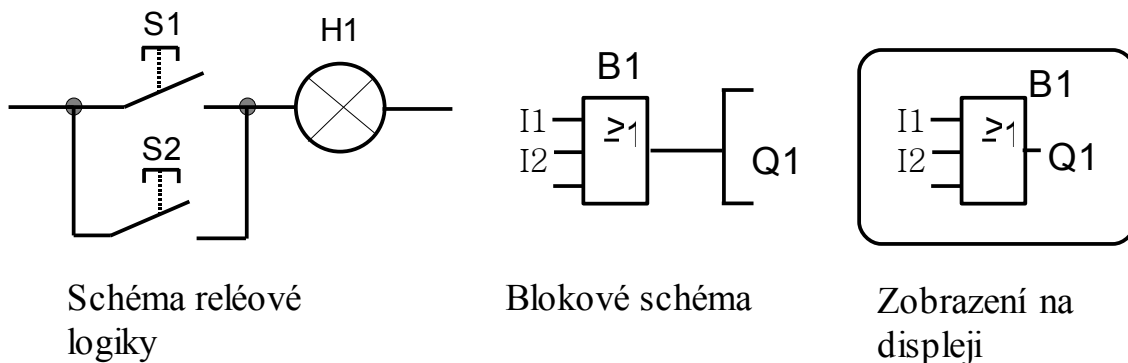
#### 4.4. Editace hotového programu

- **V této části budeme provádět:**
- vložení nového bloku do hotového programu (v bodech 1 - 5).
- nastavení parametru bloku při programování (body 6 – 9).
- přepnutí do režimu přiřazení parametrů za běhu programu (body 10 a 11).
- změnu parametru za běhu programu (bod 12) .
- potlačení režimu přiřazení parametrů (bod 13).

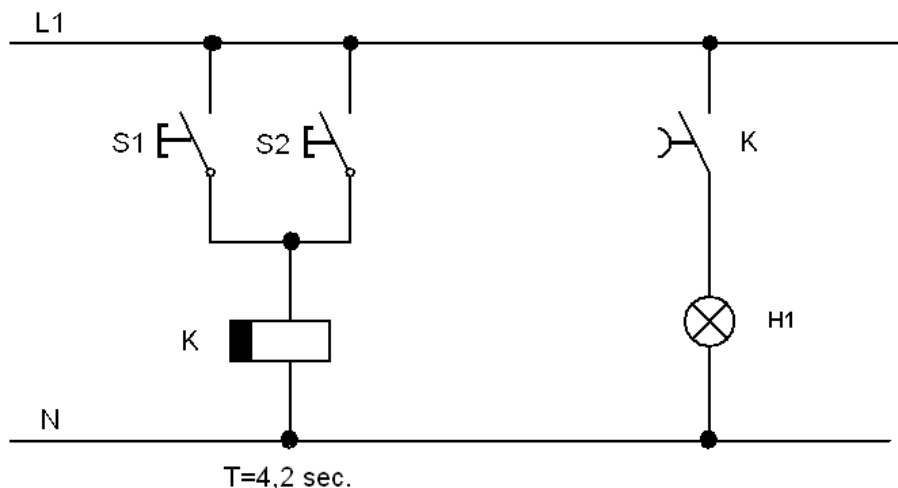
##### Výchozí podmínky:

V LOGO! je zavedený program podle blokového schéma na obr. 23.

Program upravíme v editačním režimu tak, aby odpovídal schématům na obr. 24 a 25.



*Obr. 23. Původní program*



*Obr. 24. Reléové schéma*

##### Popis funkce obvodu podle obr. 24.

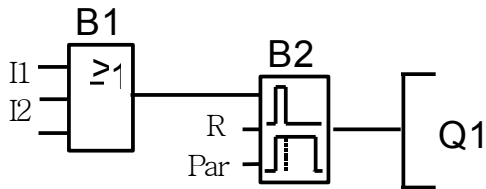
S1 nebo S2 zapínají relé se zpožděním při odpadu K. Relé sepne svým kontaktem žárovku H1.

Po přerušení napájení relé, zůstane kontakt K sepnutý ještě 4,2 sekundy.

Když do vypršení nastavené doby nestiskneme zapínací tlačítko, kontakt relé odpadne a žárovka zhasne.

## Vložení nového bloku do stávajícího programu

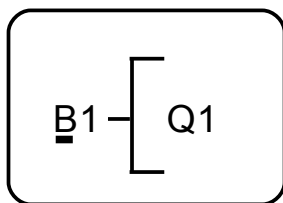
Po skončení úprav bude program vypadat následovně.



*Obr. 25. Upravený program*

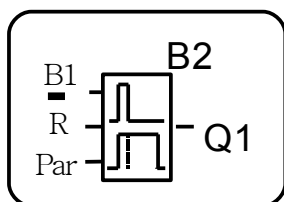
### Postup:

1. Přepneme LOGO! do editovacího režimu.
2. Obr. 26. Přesuneme kurzorovou klávesou kurzor pod písmeno B a klávesou OK potvrdíme toto místo. Ve zvoleném místě začínáme editovat program.



*Obr. 26. Začátek editace*

3. Z nabídky seznamů vybereme seznam SF a potvrdíme volbu klávesou OK.
4. V seznamu SF vybereme blok zpoždění při odpadu a potvrdíme volbu klávesou OK.  
(Blok má tři vstupy označené Trg, R a T. Než potvrdíme výběr bloku klávesou OK je u prvního vstupu popis Trg.)  
Vložený blok dostane číslo B2.  
Blok B1, až doteď připojený na Q1, je automaticky přepojený na horní vstup B2. Kurzor je zobrazený na horním vstupu nového bloku obr. 27.



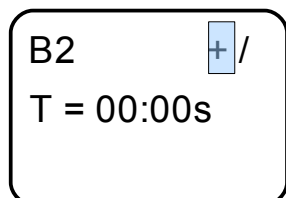
*Obr. 27. Nový blok*

5. Vstup R u B2 nebude v tomto programu použitý.

## Nastavení parametru bloku při programování

6. Přesuneme pomocí kurzorových kláves kurzor na parametrový vstup T bloku B2 a stiskneme klávesu OK. Na displeji přístroje se zobrazí formulář pro nastavení parametrů B2 (obr. 28).

### Formulář nastavení parametrů bloku.



B2 číslo bloku zpožděné vypnutí  
+ režim přiřazení parametrů přístupný  
/ remanence bloku B2 je zakázaná  
T parametr bloku je čas, viz tab. 2  
s časová základna nastavená na sekundy. Nastavení času v závislosti na časové základně v tab. 2

Obr. 28. Parametry bloku

7. Kurzor je ve tvaru plného obdélníku.

Kurzorovou klávesou přesuneme kurzor na místo jednotek sekund.

Zadáme hodnotu sekundy. Listování mezi číslicemi provádíme kurzorovými klávesami nahoru a dolů. Nastavení jednotlivých hodnot parametrů se nepotvrzuje.

Pomocí kurzorové klávesy vpravo přejdeme na pozici desetin sekundy a obdobným postupem zadáme potřebnou hodnotu.

8. Po skončení zadávání všech údajů se celkové nastavení parametru potvrdí klávesou OK.

9. Dvojnásobným stisknutím klávesy ESC postupně přejdeme do hlavního menu a upravený program můžeme spustit povelom [Start].

## Přepnutí do režimu přiřazení parametrů za běhu programu

### 10. Přepnutí do parametrického režimu.

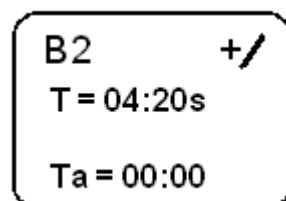
Program běží, ale ještě nebyl na vstup Trg přivedený signál.

Stisknutím klávesy ESC vyvoláme na displeji menu přiřazení parametrů.

Zvolíme [Set Param] a potvrdíme volbu klávesou OK..

Na displeji se zobrazí formulář, viz obr. 29.

Je vidět čas nastavený, T = 04:20s a čas aktuální Ta. V tomto okamžiku Ta = 00:00s.



Obr. 29. Parametry bloku

11. Po přivedení signálu 1 na vstup Trg bloku B2 sepne Q1. Čas zpožděného vypnutí neběží, aktuální čas Ta = 00:00s. Jakmile se změní signál na Trg na 0, začne blok odpočítávat čas zpožděného vypnutí a hodnota Ta se mění. Když trvá 0 na Trg dostatečně dlouho, tak po uplynutí nastaveného času se B2 vypne.



## Změna parametru bloku za běhu programu

### 12. Změna parametru bloku za běhu programu.

V režimu přiřazení parametrů stiskneme klávesu OK.

Objeví se kurzor ve tvaru plného čtverce.

Pomocí kurzorových kláves přesuneme kurzor na místo jehož hodnotu chceme změnit a postupem podle bodu 6. změním hodnotu parametru.

Konec změny potvrdíme stisknutím klávesy OK. Nové nastavení se projeví ihned.

## Potlačení režimu přiřazení parametrů

### 13. Potlačení parametrizace.

V některých případech je nežádoucí, aby obsluha zařízení měnila parametry programu.

V programovacím režimu nastavíme u příslušného bloku symbol + na - . Obr. 31.

V tomto případě se po pokusu o vyvolání režimu nastavení parametrů [Set Param] zobrazí hlášení NO PARAM PRESS ESC.

## Časová základna

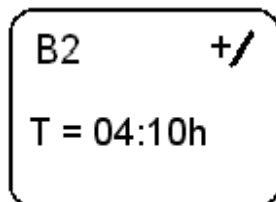
Zadání časových údajů je závislé na časové základně, k níž se tyto údaje vztahují.

Čas v závislosti na časové základně			
časová základna	- -	:	- -
s [sekundy]	sekundy	:	1/100 sekundy <sup>5</sup>
m [minuty]	minuty	:	Sekundy <sup>6</sup>
h [hodiny]	hodiny	:	minuty **

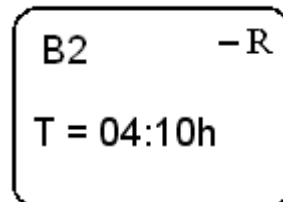
Tab. 2. Čas v závislosti na časové základně

### Ukázky obrazovek při nastavování dalších parametrů bloku zpožděného vypnutí.

Nastavený čas je 4 hodiny : 10 minut



Obr. 30



Obr. 31

**Obr. 30** Parametrický režim je povolený [ + ], remanence bloku je zakázána [ / ].

**Obr. 31** Parametrický režim je zakázáný [ - ], remanence bloku je povolena [ R ].

5 Max. hodnota 99

6 Max. hodnota 59

## 4.5. Pojmenování a ochrana programu

### Přiřazení názvu programu

[Edit Name]

Maximální délka jména programu je 16 znaků.

Může obsahovat malá, velká písmena, číslice a speciální znaky.

#### Postup:

- Přesuneme kurzor na položku [Edit Name].
- Potvrdíme volbu [Edit Name] klávesou OK.
- Použitím kurzorových kláves nahoru / dolů listujeme seznamem znaků. Listovat můžeme od první, nebo od poslední pozice. Záleží na poloze znaku v řadě. K dispozici je celkem 94 znaků. Tabulka znaků v [1] část 3.7.4.
- Pro vložení dalšího znaku posuneme kurzor pomocí kurzorové klávesy vpravo na další pozici. Jednotlivé znaky jména programu se nepotvrzují.
- Konec zadávání jména programu potvrdíme klávesou OK.

#### Příklady na vybírání písmen a znaků.

Stisknutím kurzorové klávesy kurzor nahoru, výsledek bude A.

Stisknutím kurzorové klávesy kurzor dolů 4x, výsledek bude {.

#### Úkol:

##### Pojmenovat program ABC:

- Stiskneme kurzorovou klávesu dolů, vybereme A.
- Stiskneme kurzorovou klávesu vpravo, přejdeme na další znak.
- Stiskneme dvakrát kurzorovou klávesu dolů, vybereme B.
- Stiskneme kurzorovou klávesu vpravo, přejdeme na další znak.
- Stiskneme třikrát kurzorovou klávesu dolů, vybereme C.
- Ukončíme zadávání jména programu: stisknutím klávesy OK.

Program se nyní jmenuje ABC a LOGO! je zpátky v programovacím menu.

Název programu můžeme kdykoliv změnit stejným způsobem.

## Ochrana programu heslem

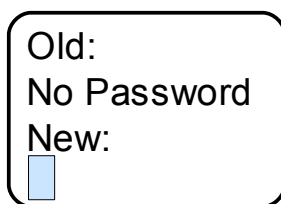
[Password]

Program můžeme chránit heslem před neoprávněným přístupem a nepovolenými změnami. Maximální délka hesla je 10 znaků a v nabídce jsou velká písmena A-Z. Nastavit, editovat a deaktivovat heslo můžeme v menu [Password].

### Úkol:

#### Nastavit heslo programu na: AA

- V programovacím režimu přesuneme kurzor na položku [Password] a potvrdíme volbu klávesou OK.  
Pro pohyb v abecedě použijeme kurzorové klávesy stejným způsobem jako při zadávání jména.



K obrázku 32.

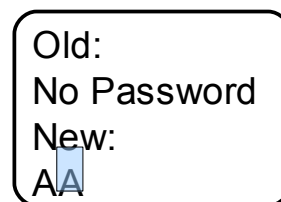
Old: (staré heslo).

No Password: (program je bez hesla).

New: (nové heslo).

*Obr. 32*

- Z místa kurzorové značky začneme vkládat heslo.



*Obr. 33*

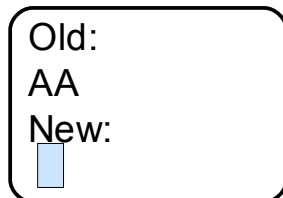
- Zadávání hesla ukončíme stisknutím klávesy OK.  
Program je nyní chráněn heslem AA.  
Pokud by bylo zadávání hesla přerušeno stisknutím klávesy ESC, vrátí se LOGO! do programovacího režimu bez uložení hesla. Heslo můžeme zadat také pomocí software LogoSoftComfort.

## Heslo změna

Když chceme změnit heslo, musíme předem zadat do pole Old aktuální heslo.

V programovacím menu:

- Přemístíme kurzor na [Password].
- Potvrdíme [Password] klávesou OK. Do pole OLD vložíme staré heslo (zde AA).
- Po stisknutí klávesy OK se zobrazí pole pro vložení nového hesla (obr. 34).

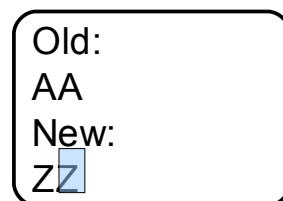


Old:  
AA  
New:  
|

*Obr. 34*

Nyní můžeme vložit do pole New nové heslo, např: ZZ.

- Stiskneme kurzorovou klávesu nahoru, vybereme Z.
- Stisknutím kurzorové klávesy vpravo přesuneme kurzor na pozici dalšího znaku.
- Stiskneme kurzorovou klávesu nahoru a vybereme Z, obr. 35.



Old:  
AA  
New:  
ZZ|

*Obr. 35*

- Potvrdíme nové heslo klávesou OK.

## Heslo deaktivace

Stejně jako při změně hesla i při deaktivaci musíme zadat nejprve heslo aktuální.

V programovacím menu zvolíme nabídku [Password].

- Zadáme staré heslo do pole Old.
- Do pole New ne zadáme žádné nové heslo.
- Potvrdíme toto „prázdné“ heslo klávesou OK.

Původní heslo bylo deaktivováno.

## Vymazání programu chráněného heslem.

Jsme požádáni o zadání hesla. Protože heslo neznáme, potvrdíme opakovaně prázdné pole. Po třetím vložení chybného hesla se program smaže.

## 4.6. Ruční programování podle výkresu

Podle přiložených logických schémat na obr. 36 – 40 naprogramujte ručně LOGO!. Na obrázcích jsou sejmuté obrazovky při simulaci programu v LogoSoftComfort. Proto jsou bloková čísla uvedena v jiném tvaru, než je známe z displeje modulu.

Modrá barva znázorňuje signál 0. Červeně je označený signál 1.

Na vstupech modulu I1 – I8 jsou připojena tlačítka a vypínače. Po dobu jejich aktivace je na příslušném vstupu 1.

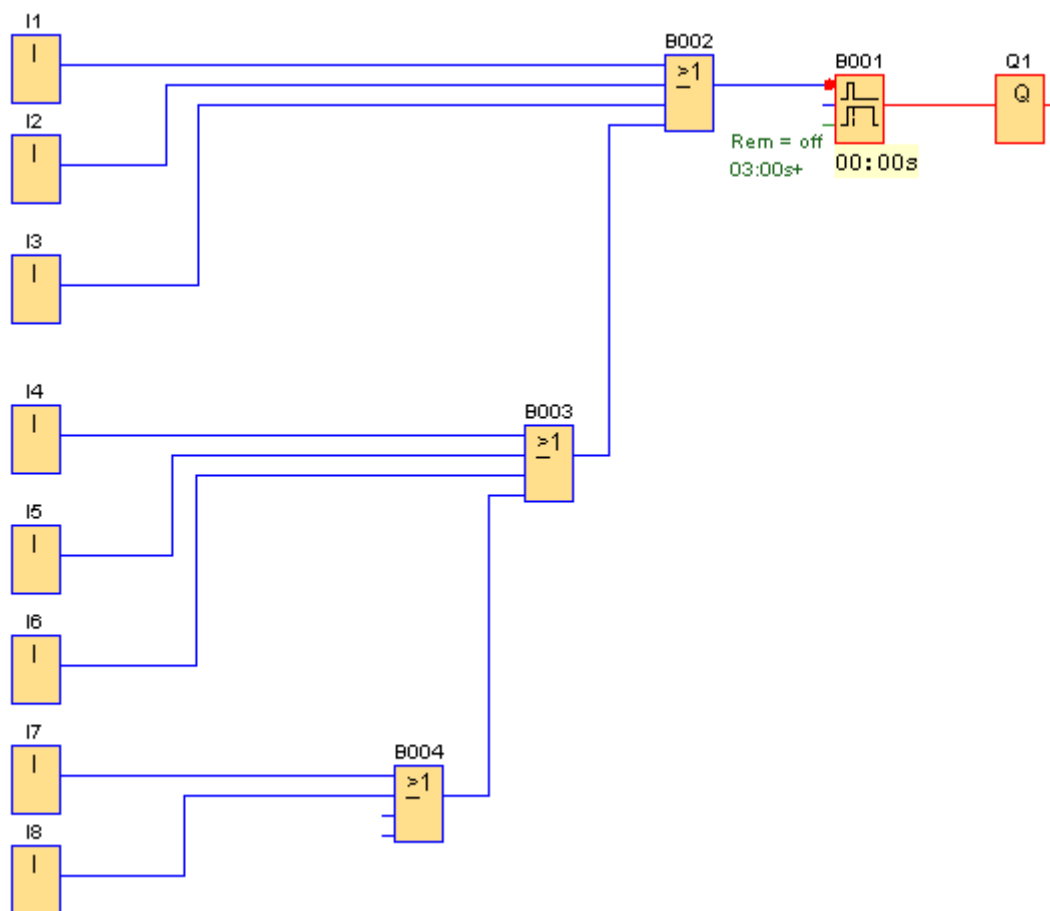
Na Q1 – Q4 jsou připojené LED.

### Příklad 5. Obr. 36

Po spuštění programu Q1 sepně. Tento stav je zachycený na obr. 36.

Přivedením dostatečně dlouhého signálu 1 na libovolný vstup se Q1 vypne se zpožděním 3 sekundy.

Nepřehlédněte, že vstup Trg u B001 je negovaný.



Obr. 36. K příkladu 5

## Příklad 6. Obr. 37

Po spuštění programu Q1 sepne. Tento stav je zachycený na obr. 37.

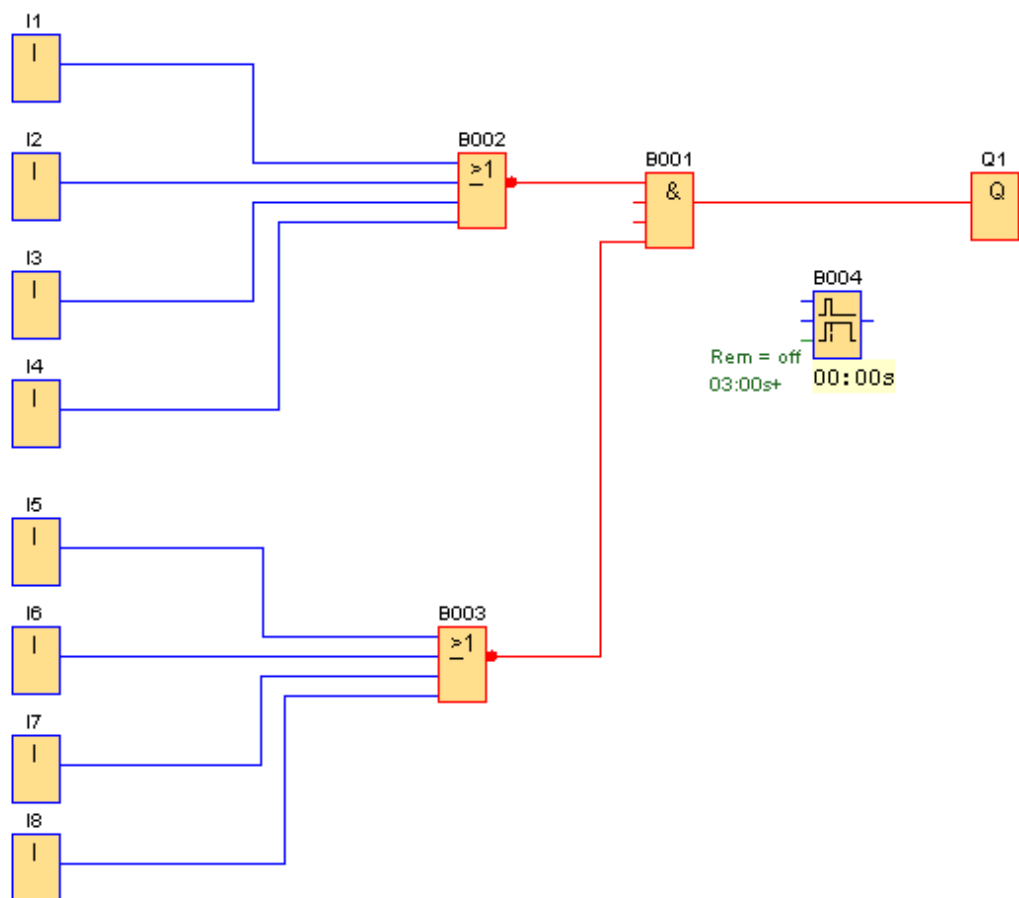
Přivedením signálu 1 na libovolný vstup vypne Q1.

Po odladění programu vložte v editačním režimu mezi konektor Q1 a B001 blok zpožděného vypnutí (B004).

Čas zpoždění nastavte na  $T = 3$  sekundy.

Potom po přivedení dostatečně dlouhého signálu 1 na libovolný vstup vypne Q1 se zpožděním 3 sekundy

V režimu přiřazení parametrů pozorujte průběh časování B004.



*Obr. 37. K příkladu 6*

### Příklad 7. Obr. 38

Q1, Q2, Q3, Q4 sepnou, když je na vstupech I5 a I6 a (I1 nebo I2) signál 1.

Tento stav je zachycený na obr. 38.

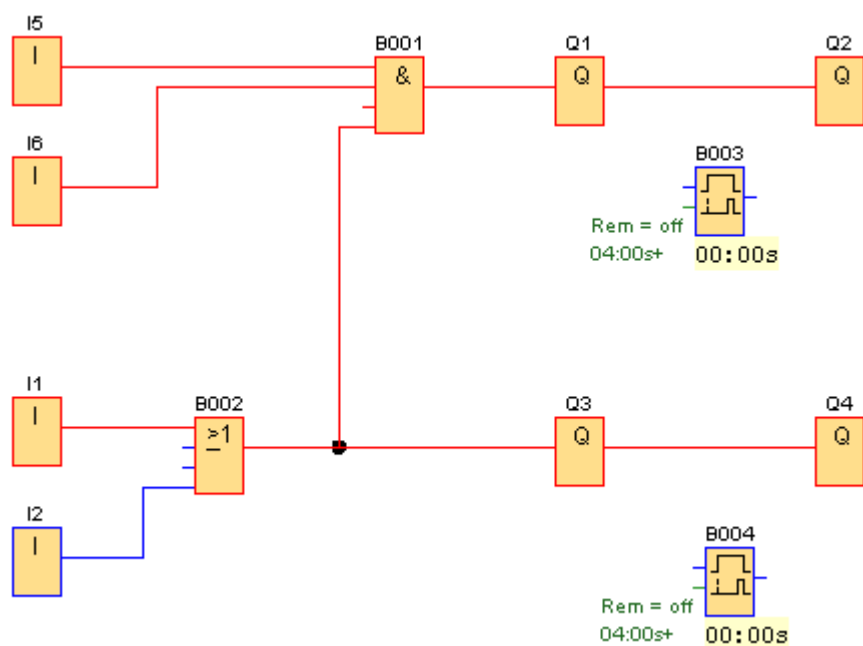
Q3 a Q4 sepnou, když je signál 1 na vstupu I1 nebo I2.

Po odladění programu vložte v editačním režimu mezi konektory Q1, Q2 a Q3, Q4 bloky zpožděného zapnutí.

Čas zpoždění nastavte na  $T = 4$  sekundy.

Potom, při splnění výše uvedených vstupních podmínek, bude mezi sepnutím Q1 a Q2, (Q3 a Q4) zpoždění 4 sekundy.

V režimu přiřazení parametrů pozorujte průběh časování B003 a B004.



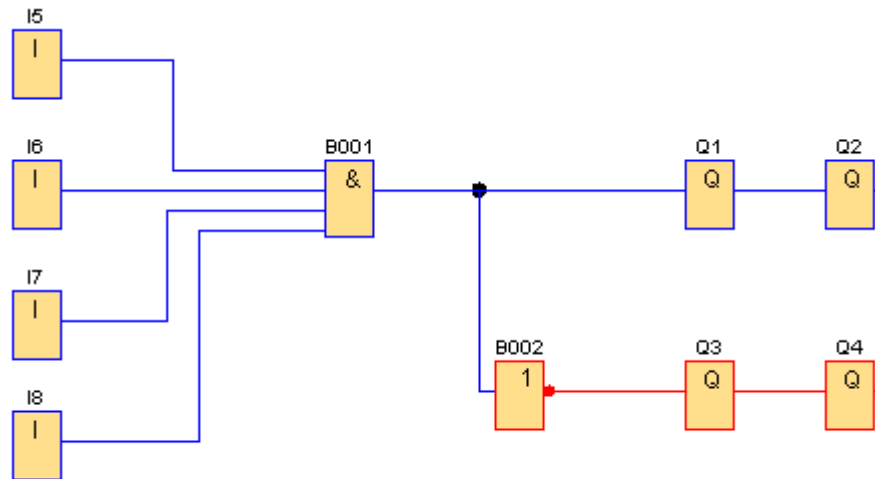
**Obr. 38.** K příkladu 7

### Příklad 8. Obr. 39

Po spuštění programu sepnou Q3 a Q4. Tento stav je zachycený na obr. 39.

Když je přiveden současně na všechny čtyři vstupy signál 1, vypne Q3 a Q4 a zároveň zapne Q1 a Q2.

Na obrazovce vstupů a výstupů pozorujte jak se mění jejich vzhled v závislosti na druhu signálu.

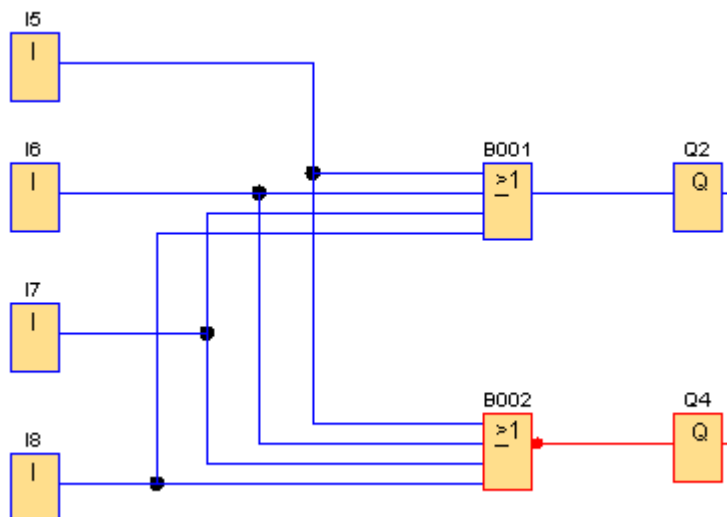


*Obr. 39. K příkladu 8*



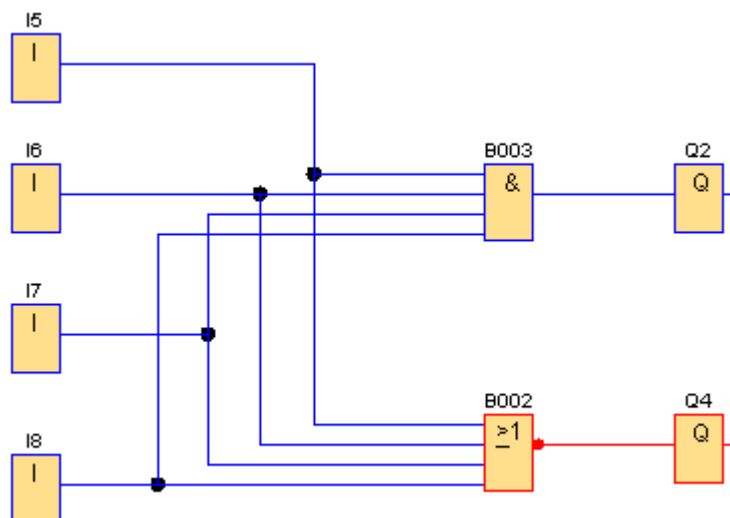
## Příklad 9. Obr. 40

Po spuštění programu sepne Q4. Tento stav je zachycený na obr. 40. Když je na kterýkoliv vstup přiveden signál 1, Q4 vypne a Q2 zapne.



*Obr. 40. K příkladu 9 výchozí stav*

Po odladění programu nahradte v editačním režimu blok B001 (OR) blokem B003 (AND). Po spuštění programu sepne Q4. Tento stav je zachycený na obr. 41. Q4 vypne, když je na kterýkoliv vstup přiveden signál 1. Q2 zapne, když je současně na všech vstupech signál 1.



*Obr. 41. K příkladu 9 stav po editaci*

## 4.7. Náměty k opakování

- Písmenová značka pro stejnosměrné a střídavé napětí.
- Oblasti použití univerzálního programovacího modulu. Jeho výhody a nevýhody.
- Rok výroby modelu LOGO! OBA0, typové označení aktuálního modelu LOGO!.
- Maximální počet digitálních vstupů a výstupů při použití rozšiřujících modulů.
- Maximální průřezy vodičů pro připojení LOGO! 12/24 RC, doporučená šířka čepele šroubováku, způsoby upevnění základního modulu při montáži.
- Minimální napětí a proud signálu 1 pro LOGO! 12/24RC .
- Maximální napětí a proud signálu 0 pro LOGO! 12/24RC.
- Velikost a druh napájecího napětí a hodnota pojistky jistící napájení pro základní modul LOGO! 12/24RC.
- Remanence bloku a jaké je výchozí nastavení.
- Přiřazení parametrů bloku a jaké je výchozí nastavení.
- Činnost LOGO! v RUN.
- Na které vstupy (LOGO! 12/24RC) je možno připojit ultrazvukový polohový snímač a které vstupy můžeme použít pro vyhodnocení analogového signálu.
- Maximální výše a druh napětí a maximální zatěžovací proud výstupů LOGO! 12/24RC.
- Maximální hodnota jističe výstupu LOGO! 12/24RC.
- Písmenové označení digitálních vstupů a výstupů pro LOGO! 12/24RC.
- Maximální počet použitelných bloků v programu a počet předprogramovaných základních a speciálních funkcí pro aktuální model LOGO!.

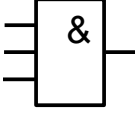

## 5. Základní funkce 1. část

Bloky základních funkcí (bloky GF) představují jednoduché funkce Booleovy logiky. Jednotlivé vstupy bloků GF můžeme invertovat. Postup při ručním invertování vstupu v části 4.2.

Zobrazení v LOGO!	Zobrazení ve schématu	Popis základní funkce
		AND, logický součin
		NAND, negovaný součin
		OR, logický součet
		NOR, negovaný součet
		XOR, exklusivní součet
		NOT, invertor

Tab. 3. Bloky GF

## 5.1. AND, logický součin

Zobrazení v LOGO!	Zobrazení ve schématu	Popis základní funkce
		Stav výstupu AND je 1, když jsou všechny vstupy 1. Nepřipojený vstup hradla je nastaven automaticky na 1.
	Sériové spojení více kontaktů. Poloha kontaktů odpovídá stavu, kdy je na vstupech hradla 0.	

**Tab. 4.** AND, logický součin

Ke každé funkci je možné na základě příslušného pravidla sestavit logickou (pravdivostní) tabulku, která určuje hodnotu funkce pro všechny možné kombinace vstupních hodnot.

Počet sloupců tabulky je  $n+1$ , počet řádků je  $2^n$ .

$n$  = počet vstupů.

Pole s označením vstupů a výstupu se nezapočítávají.

Z důvodu úspory místa jsou ve všech následujících textech příslušné bloky GF se třemi vstupy, přestože OBA5 má bloky se čtyřmi vstupy. Logické tabulky by byly sice jen o jeden sloupec širší, ale o osm řádků delší.

### Logická tabulka pro AND

I1	I2	I3	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

**Tab. 5**

**Příklad.**

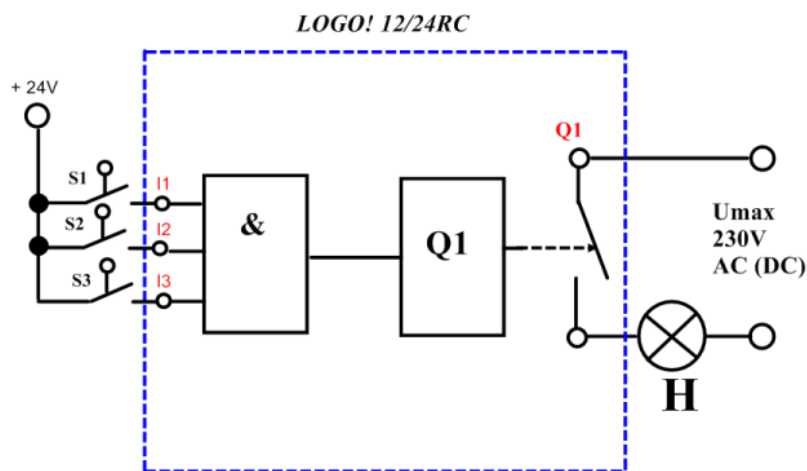
Když je bezpečnostní záložka na svém místě dává čidlo signál 1. Signalizace sepne když jsou všechny tři bezpečnostní záložky v požadované poloze.

Řešení příkladu odpovídá poslední řádka v tabulce 5.

**Blokové řešení příkladu a vnější spoje LOGO! obr. 42.**

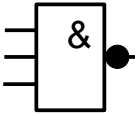
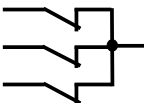
I a Q je označení vstupních a výstupních konektorů v programu. Stejně jsou označené vstupní a výstupní svorky modulu LOGO!.

Spínače S1 – S3 jsou připojené na vstupní svorky I1 – I3. Při stisknutí spínače je na vstup přiveden signál 1. Přes výstupní kontakt Q1 je připojena signální žárovka s vlastním napájením.



*Obr. 42. Program v LOGO! a vnější spoje*

## 5.2. NAND, negovaný součin

Zobrazení v LOGO!	Zobrazení ve schématu	Popis základní funkce
		Stav výstupu NAND je 0 jen v případě, když jsou všechny vstupy 1. Nepřipojený vstup hradla je nastaven automaticky na 1.
	Paralelní zapojení rozpínacích kontaktů. Poloha kontaktů odpovídá stavu, kdy je na vstupech hradla 0.	

**Tab. 6.** NAND, negovaný součin

### Logická tabulka pro NAND

I1	I2	I3	Q
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

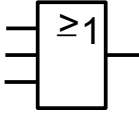
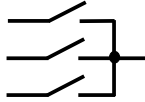
**Tab. 7**

### Příklad.

Při nedostatku vody má příslušný senzor signál 0.

Výstražná siréna zazní když kterákoliv ze tří nádrží má nedostatek vody.

### 5.3. OR, logický součet

Zobrazení v LOGO!	Zobrazení ve schématu	Popis základní funkce
		Stav výstupu OR je 1, když je alespoň jeden vstup 1. Nepřipojený vstup hradla je nastaven automaticky na 0.
	Paralelní spojení spínacích kontaktů Poloha kontaktů odpovídá stavu, kdy je na vstupech hradla 0.	

**Tab. 8.** OR, logický součet

#### Logická tabulka pro OR

I1	I2	I3	Q
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

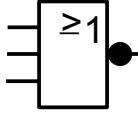

**Tab. 9**

#### Příklad.

V domě jsou čtyři byty.

Hlavní domovní dveře je možné dálkově otevřít z kteréhokoliv bytu.

## 5.4. NOR, negovaný součet

Zobrazení v LOGO!	Zobrazení ve schématu	Popis základní funkce
		Stav výstupu NOR je 0 když je alespoň jeden vstup 1. Nepřipojený vstup hradla je nastaven automaticky na 0.
	Sériové spojení rozpínacích kontaktů Poloha kontaktů odpovídá stavu, kdy je na vstupech hradla 0	

**Tab. 10.** NOR, negovaný součet

### Logická tabulka pro NOR

I1	I2	I3	Q
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

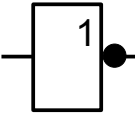

**Tab. 11**

### Příklad.

Při zaplavení vodou dá čidlo signál 1. Při zaplavení i jednoho čidla zhasne nápis „VSTUP POVOLEN“.



## 5.5. NOT, negace, invertor

Zobrazení v LOGO!	Zobrazení ve schématu	Popis základní funkce
		Stav výstupu je 0 když je vstup 1. Blok NOT je invertor signálu.
	Rozpojovací kontakt. Poloha kontaktu odpovídá stavu, kdy je na vstupu hradla 0	

**Tab. 12:** NOT, invertor

### Logická tabulka pro NOT

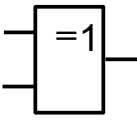
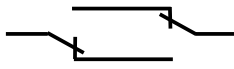
I	Q
1	0
0	1

**Tab. 13**

### Příklad.

Funkční senzor je charakterizován log.1.  
 Při poruše senzoru zazní výstraha.

## 5.6. XOR, exkluzivní OR

Zobrazení v LOGO!	Zobrazení ve schématu	Popis základní funkce
		Stav výstupu XOR je 1, když mají vstupy rozdílné stavy. Nepřipojený vstup hradla je nastaven automaticky na 0.
	XOR znázorňují 2 přepínací kontakty zapojené do série Poloha kontaktů odpovídá stavu, kdy je na vstupech hradla 0	

Tab. 14. XOR, exkluzivní OR

### Logická tabulka pro XOR

I1	I2	Q
1	1	0
1	0	1
0	1	1
0	0	0

Tab. 15

### Příklad.

V objektu jsou dva senzory přítomnosti.

Siréna zazní v případě že jen jeden, nebo jen druhý senzor dává signál 1.

### Shrnutí o vstupech GF

Jednotlivé vstupy můžeme invertovat. Potom signál 1 na vstupu se překonvertuje a do bloku vstupuje jako 0. Výsledek stejný, jako když neinvertovanému vstupu předřadíme NOT. Vstupy jsou označeny od 1 do max.4 a mimo odkazů u dlouhých, přerušovaných spojů se v dokumentaci jejich označení neuvádí. Pro správnou funkci bloku není rozhodující, v jakém pořadí se vstupy obsazují signály, či který vstup zůstane neobsazený.

Nepoužitý vstup je nastaven automaticky na takovou log. hodnotu, aby funkce byla zachována. AND má výstup 1, jsou-li všechny vstupy 1, neobsazený vstup AND musí mít hodnotu 1. OR má výstup 1, je-li alespoň jeden ze vstupů 1, neobsazený vstup OR musí mít hodnotu 0.

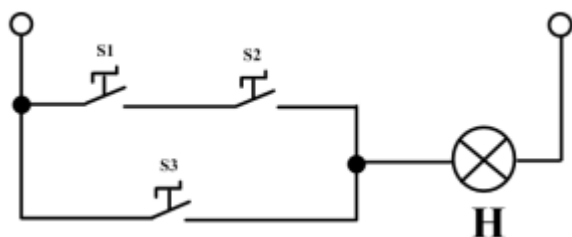
Obecně řečeno, u bloků GF je hodnota neobsazeného vstupu různá, v závislosti na druhu funkce.

## 5.7. Řešené příklady

### Příklad 10.

Zařízení je možné spustit buď současným aktivováním dvou klíčů směnových mistrů, nebo aktivací jednoho klíče vrchního mistra.

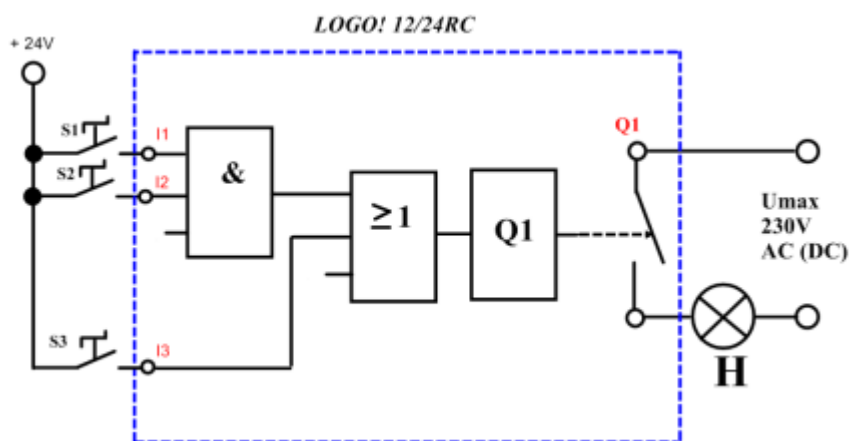
Schéma zapojení je znázorněno jak pomocí drátové logiky na obr. 43, tak pomocí bloků v modulu LOGO! na obr. 44.



*Obr. 43. Příklad 10, schéma reléové logiky*

### Vnější zapojení vstupů, výstupů do modulu a program v LOGO!.

Svorky označené I1, I2, I3 jsou vstupní svorky LOGO!. Konektor Q1 spíná výstupní kontakt relé.



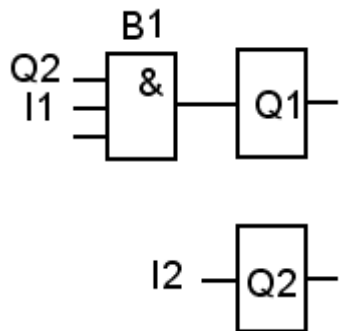
*Obr. 44. Příklad 10, vnější spoje a blokové schéma v LOGO!*

**Příklad 11.** Sepnutí výstupu je podmíněno sepnutím jiného výstupu.

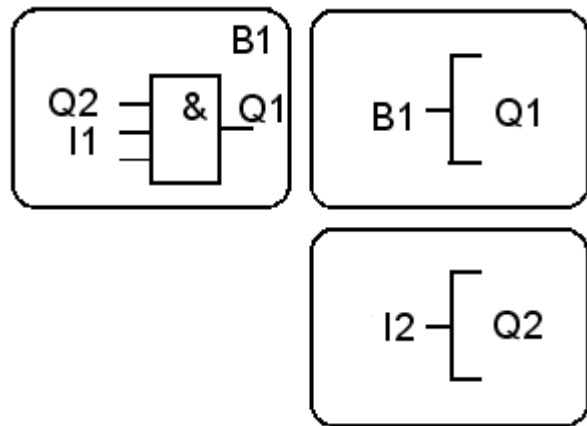
Motor vrtačky (Q1) nesmí běžet bez spuštěného čerpadla chlazení (Q2).

Spínač na vstupu I1 ovládá Q1.

Spínač na vstupu I2 ovládá Q2.



*Obr. 45. Blokové schéma*



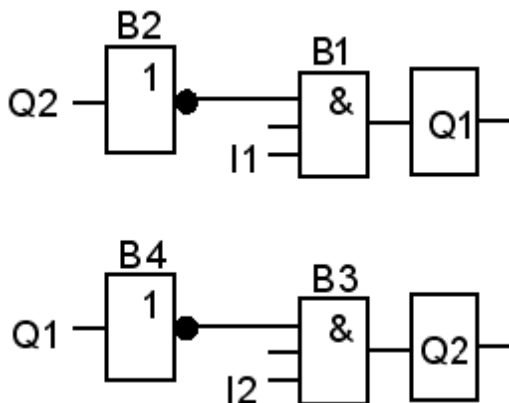
*Obr. 46. Náhledy jednotlivých obrazovek*

**Příklad 12.** Když je sepnutý jeden výstup, nesmí sepnout druhý.

Spínač na vstupu I1 ovládá Q1.

Spínač na vstupu I2 ovládá Q2.

Když běží Q1 nejde spustit Q2 a naopak.



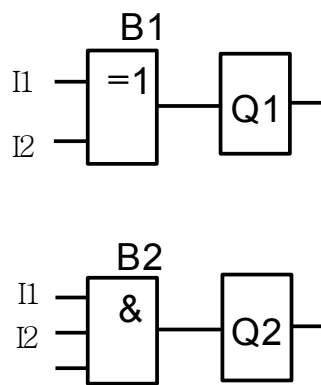
*Obr. 47*

Poprvé naprogramujte ručně podle obr.47.  
Podruhé místo bloků B2 a B4 (NOT) invertujte příslušné vstupy na B1 a B3.

**Příklad 13.**

vstup		vstup	výstup
Jen I1 = 1	nebo	Jen I2 = 1	Q1 = 1
I1 = 1	a	I2 = 1	Q2 = 1
I1 = 0	a	I2 = 0	Q1 = 0 Q2 = 0

*Tab. 16*



*Obr. 48*

## 5.8. Příklady na procvičení

*Pro všechny příklady platí: Na vstupech LOGO! jsou připojené spínače. Po dobu aktivace spínače je na vstupu signál 1.*

### Příklad 14.

Čtyři výstupy jsou zapojené v sérii. Výstupy sepnou po stisknutí i jednoho ze 6 spínačů.

### Příklad 15.

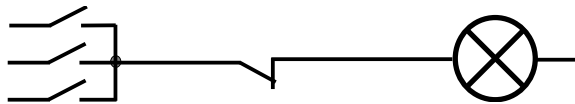
Čtyři výstupy jsou zapojené paralelně. Výstupy sepnou až po stisknutí všech 7 spínačů.

### Příklad 16.

Ovládání tabulových nůžek: Nůžky pracují, když je současně  $S1, S2, S3 = 1$ . Aby nedošlo k úrazu, musí obsluha stisknout současně rukama tlačítka a zároveň sešlápnout nožní spínač. Dálkově se nůžky vypínají nouzovým tlačítkem  $S5$  s aretací polohy VYPNUTO.

### Příklad 17.

Podle obr. 49 navrhnete blokové schéma pro LOGO!.



Obr. 49

### Příklad 18.

Dopravník obilí plnicí silo obilím se ovládá vypínačem  $S5$ . Silo má pod podlahou tlakové spínače  $S2, S3$ . Když se jeden z nich aktivizuje hmotností obilí, rozsvítí se signální žárovka a zastaví se dopravník.

### Příklad 19.

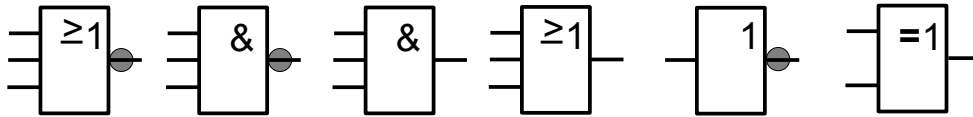
Po spuštění LOGO! sepnou výstupy  $Q1$  a  $Q3$ . Signál na kterémkoliv ze tří vstupů LOGO! vypne  $Q1, Q3$  a sepne  $Q2, Q4$ .

### Příklad 20.

Pod podlahou sila jsou tři spínače ovládané hmotností obilí. Když sepnou právě dva spínače, svítí žárovka na výstupu  $Q2$  (2/3 SILA PLNÉ). Když sepnou tři spínače, svítí jen žárovka na výstupu  $Q3$  (SILO JE PLNÉ).

## 5.9. Opakování, základní funkce

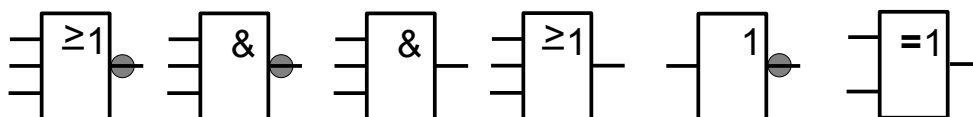
- Označ číslem 1 značku bloku NAND.
- Označ číslem 2 značku bloku OR.
- Označ číslem 3 značku bloku XOR.



*Obr. 50. Bloky základních funkcí*

- Nakresli zobrazení bloku AND ve schématu pomocí kontaktů relé.
- Nakresli zobrazení bloku NOR ve schématu pomocí kontaktů relé.
- Nakresli zobrazení bloku NOT ve schématu pomocí kontaktů relé.
- Vypiš pravdivostní tabulku pro logický součin (tři vstupy).
- Vypiš pravdivostní tabulku pro negovaný součet (tři vstupy).
- Vypiš pravdivostní tabulku pro invertor.
- Při ručním programování LOGO! nepoužiješ jeden vstup bloku OR. Jakou logickou hodnotu bude mít neobsazený vstup?

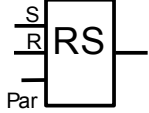
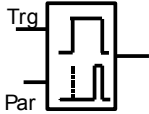
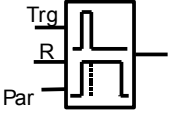
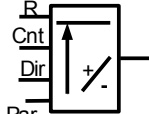
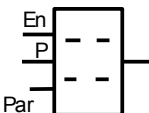
- 
- Označ číslem 1 značku bloku AND.
  - Označ číslem 2 značku bloku NOR.
  - Označ číslem 3 značku bloku NOT.



*Obr. 51. Bloky základních funkcí*

- Nakresli zobrazení bloku OR ve schématu pomocí kontaktů relé.
- Nakresli zobrazení bloku XOR ve schématu pomocí kontaktů relé.
- Nakresli zobrazení bloku NAND ve schématu pomocí kontaktů relé.
- Vypiš pravdivostní tabulku pro logický součet (tři vstupy).
- Vypiš pravdivostní tabulku pro exkluzivní součet.
- Vypiš pravdivostní tabulku pro negovaný součin (tři vstupy).
- Při ručním programování LOGO! nepoužiješ jeden vstup bloku AND. Jakou logickou hodnotu bude mít neobsazený vstup?

## 6. Speciální funkce 1. část

Zobrazení v LOGO!	Popis funkce	Název bloku
	Samodržné relé je jednoduchá paměť	Samodržné relé RS
	Výstup sepne, když uplyne nastavený čas	Blok zpožděného zapnutí
	Výstup vypne, když uplyne nastavený čas	Blok zpožděného vypnutí.
	Výstup sepne / vypne, když byla dosažena nastavená hodnota	Dopředný a zpětný čítač.
	Slouží k zobrazení předdefinovaného textu, nebo parametru v režimu RUN	Textové zprávy

Tab. 17. Speciální funkce – výběr



## Vstupy speciálních funkcí

### Označení vstupů

Vstupy bloků speciálních funkcí (vstupy SF) jsou označené písmenovými zkratkami, odvozenými z anglických názvů charakterizujících činnost vstupu. Některé druhy vstupů najdeme u mnoha funkcí, jiné jen ojediněle.

Hojně používaný vstup je Trg (trigger *vysl. trigr* = spouštěč), který má téměř polovina SF bloků.

Na opačném pólu stojí Dir (direction *vysl. direkšn* = směr), který najdeme u bloku jediného.

Více se dozvíme o funkcích vstupů v části popisující jednotlivé bloky SF.

Rozlišujeme dvě skupiny vstupů SF bloků.

### Logické vstupy

Na logické vstupy přivádíme signály a použijeme je pro spojení s jinými bloky, nebo konektory. Tyto vstupy můžeme invertovat.

Neobsazený vstup je vždycky = 0.

### Parametrové vstupy

Parametrovým vstupům nejsou přiřazeny žádné signály, slouží ke konfiguraci příslušné hodnoty bloku.

Nejběžnější parametrové vstupy:

- T (time *vysl. tajm*) časová hodnota daného bloku. (Př: čas pro vypnutí / zapnutí).
- Par (parametr) označuje vstup, kde můžeme povolit remanenci<sup>7</sup> bloku.

**Povolená remanence** znamená, že aktuální data bloku jsou po výpadku napájení uchována a po obnovení napájení pokračuje blok v činnosti z místa přerušení.

Existují dvě volitelné možnosti nastavení remanence.

/ zakázaná remanence, aktuální data nejsou uchována (**výchozí stav bloku**).

R povolená remanence, data jsou uchována

Čítač provozních hodin, týdenní spínač, roční spínač a regulátor PI mají vždy povolenou remanenci.

**Ochrana parametrů.** (Viz též část 4.4. Nastavení parametru bloku.) Při programování můžeme určit zda mají či nemají být parametry bloku zobrazené a editované v režimu přiřazení parametru.

Tato možnost se netýká remanence bloku! Remanence bloku sice patří mezi parametry, ale její nastavení můžeme měnit pouze v programovacím / editačním režimu.

Existují dvě volitelné možnosti nastavení ochrany parametrů.

+ : Atribut dovoluje přístup ke čtení a změně parametru v módu přiřazení parametru.

**Toto nastavení je výchozím stavem bloku.**

- : Parametry bloku jsou v módu přiřazení parametrů chráněny proti čtení a zápisu.

V tomto případě je můžeme editovat jen v programovacím (editovacím) módu.

Při pokusu o editaci takto chráněných parametrů se zobrazí na displeji text „NO PARAM PRESS ESC“. Stejný text se objeví i v případě, že program neobsahuje blok s parametrem.

**Parametr některých bloků je možné měnit i hodnotou jiného, už nakonfigurovaného bloku.**

Master (*doslovný překlad pán*) je v tomto případě označení pro řídicí a slave (*sluha*) pro řízený blok. Praktická ukázka programování dvojice čítač (master) a blok zpožděné zapnutí (slave) je v části 8.1. Dopředný a zpětný čítač. V [1] v části 4.4.1. jsou uvedeny všechny řídicí funkce.

---

<sup>7</sup> Remanence je paměť bloku.

## Přesnost času

Maximální chyba pro LOGO! je  $\pm 0,02\%$ . (*Chyba za 24 hodin<sup>8</sup> je max  $\pm 5$  sekund*).

Pokud je  $\pm 0,02\%$  z časového úseku menší než 0,02 sekundy, potom je max. chyba 0,02 s.

Všechny elektronické komponenty mají nepatrně odlišné parametry a to může způsobit rozdíly mezi nastavenými a skutečně dosaženými časy.

V případě že se dovolené tolerance u různých bloků sečtou, může dojít ke kolizi programu.

Tento stav se nazývá hazard.

## Chod vnitřních hodin při výpadku napájení

Vnitřní hodiny LOGO! se nezastaví při výpadku napájení okamžitě. Doba překlenutí výpadku je závislá na okolní teplotě. Při 25°C je to cca 80 hodin. Zastavené vnitřní hodiny se navenek projeví prohlášením obrazovky datum / čas a jsou nastavené na 00:00, neděle 1. ledna 2003.

Pokud by některé bloky měly uvedené hodnoty jako parametr, tak by s ním program po spuštění začal normálně pracovat.

## Chybová hlášení LogoSoftComfort

Při řešení příkladů pomocí software se mohou objevit tato chybová hlášení:

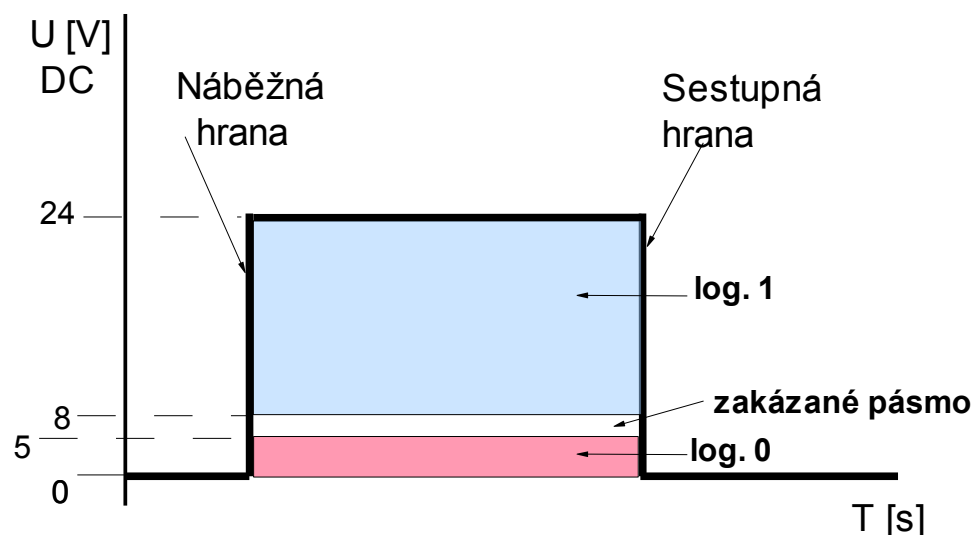
a) „Nekompatibilní konektory“ se objeví při pokusu napojit přípoj:

- ze vstupu na vstup
- z výstupu na výstup
- na vstup bloku, kde už je jeden přípoj připojen (viz obrázek č.57)

b) „Rekurze<sup>9</sup> dovoleny pouze přes výstupy nebo paměťové bity“ (viz obrázek č. 59) se objeví při pokusu vytvořit v programu vazbu, aniž by spoj vycházel z výstupu Q, nebo paměťového bitu M.

## Náběžná a sestupná hrana signálu

V souvislosti s činností některých bloků budeme mluvit o náběžné a sestupné hraně signálu.

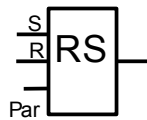


Obr. 52. Ideální tvar logického signálu

<sup>8</sup> Časovač je porovnáván s vysoce přesnou časovou základnou a jeho čas je zpřesňován.

<sup>9</sup> Rekurze - program volá sám sebe.

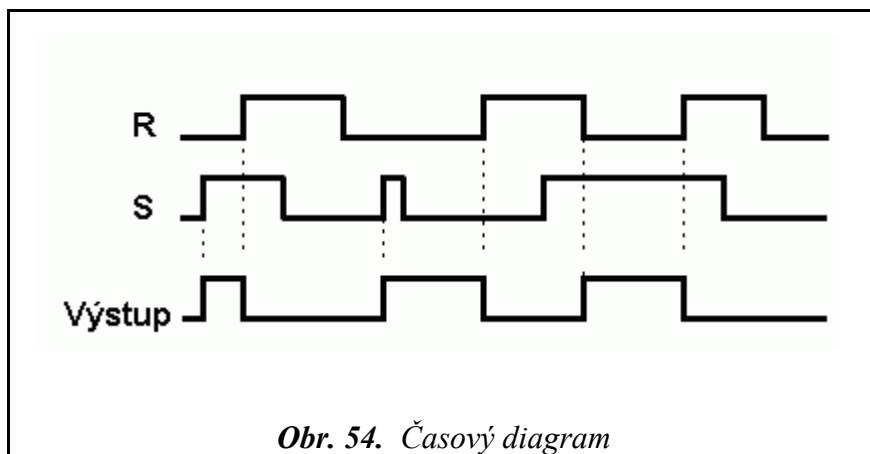
## 6.1. Blok samodržné relé RS



Samodržné relé je jednoduchá paměť.

**Obr. 53**

- Vstup S (Set *vysl. set* = nastavit): Signál na vstupu sepne výstup bloku.
- Vstup R (Reset *vysl. reset* = nulovat): Signál na vstupu vypne výstup bloku.  
Pokud je signál na obou vstupech současně je výstup vypnutý.  
Signál na vstupu R má prioritu.
- Vstup Par: Použijeme pro zapnutí / vypnutí remanence  
/ (remanence vypnutá)  
R (remanence zapnutá), stav výstupního signálu se ani po výpadku napájení nezmění.
- Výstup bloku je vstupem S zapnutý tak dlouho, dokud není přivedený signál na vstup R.



**Obr. 54. Časový diagram**

Hodnota výstupu závisí na stavu vstupu a na předchozím stavu výstupu.

### **Příklad.**

Výstup LOGO! je spínaný a vypínaný pomocí dvou tlačítek na vstupech I1 a I2.

Tlačítko S1 na vstupu I1 zapne výstup Q1.

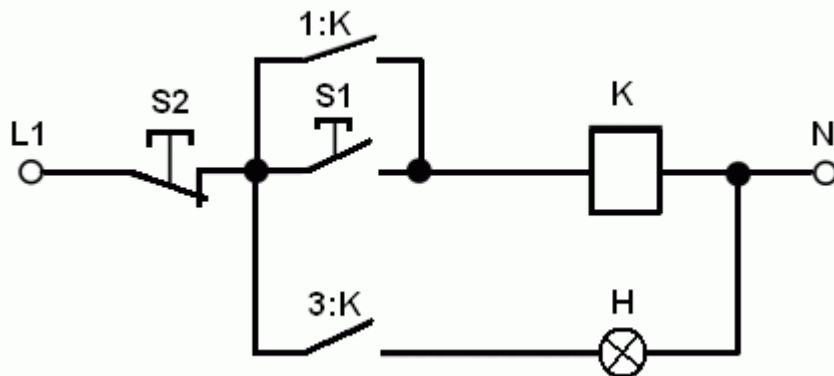
Tlačítko S2 na vstupu I2 vypne výstup Q1.

### Blok RS můžeme srovnat se stykačem.

Na obr. 55.

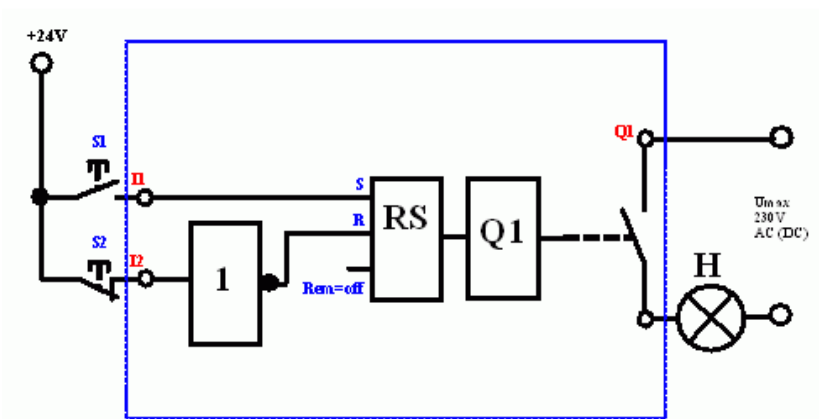
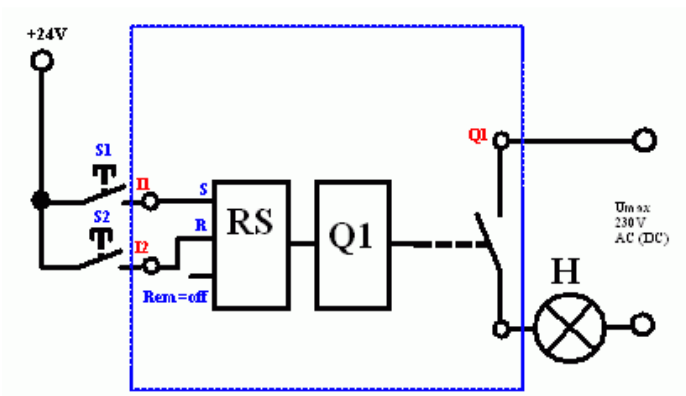
Po stisknutí S1 sepne cívka stykače K. Zároveň sepne kontakt 1:K a 3:K.

Kontakt 1:K přivádí proud do cívky stykače i po uvolnění S1, kontakt 3:K spíná spotřebič, v našem případě žárovku H.



Obr. 55. Ovládání stykače dvěma tlačítky

Blok RS vypneme přivedením napětí na vstup R. Na obr. 56 jsou dvě možná řešení připojení vypínacího tlačítka S2. Z hlediska bezpečnosti je vhodnější použít rozpínací kontakt S2<sup>10</sup>.



Obr. 56. Ovládání výstupu LOGO! dvěma tlačítky

10 Při simulaci v LogoSoftComfort nastavíme konektor I2 jako tlačítko rozpíjít.

## 6.1.1. Řešené příklady

### Příklad 21. Běh motoru jedním směrem

Navrhněte a odlaďte pomocí LogoSoftComfort:

Ovládání motoru je dvěma tlačítky.

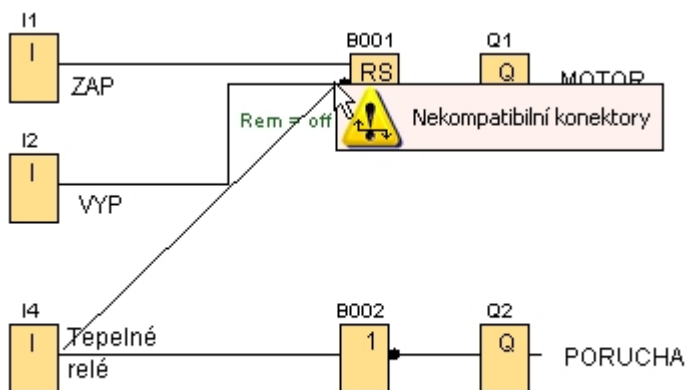
Výstup pro motor ... Q1

Tlačítko S1 na vstupu I1 ... ZAP

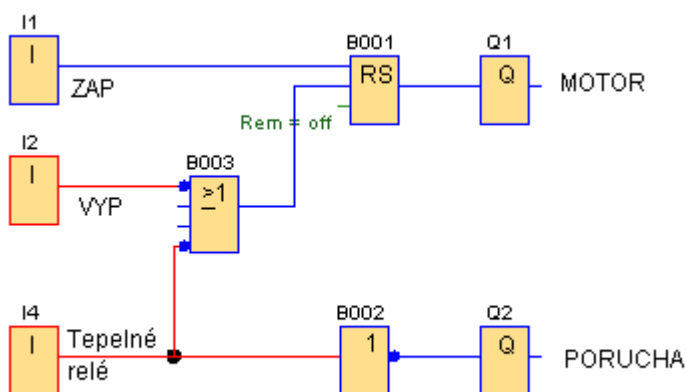
Tlačítko S2<sup>11</sup> na vstupu I2 ... VYP (v klidu sepnutý kontakt)

Motor je chráněn před přetížením tepelným relé (vstup I4, v klidu sepnutý kontakt). Když jde tepelné relé do funkce (I4=0), vypne se výstup Q1 a sepne výstup Q2 (PORUCHA).

Dokud není tepelné relé v klidovém stavu, motor nejde zapnout. Na obr. 57 a 58 je naznačen způsob řešení s ohledem na chybové hlášení. Blok B003 může být i NAND. Ovšem bez negovaných vstupů.



Obr. 57. Chybové hlášení LogoSoftComfort



Obr. 58. Řešení s ohledem na chybové hlášení

11 Při simulaci v LogoSoftComfort nastavíme konektory I2, I4 jako tlačítka rozpojit.

## Příklad 22. Reverzace otáček

Navrhněte a odlaďte pomocí LogoSoftComfort:

Motor se otáčí dvěma směry. Směry otáčení jsou navzájem blokovány.<sup>12</sup>

Vpravo..... výstup Q1.

Vlevo .....výstup Q2.

Zařízení je ovládáno třemi tlačítky.

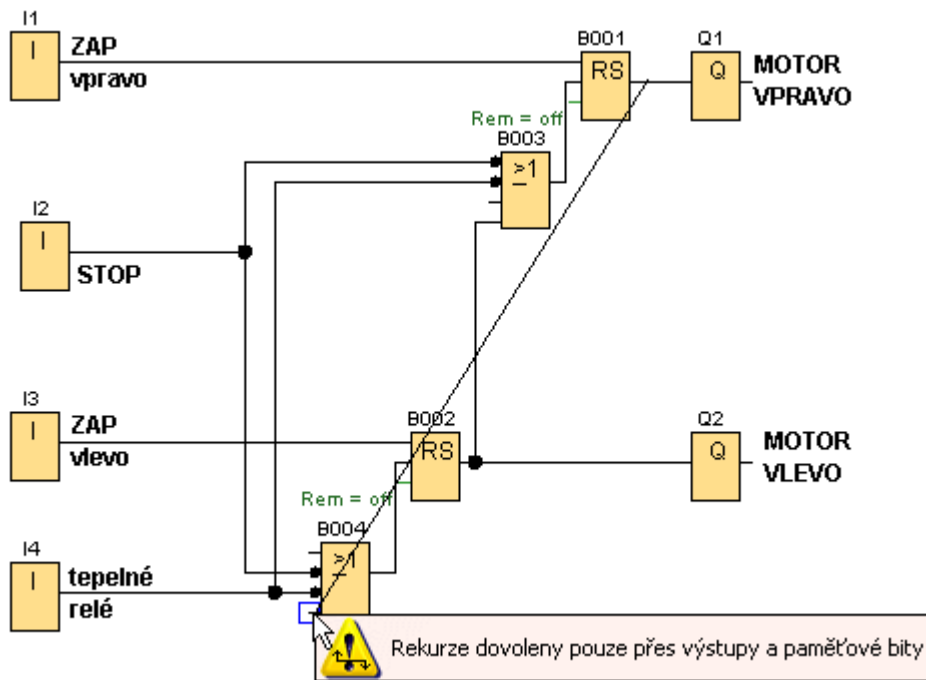
S1 (na vstupu I1)...vpravo.

S2 (I2, v klidu sepnutý kontakt)...stop.

S3 (I3)...vlevo.

Motor je chráněn před přetížením tepelným jisticím relé (vstup I4, v klidu sepnutý kontakt).

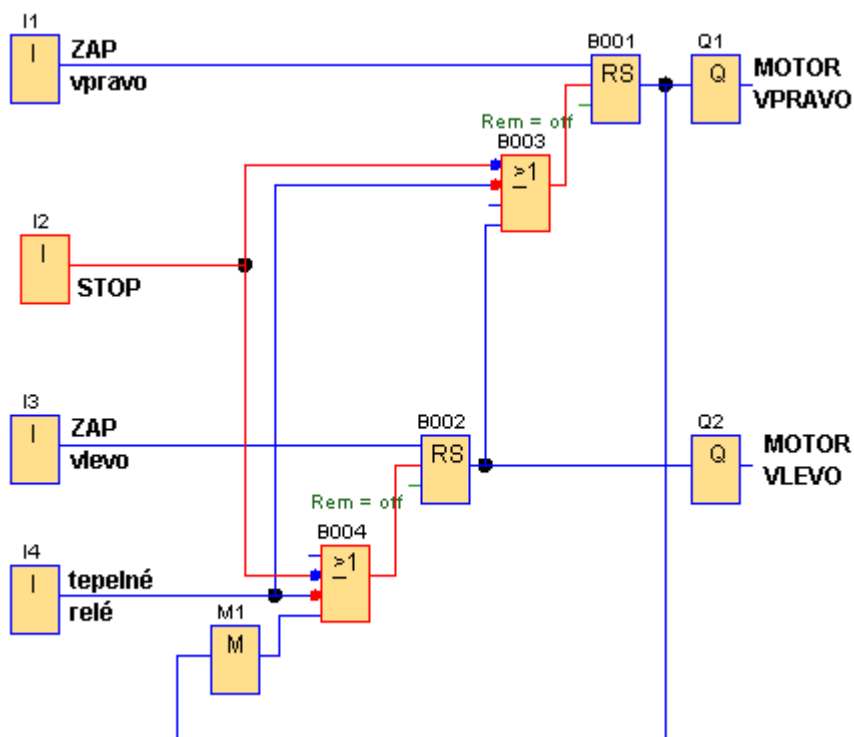
Na obr. 60 až 61 jsou dva způsoby řešení s ohledem na chybové hlášení.



Obr. 59. Chybové hlášení LogoSoftComfort

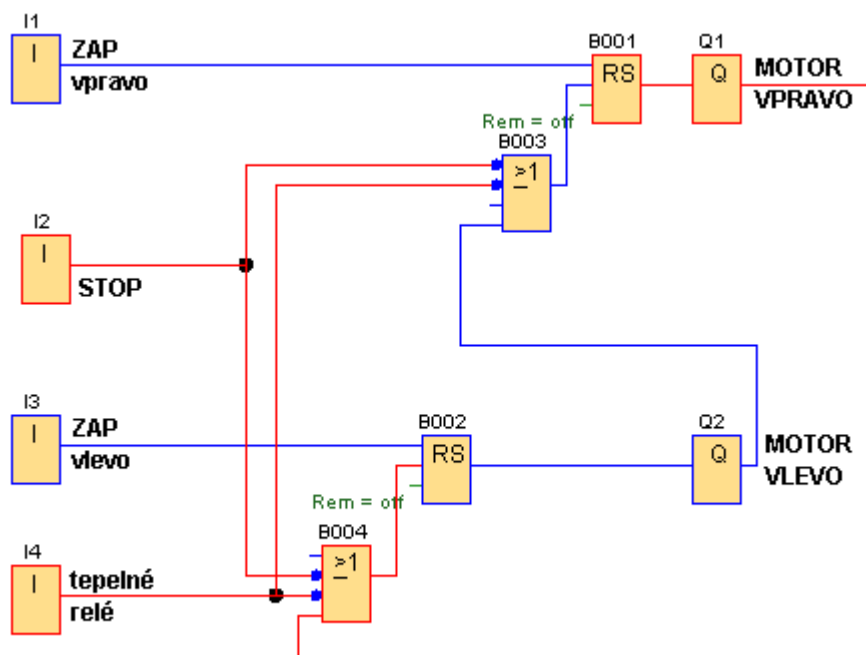
<sup>12</sup> takový způsob zapojení, který nedovolí spustit motor běžící ve směru Q1 zároveň ve směru Q2 a naopak.

Řešení rekurze pomocí paměťového bitu M1.



Obr. 60. K příkladu 22, tepelná ochrana je ve funkci

Řešení rekurze pomocí výstupů Q.



Obr. 61. K příkladu 22, sepnutý výstup pro směr MOTOR VPRAVO

**Příklad 23.** Hlavní motor stroje a čerpadlo chladící emulze

Navrhněte a odlaďte pomocí LogoSoftComfort.

Hlavní motor stroje je ovládán třemi tlačítky.

S1 (I1) ZAPNOUT směr točení vpravo (Q1).

S2 (I2, v klidu sepnutý kontakt) STOP oba směry.

S3 (I3) ZAPNOUT směr točení vlevo (Q2).

Oba směry jsou navzájem blokovány.

Motor čerpadla chladící emulze (Q3) je ovládán dvěma tlačítky.

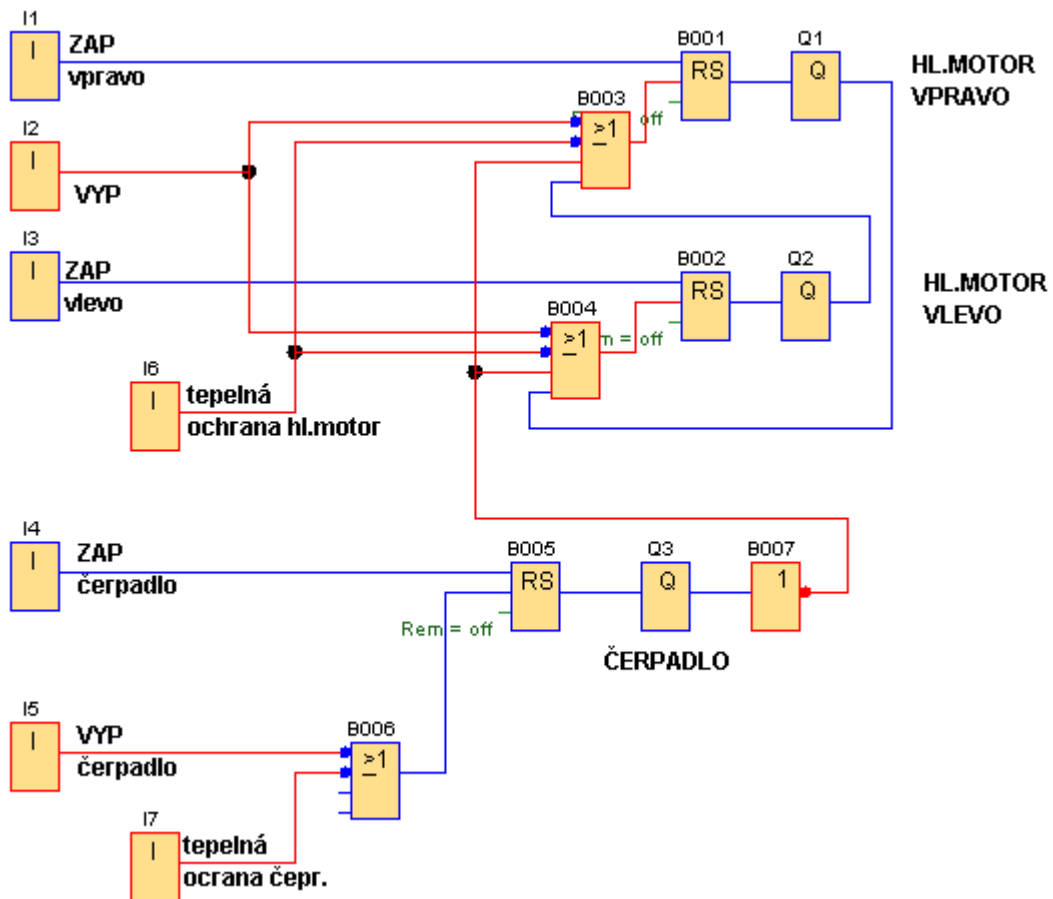
S4 (I4) ZAPNOUT

S5 (I5, v klidu sepnutý kontakt) VYPNOUT

Hlavní motor nesmí běžet bez čerpadla chlazení.

Motory jsou před přetížením chráněny tepelnými ochranami (I6, I7, v klidu sepnuté kontakty).

Při ručním programování vynechte B007 (NOT) a neguňte příslušné vstupy na B003, B004.



*Obr. 62. K příkladu 23*



### 6.1.2. Příklad na procvičení

**Příklad 24.** Hlavní a pomocný motor  
*Reverzace a blokování pro jeden směr.*

Navrhněte a odlaďte pomocí LogoSoftComfort.

Hlavní motor je ovládán třemi tlačítky.

S1 ZAP. směr točení vpravo.

S2 Stop.

S3 ZAP směr točení vlevo.

Oba směry jsou navzájem blokovány.

Pomocný motor je ovládán dvěma tlačítky.

S4 ZAP

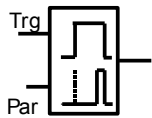
S5 VYP

Z provozních důvodů je nutné, aby běh pomocného motoru byl blokován pro běh hl. motoru ve směru vpravo.

Pokud byl běh pomocného motoru ukončen z důvodů blokování, jeho další spuštění je možné pouze příslušným zapínacím tlačítkem.

Motory jsou před přetížením chráněné tepelnými ochranami.

## 6.2. Blok zpožděného zapnutí



Výstup bloku zpožděné zapnutí sepne když vyprší nastavený čas, ale po celou dobu musí být  $Trg = 1$ .

Obr. 63

- Vstup Trg: Náběžná hrana signálu na tomto vstupu spustí čas zpoždění.
- Parametr T: Čas, po jehož vypršení bude výstup bloku sepnut.  
*Hodnota parametru T může být také nastaven na základě skutečné hodnoty jiné, již konfigurované funkce. Úplný seznam řídicích (master) funkcí je v [1] str.75. Mezi master funkce patří čítač. Způsob nastavení dvojice master – slave ručně i softwarově, viz část 8.1.1. čítač dopředný a zpětný.*
  - T - nastavený čas.
  - Ta - aktuální čas - zadávání parametru viz část 4.4.
- Výstup bloku: je po uplynutí nastaveného času sepnut, pokud je ovšem na Trg signál 1.

### Popis funkce.

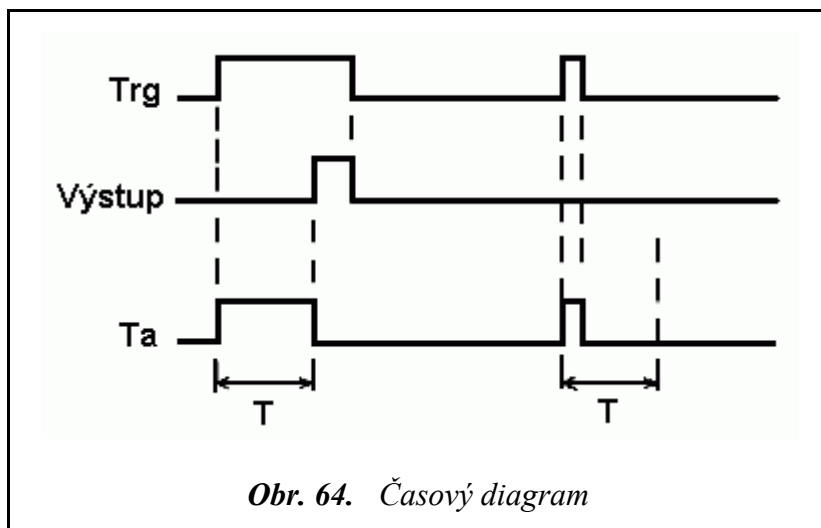
Při přechodu Trg z 0 do 1 se spustí nastavený čas.

Změní-li se signál na vstupu Trg před uběhnutím doby zpoždění opět na 0, bude čas vynulován.

Je-li signál na vstupu Trg dostatečně dlouho na 1, bude po uplynutí doby zpoždění výstup bloku sepnut.

Výstup je vypnut, změní-li se signál na vstupu Trg na 0.

Při výpadku napájení bude čas vynulován.



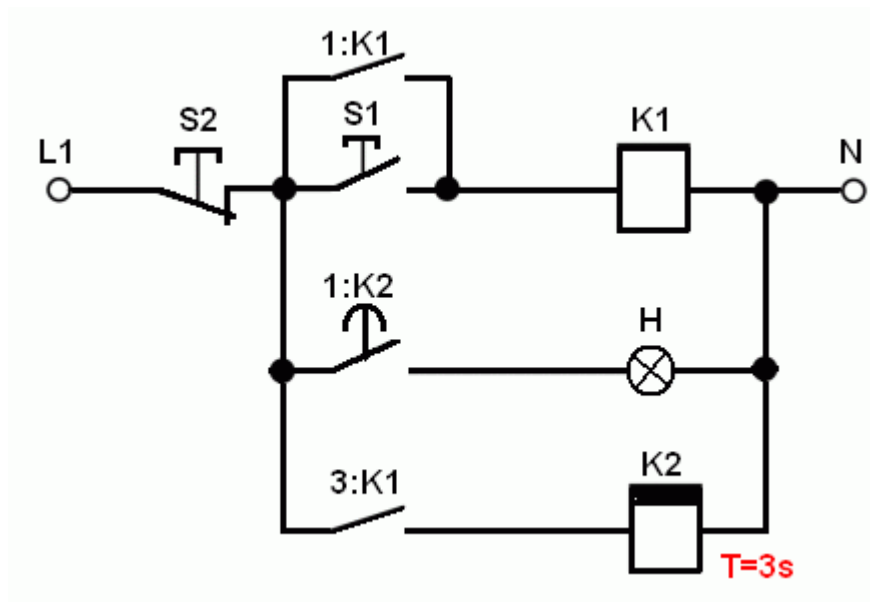
Obr. 64. Časový diagram

### Příklad.

Výstup LOGO! je spínaný a vypínaný pomocí tlačítek na vstupech I1 a I2.

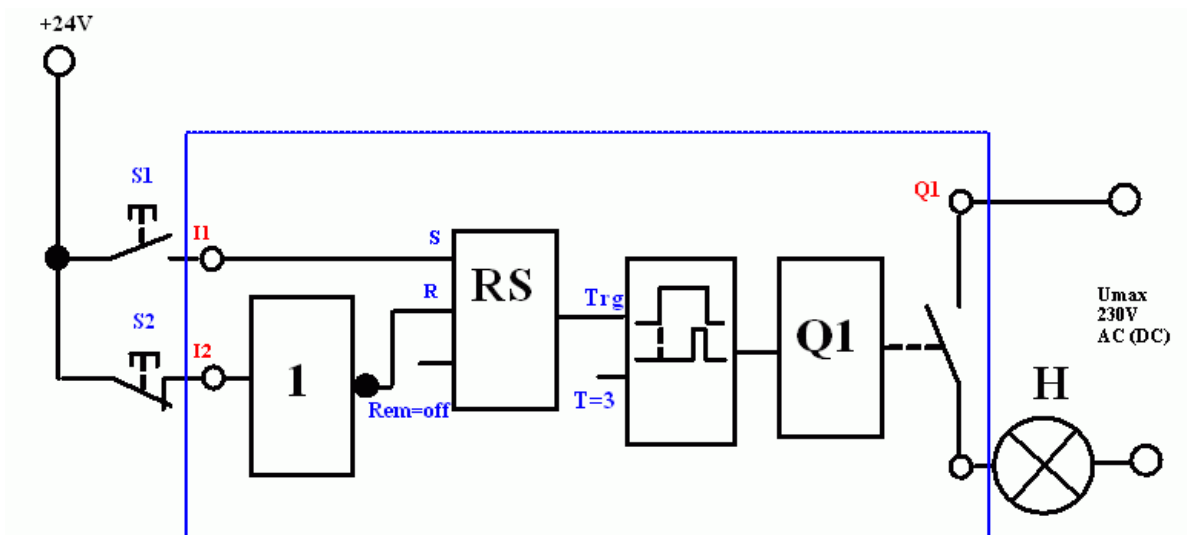
Výstup spíná se zpožděním 3 sekundy a vypíná bez zpoždění.

Blok zpožděné zapnutí můžeme přirovnat k relé se zpožděním při zapnutí.



Obr. 65. Relé se zpožděním při zapnutí

Na obr. 66 je vyřešený předchozí příklad pomocí programu v LOGO! Jsou zde znázorněny i vnější spoje připojené na vstupech a výstupu modulu. Abychom mohli k ovládání výstupu použít tlačítka, předřadili jsme blok RS. Vypínací tlačítko je z ohledem na větší bezpečnost v klidu sepnuté.



Obr. 66. Vnější spoje a blokové schéma v LOGO!

## 6.2.1. Řešený příklad

**Příklad 25.** Postupné sepnutí více výstupů a chybová hlášení

Navrhněte a odlaďte pomocí LogoSoftComfort:

Tlačítko ve funkci ZAP připojené na vstup I1.

Tlačítko ve funkci VYP (kontakty v klidu sepnuté) připojené na vstup I2.

Po stisknutí tlačítka ZAP.

Q1 sepne ihned.

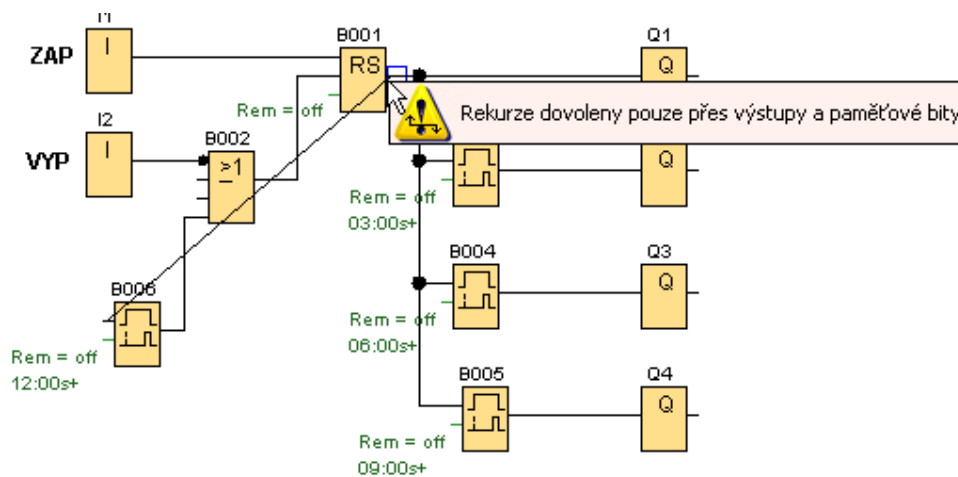
Q2 sepne se zpožděním 3 sekundy.

Q3 sepne se zpožděním 6 sekund.

Q4 sepne se zpožděním 9 sekund.

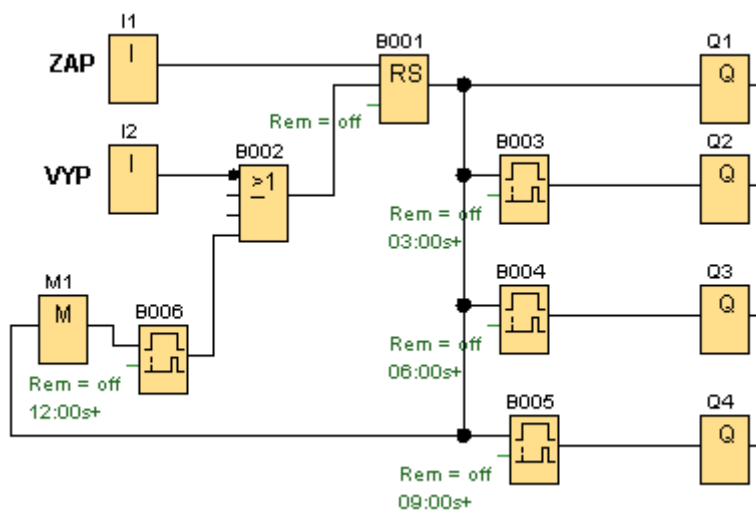
Výstupy se automaticky vypnou 12 sekund po sepnutí Q1.

Na obr. 67 až 69 je naznačen jeden způsob řešení i s ukázkou chybového hlášení.



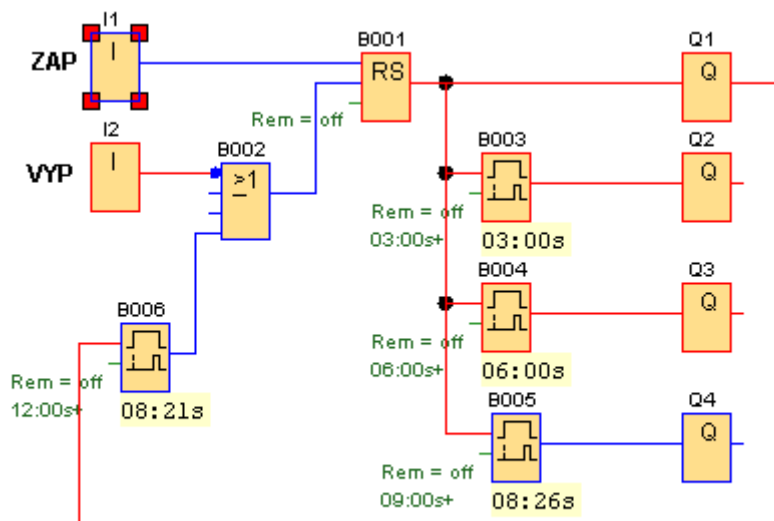
**Obr. 67.** K příkladu 25, chybové hlášení

Řešení rekurze pomocí paměťového bitu M.



*Obr. 68. K příkladu 25 řešení rekurze pomocí paměťového bitu*

Řešení rekurze pomocí konektoru Q.



*Obr. 69. K příkladu 25 řešení rekurze pomocí konektoru Q*

## 6.2.2. Příklady na procvičení

**Příklad 26.** Hlavní motor a kompresor

*Reverzace a rozběh se zpožděním.*

Navrhněte a odlaďte pomocí LogoSoftComfort.

Hlavní motor a motor kompresoru vzduchu jsou ovládány třemi tlačítky.

S1..... ZAP hlavní motor vpravo a ZAP kompresor vzduchu.

S2 .....STOP hlavní motor a kompresor.

S3 .....ZAP hlavní motor vlevo a ZAP kompresor vzduchu.

Směry otáčení hlavního motoru jsou navzájem blokovány.

Z provozních důvodů je nutné, aby se kompresor rozbíhal ihned a hlavní motor se zpožděním 3 sekundy.

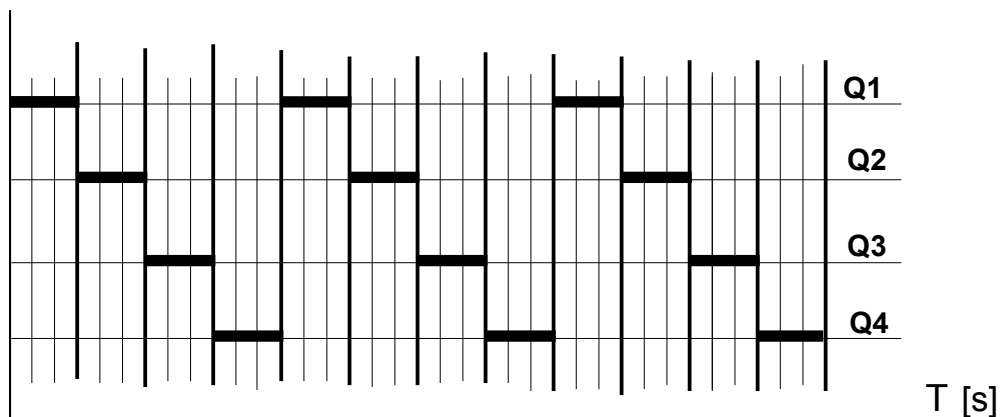
Vypnutí obou motorů je bez zpoždění.

Motory jsou před přetížením chráněné tepelnými ochranami.

**Příklad 27.** Cyklus

Signál 1 na I1 cyklus ZAP.

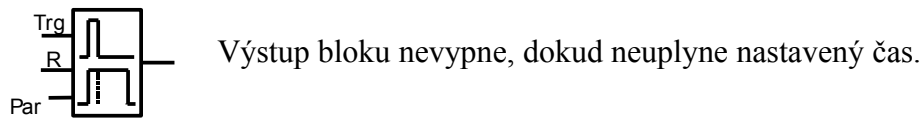
Signál 1 na I2 cyklus VYP.



*Obr. 70. K příkladu 27*

Po signálu na I1 je každý výstup postupně sepnutý 3 sekundy (obr. 70). Časy pro jednotlivé výstupy je možné v režimu přiřazení parametrů měnit. Signál na I2 vypne cyklus kdykoliv.

### 6.3. Blok zpožděného vypnutí



Obr. 71

- Vstup Trg: náběžná hrana signálu sepne výstup bloku a sestupná hrana signálu spustí čas zpožděného vypnutí.
- Vstup R: signál na vstupu vypne výstup bloku a resetuje čas.
- Parametr T: je nastavený čas, po jehož uplynutí výstup rozezne.

T - nastavený čas

T<sub>a</sub> - aktuální čas

Čas parametru T může být také nastaven na základě skutečné hodnoty jiné, již konfigurované funkce. Úplný seznam řídicích (master) funkcí je v [1] str.75. Mezi master funkce patří čítač. Způsob nastavení dvojice master – slave ručně i softwarově, viz část 8.1.1. čítač dopředný a zpětný.

- Výstup bloku je spínán vstupem Trg. Výstup je sepnutý do té doby, dokud neuplyne nastavený čas.

#### Popis funkce.

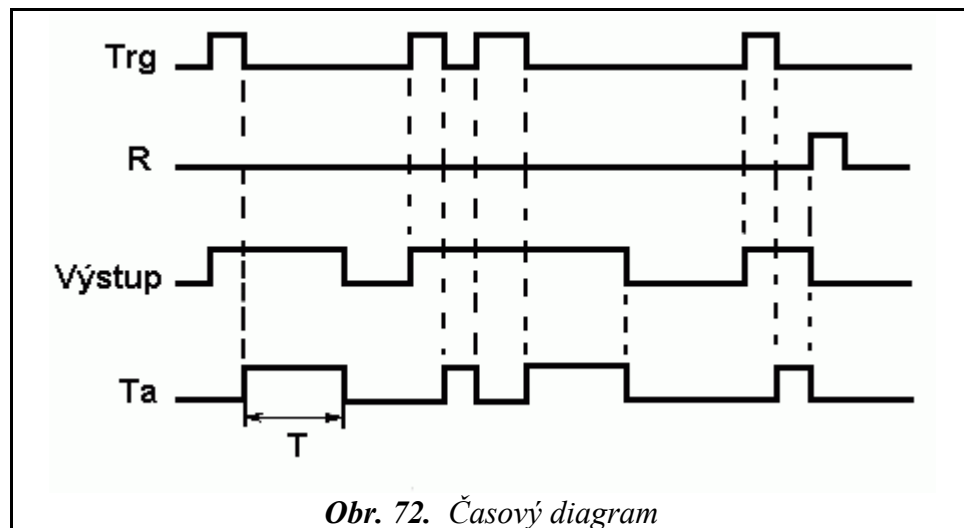
Náběžná hrana signálu na vstupu Trg sepne výstup bloku.

Sestupná hrana signálu na vstupu Trg odstartuje nastavený čas, výstup zůstane sepnutý.

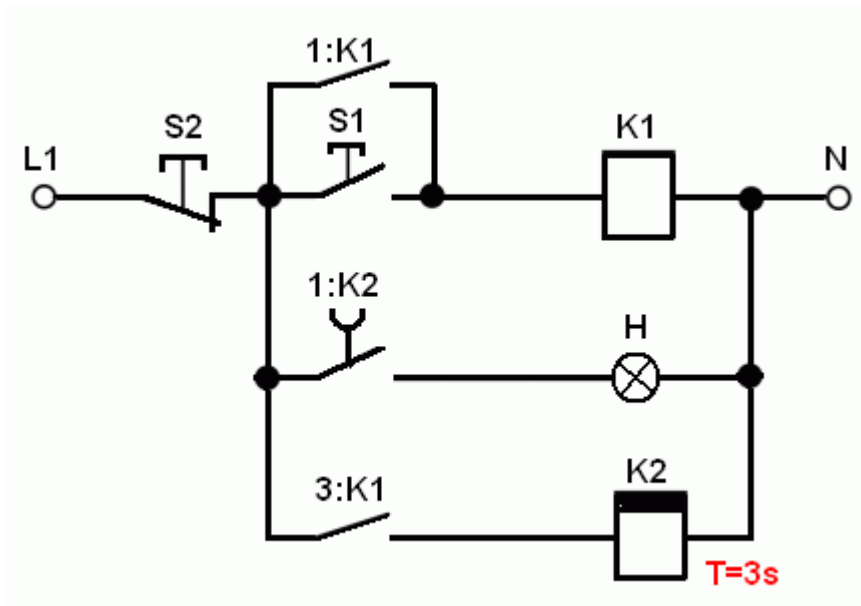
Dosáhne-li aktuální čas hodnotu nastaveného času, výstup bloku vypne.

Signál na vstupu R nuluje aktuální čas a výstup se vypne bez časového zpoždění.

Při výpadku napájení je čas nulován.



Blok zpožděného vypnutí můžeme přirovnat k relé se zpožděním při vypnutí.



*Obr. 73. Relé se zpožděním při vypnutí*

**Příklad.**

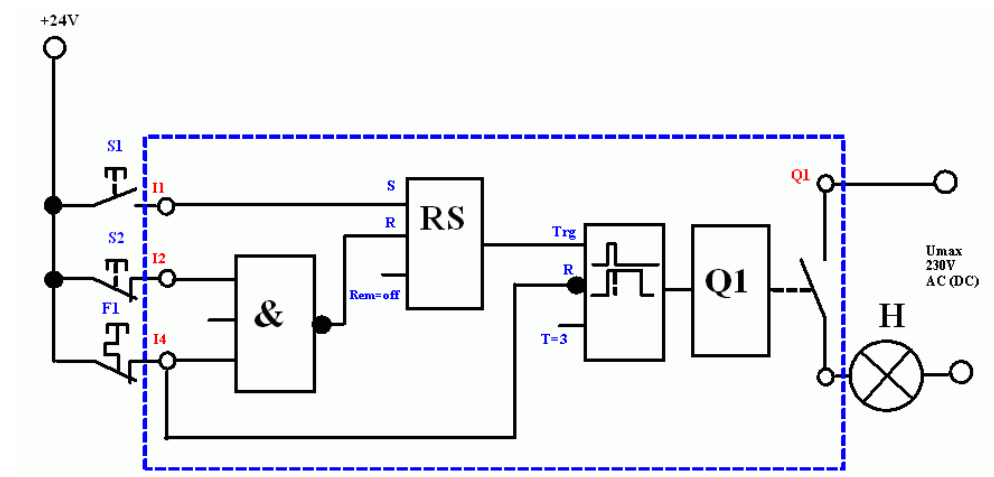
Výstup LOGO! je spínáný a vypínáný pomocí tlačítek na vstupech I1 a I2.

Signál na vstupu I1 sepne výstup Q1 bez zpoždění.

Signál na vstupu I2 vypne výstup Q1 se zpožděním 3 sekundy.

Tepelné jističí relé (I4) vypíná výstup Q1 bez zpoždění.

Na obr. 74 je vyřešený předchozí příklad pomocí programu v LOGO! Abychom mohli k ovládání výstupu použít tlačítka, předradili jsme blok RS. Tlačítko S2 a kontakt tepelného relé jsou v klidu sepnuté.



*Obr. 74. Vnější spoje a blokové schéma v LOGO!*



## 6.4. Příklady na procvičení

**Příklad 28.** Spínání výstupu se zpožděním při sepnutí a při vypnutí  
Ovládání motoru je dvěma tlačítky.

S1 ... ZAP.

S2 ... VYP.

Motor je chráněný před přetížením tepelným relé.

Motor se rozběhne se zpožděním 3 sekundy.

Motor zastaví se zpožděním 4 sekundy.

Pokud jde do funkce tepelná ochrana, motor se vypne ihned.

**Příklad 29.** Ovládání dvou dopravníků

S1 ... ZAP.

S2 ... VYP.

S4... TOTAL STOP.

První dopravník se rozbíhá se zpožděním 4 sekundy a vypíná bez zpoždění.

Druhý dopravník se rozbíhá bez zpoždění a vypíná se zpožděním 4 sekundy.

TOTAL STOP vypíná oba dopravníky bez zpoždění.

**Příklad 30.** Garážová vrata

Motor pro spouštění a vytahování garážových vrat spínají stykače.

Ovládání je pomocí tlačítek.

Oba směry jsou navzájem blokovány.<sup>13</sup>

Pohyb vrat NAHORU ovládají tlačítka ZAP a VYP.

Pohyb vrat DOLŮ ovládá jedno tlačítko pracující jako impulzovací<sup>14</sup>.

Krajní polohy vrat vymezují koncové spínače.

Na spodní hraně vrat je ochranná lišta. Po dobu její aktivace je blokován pohyb vrat DOLŮ.

Motor je chráněn před přetížením tepelným relé.

---

13 Zapojení nedovolující spustit motor běžící už ve směru NAHORU zároveň ve směru DOLŮ a naopak.

14 Motor běží pouze po dobu stisknutí tlačítka.

**Příklad 31.** Rozběh Y/D

S1...ZAP.

S2...VYP.

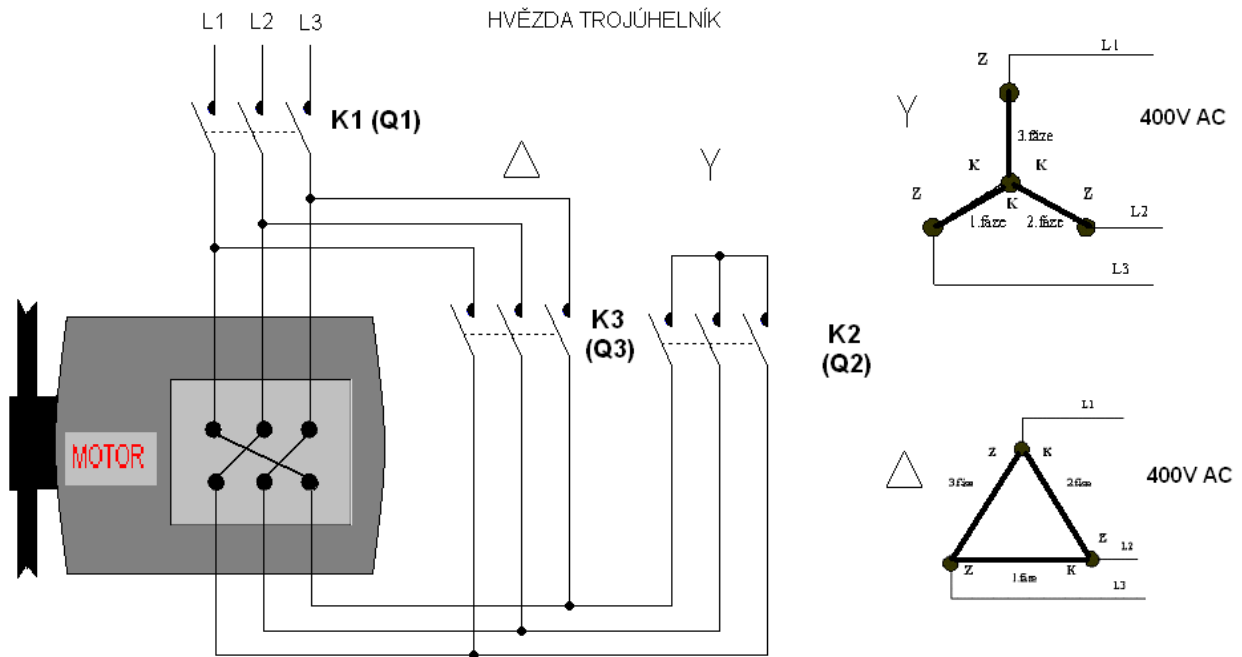
Výstup Q2 (hvězda) sepne okamžitě po stisknutí S1 a vypíná automaticky za 7 sekund.

Výstup Q1 (přívod na motor) se spíná 1 sekundu po sepnutí Q2.

Výstup Q3 (trojúhelník) sepne 1 sekundu po rozepnutí Q2.

Poznámka:

Po odladění změňte časy pro Q1 a Q3 na  $T = 0,05$  sekundy.



*Obr. 75. Silová část zapojení Y/D*

### **Příklad 32.** Pásová doprava

Tři pásové dopravníky se mají postupně rozbíhat a zastavovat se zpožděním, jeden po druhém. Předpokládají se tři druhy provozu:

- 1) Postupný rozběh START.
- 2) Postupné zastavování STOP s doběhem.
- 3) Rychlé zastavení TOTAL STOP.

Mají být sledovány stavy motorů a pokud některý z nich uvede svým působením do činnosti tepelné jistící relé, má se soustava dopravníků postupně zastavit.

Dojde-li k poruše, má být ohlášena blikajícím světlem. Systém nelze spustit do odstranění poruchy.

#### Popis funkce:

Tři pásové dopravníky jednoho transportního zařízení se mají postupně rozbíhat a zastavovat, aby byla zajištěna hladká a bezporuchová doprava materiálu.

#### Rozběh:

Po stisknutí tlačítka START nabíhají pásy v pevně naprogramovaných intervalech po 5 s. Jako první se rozbíhá pás č.3.

#### Zastavení:

Po stisknutí tlačítka STOP s doběhem se pásy začnou zastavovat v opačném pořadí než při rozběhu, počínaje pásem č.1 *Tím se zajistí, že budou dobíhat prázdné pásy a že při příštím spuštění se budou rozbíhat pásy odlehčené, prázdné.*

Mezi stisknutím tlačítka a zastavením pásu č.1 uplyne 5 s.

Následující pásy se budou zastavovat rovněž po 5 s. Nastavené časové intervaly má být možné měnit obsluhou pásů pomocí relé LOGO!.

Tlačítkem TOTAL STOP se vypínají motory všech tří pásů okamžitě, bez zpoždění.

#### Porucha motoru:

Jestliže se některý motor porouchá, odpojí se od sítě.

Zjištěná porucha se ohlásí blikavým světlem a samočinně vyvolá funkci STOP.

To v tomto případě znamená, že motory následující za porouchaným se budou vypínat postupně v 5 s. intervalech zatímco motory před porouchaným budou vypnuty okamžitě.

### **Příklad 33.** Odstředivka

Ovládání dvoustupňové odstředivky je pomocí tlačítek

Tlačítkem PRVNÍ STUPEŇ se spustí NÍZKÉ OTÁČKY

Tlačítkem DRUHÝ STUPEŇ se spustí VYSOKÉ OTÁČKY.

Tlačítkem STOP se vypne zařízení kdykoliv.

Oba stupně odstředivky nesmí běžet současně.

VYSOKÉ OTÁČKY nesmí běžet jako první.

Mimo první čtyři sekundy může obsluha přepnout na vysoké otáčky ručně kdykoliv

**Příklad 34.** Postupné spouštění

Navrhněte obvod.

Zapínací tlačítko S1 (I1), vypínací tlačítko S2 (I2).

Po stisku tlačítka S1 sepne výstup Q1.

Po uvolnění tlačítka S1 sepne výstup Q2.

Dalším stisknutím a uvolněním tlačítka S1 sepne Q3 a Q4.

**Příklad 35.** Semafor

Navrhněte zapojení podle obr. 76.

**Noční režim**

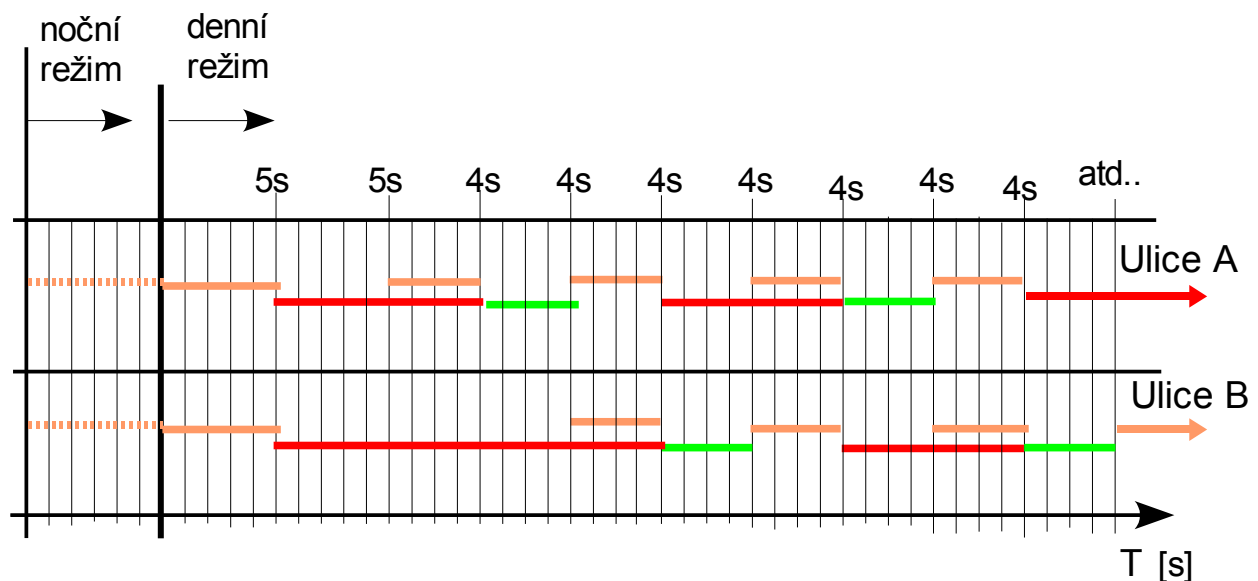
Se spustí automaticky po zapnutí LOGO! nebo tlačítkem NOČNÍ REŽIM kdykoliv v průběhu denního režimu.

Oranžová pro ulice A a B kmitá v rytmu 1 sekundy.

Pro tuto funkci použijte blok *Asynchronní pulzní generátor* ze seznamu SF. Funkci bloku vyhledejte v nápovědě LogoSoftComfort.

**Denní režim** se spouští tlačítkem DENNÍ REŽIM.

Při přechodu z nočního na denní režim se pro uklidnění situace rozsvítí klidným světlem pro A i B na 5 sekund oranžová a červená, další spínání výstupů je po 4 sekundách a jejich násobcích, jak je naznačeno na obr. 76.



Obr. 76. K příkladu 35

### Příklad 36. Pohyb stolu brusky

Stroj se spouští pouze zapínacím tlačítkem. Po stisknutí tlačítka ZAP vykonává pracovní stůl rovinné brusky stálý pohyb mezi koncovými spínači.

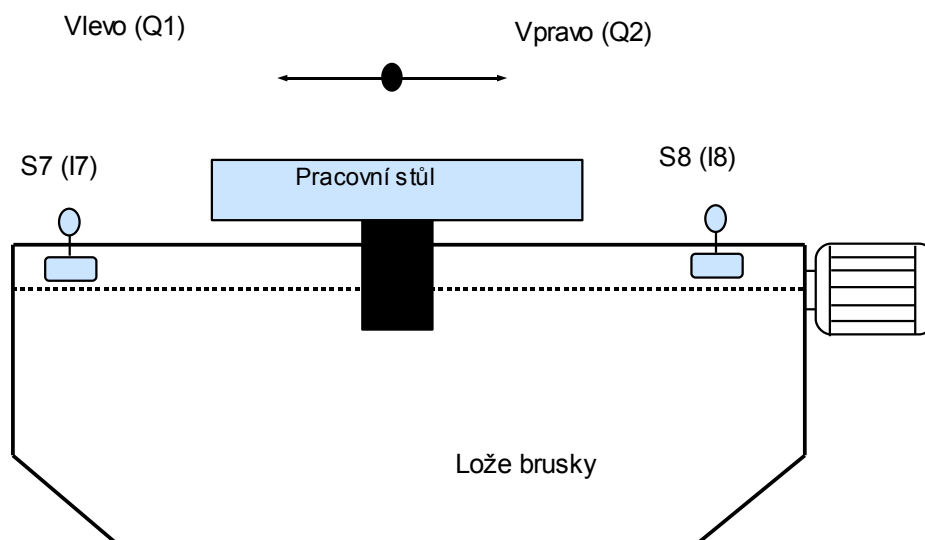
Při návrhu zohledněte následující body.

1. **Před spuštěním stojí stůl brusky mezi koncovými spínači  $I7=0$  a  $I8=0$ .** Stůl brusky se při spuštění začne pohybovat nejprve vlevo (Q1) až do polohy dané koncovým spínačem S7 (I7) a potom vpravo (Q2), až do polohy dané pravým koncovým spínačem S8 (I8).
2. **Před spuštěním stojí stůl brusky v jedné z krajních poloh  $I7=1$  nebo  $I8=1$ .** Stůl brusky se při spuštění začne pohybovat ve volném směru.
3. **Oba koncové spínače jsou sepnuté  $I7=1$  a  $I8=1$ . Jde o PORUCHOVÝ STAV.** Stroj nelze spustit a běží-li, zastaví se. Po odstranění poruchy je spuštění možné pouze tlačítkem ZAP.

Po stisknutí tlačítka VYP se ukončí pohyb stolu.

Motor je jištěný před přetížením tepelnou ochranou.

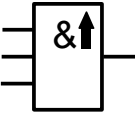
Pohon brusného kotouče není součástí řešení.



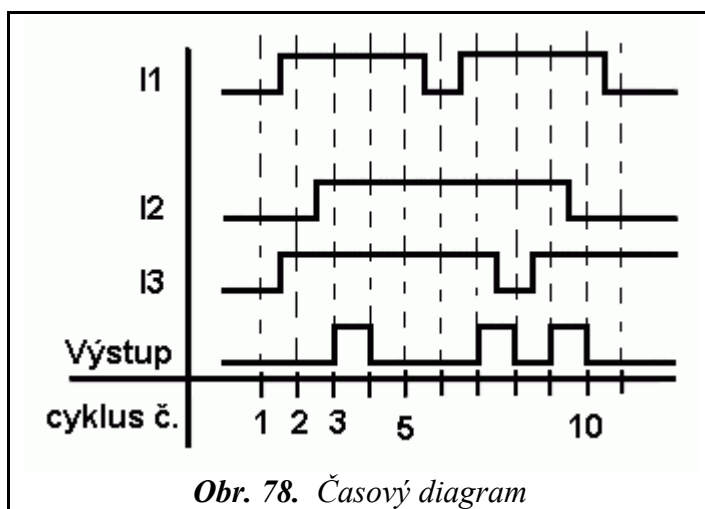
*Obr.77. K příkladu 36*

## 7. Základní funkce 2. část

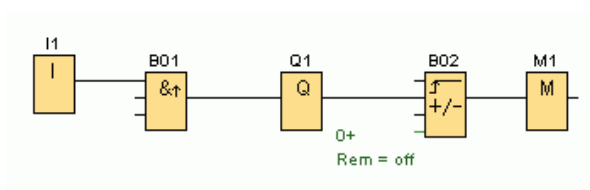
### 7.1. AND s detekcí náběžné hrany

Zobrazení v LOGO!	Popis základní funkce
	<p>Výstup nabývá stavu 1 jen tehdy, jsou-li všechny vstupy ve stavu 1 a nejméně jeden měl v předchozím programovém cyklu stav 0. Nepřipojený vstup hradla je nastaven automaticky na 1.</p>

Tab. 18. AND s detekcí náběžné hrany



Z diagramu na obr. 78 je patrné, že pro splnění podmínky sepnutí stačí, aby byl vstup ve stavu 1 či 0 i jen v části cyklu.



Obr. 79. AND s náběžnou hranou

K obr. 79.

Při stisknutí tlačítka na vstupu I1 sepne B01 na 1 programový cyklus. Pro větší názornost je na Q1 připojen čítač B02. Nemá jinou funkci, než načítat a zobrazit každé sepnutí Q1.

*Paměťový bit, nebo virtuální výstup na výstupu B02 je nutný pro korektní přenos z PC do LOGO!*

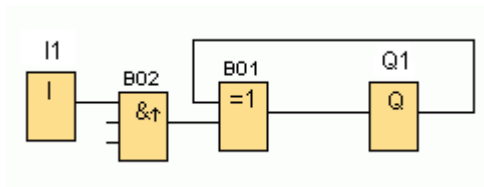
### 7.1.1. Řešený příklad

**Příklad 37.** Varianty ovládání výstupu jedním tlačítkem

Tlačítkem na vstupu I1 střídatě spínáte a vypínáte výstup.

Navržené řešení na obr. 80 vychází z funkce XOR. *Funkce XOR = 1, jsou-li vstupy různé.*

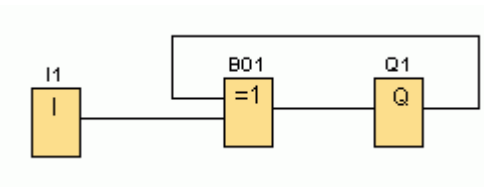
AND s náběžnou hranou sepne na dobu trvání 1 programového cyklu, takže spínání a rozpínání výstupu není ovlivňováno délkou signálu na I1. Porovnejte s funkcí podle obr. 81.



**Obr. 80.** K příkladu 37

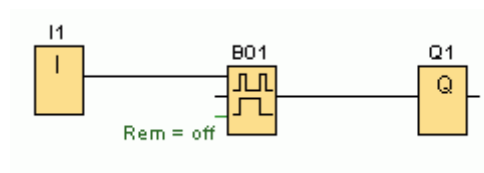
Na obr.81 je ukázka obvodu bez B02.

Výstup občas sepne správně, ale častěji se rozkmitá, protože funkce programu je velice velmi závislá na délce signálu na I1. Bez bloku B02, je tento způsob řešení nepoužitelný.



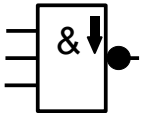
**Obr. 81.** K příkladu 37

Na obr. 82 je řešení s použitím bloku pulzní proudové relé.

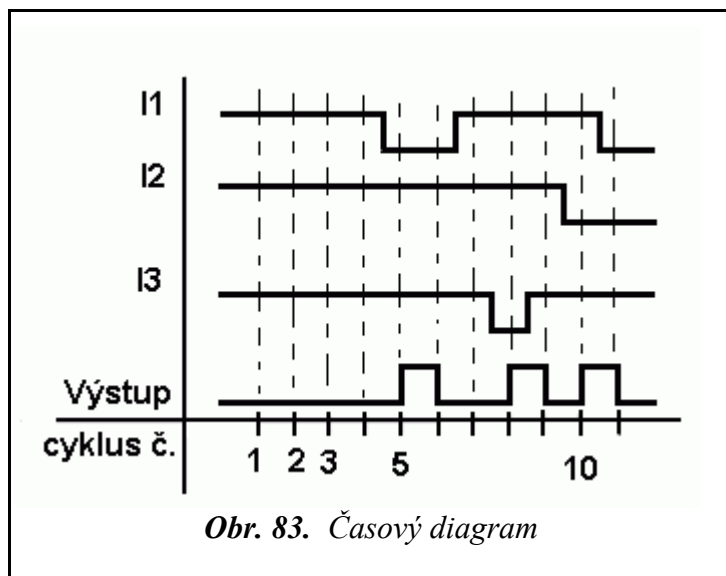


**Obr. 82.** K příkladu 37

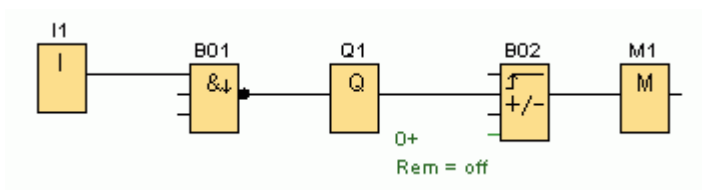
## 7.2. NAND s detekcí sestupné hrany

Zobrazení v LOGO!	Popis základní funkce
	Stav NAND s vyhodnocením hrany je 1 pouze pokud alespoň jeden vstup je 0 a všechny vstupy byly v předchozím cyklu 1. Nepřipojený vstup hradla je nastaven automaticky na 1.

Tab. 19. NAND s detekcí sestupné hrany



Z diagramu na obr. 83 je patrné, že pro splnění podmínky sepnutí stačí, aby byl vstup ve stavu 1 či 0 i jen část cyklu.



Obr. 84. NAND se sestupnou hranou

K obr. 84

Při uvolnění sepnutého tlačítka na vstupu I1 sepnou B01 na 1 programový cyklus. Pro větší názornost je na Q1 připojen čítač B02. Nemá jinou funkci, než načítat a zobrazit každé sepnutí Q1. *Paměťový bit, nebo virtuální výstup na výstupu B02 je nutný pro korektní přenos z PC do LOGO!*



## 7.2.1. Řešené příklady

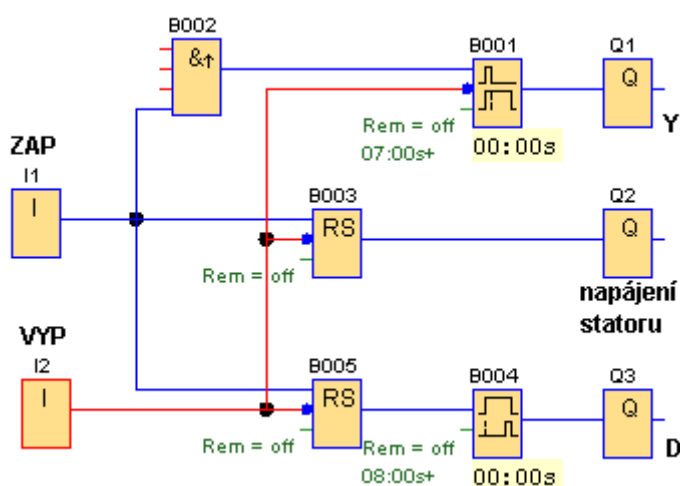
### Příklad 38. Spouštění YD

Pro snížení proudového nárazu je motor spouštěný přepínáním vinutí YD.

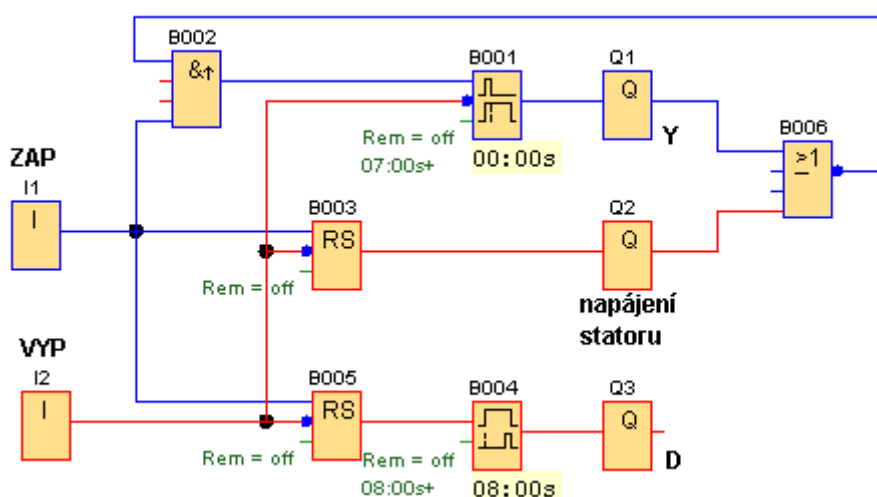
Ovládání motoru je tlačítka a stykači. Rozběh motoru trvá 7 sekund. Mezi přepnutím z Y na D bude časová prodleva 0,05sec. Vytvořte program pro LOGO!

K obr. 85. Při ladění programu byla nastavena prodleva přepnutí 1 sec.

Pokud bychom nepoužili B02 a spouštěli blok B01 sestupnou hranou signálu (tedy až po uvolnění tlačítka na vstupu I1), byl by čas B01 ovlivněn délkou stisknutí (respektive uvolnění) tlačítka na I1. V určitém případě by mohlo dojít k současnému sepnutí Q1, Q3. To by v tomto návrhu způsobilo mezifázový zkrat v silové části obvodu. Ze stejného důvodu musíme ihned po spuštění programu zablokovat spouštěcí tlačítko (viz obr. 86).



Obr. 85. K příkladu 38



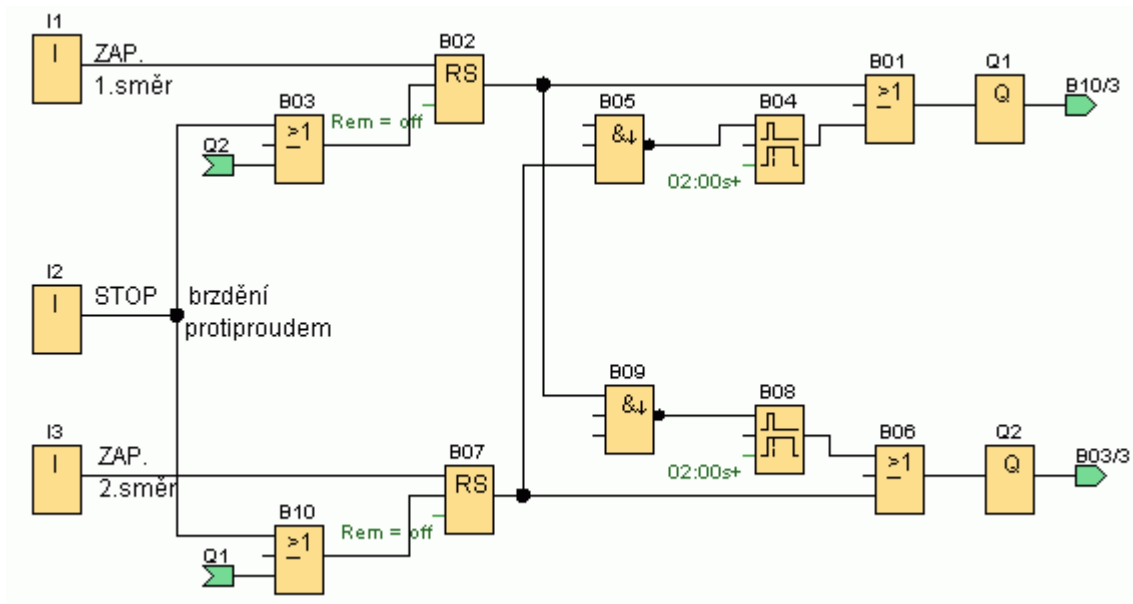
Obr. 86. K příkladu 38, rozběh ukončen, konektor I1 zablokován

V zapojení na obr. 86 nahraďte bloky B004, B005 jedním blokem zpožděného zapnutí s pamětí.

Funkce programu se tím nezmění.

Po odladění nastavte prodlevu mezi přepnutím z Y na D na 0,05sec.

**Příklad 38.** Reverzace otáček motoru a brzdění protiproudem  
Ovládání motoru je provedeno tlačítky.  
Otáčky motoru jsou reverzovány.  
Při vypnutí je motor brzděn 2 sekundy protiproudem.



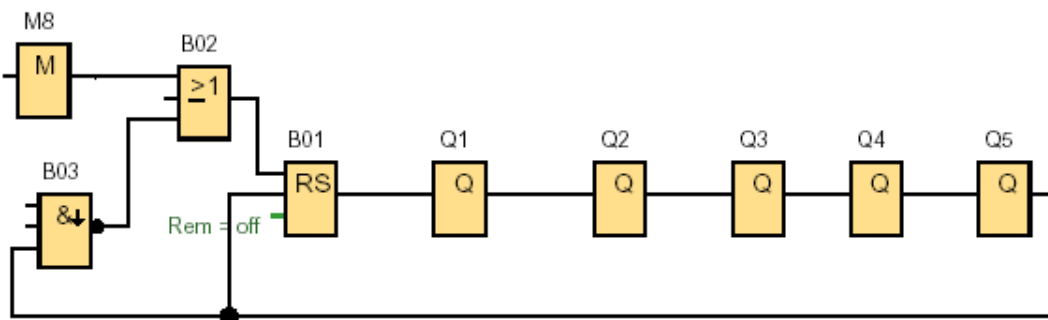
*Obr. 87. K příkladu 38*

Doplňte o tepelné jističí relé motoru. STOP tlačítko i kontakt tepelného relé bude v klidu sepnutý. Upravte zapojení na obr. 87.  
V parametrickém režimu nastavte čas brzdění podle skutečného motoru, aby nedošlo k přikývnutí rotoru.

### 7.2.2. Příklad na procvičení

#### **Příklad 39.**

Popište funkci programu na obr. 88

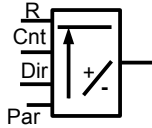


*Obr. 88. K příkladu 39*

Příklad na obr. 88 upravte tak, aby po spuštění programu jednotlivé výstupy spínaly s prodlevou 1,5 sekundy a vypínaly s prodlevou 2 sekundy.

## 8. Speciální funkce 2. část

### 8.1. Dopředný a zpětný čítač



Obr. 89

Podle nastavení, je při každém impulsu na vstupu Cnt zvětšena / zmenšena vnitřní hodnota čítače. Výstup sepne / vypne když je dosaženo nastavené prahové hodnoty čítače.

*Limity pro parametry zapnutí / vypnutí mohou být také nastaveny na základě skutečné hodnoty jiné, již konfigurované funkce. Úplný seznam řídicích (master) funkcí je v [1] str.75. Mezi master funkce patří čítač. Způsob nastavení dvojice master – slave ručně i softwarově, viz část 8.1.1.*

- Vstup R: nuluje vnitřní hodnotu čítače a výstupu.
- Vstup Cnt:
  - náběžné hrany signálu<sup>15</sup> na vstupu Cnt jsou čítány.
  - sestupné hrany nejsou čítány.

Maximální čítací frekvence na normálních digitálních vstupech vstupech LOGO! (I1-I4 a I7,I8) je 4Hz.

Maximální čítací frekvence na rychlých digitálních vstupech LOGO! (I5, I6) je 2kHz. Více o problematice rychlých vstupů v [1] poznámka v části 4.4.13.

- Vstup Dir: Se používá pro nastavení směru počítání.
  - Dir = 0: Přičítání impulsů.
  - Dir = 1: Odčítání impulsů.
- Parametr Par:
  - Prahová hodnota zapnutí: 0 - 999 999
  - Prahová hodnota vypnutí: 0 - 999 999
  - Rem: remanence vnitřní hodnoty čítače
    - / remanence zakázaná (**výchozí stav**)
    - R remanence povolena

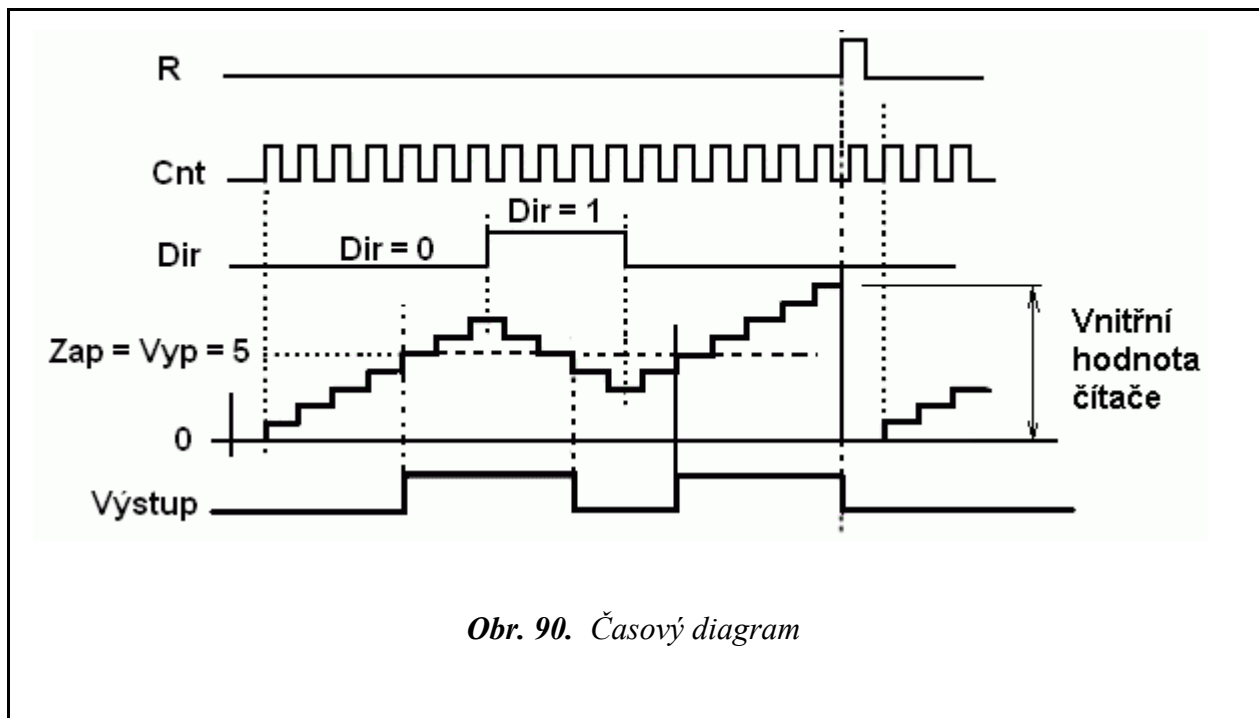
Popis funkce:

Při každé náběžné hraně na vstupu Cnt je čítač zvýšen o jedničku (když Dir = 0), nebo snížen o jedničku (když Dir = 1). Výstup je zapnut, nebo vypnut v závislosti na aktuální hodnotě čítače a nastavení prahových hodnot.

Vstup R použijeme pro resetování (nulování) výstupu a vnitřní hodnoty čítače na 0. Pokud je vstup R = 1, je výstup 0 a náběžné hrany na Cnt nejsou počítány.

---

15 Přechody ze stavu 0 do 1



**Obr. 90.** Časový diagram

#### **Příklad.**

Pro příklady v části 8.1. platí, že stisknutím tlačítka na příslušném vstupu LOGO! je na vstup Cnt přiveden signál 1.

Navrhněte obvod a ručně naprogramujte modul LOGO!.

Výstup Q sepne potom, když 3x stiskneme tlačítko S1.

Výstup Q vypne potom, když 5x stisknete tlačítko S1.

Tlačítkem S2 výstup Q vypneme.

Vypínač S5 použijte pro změnu směru čítání.

Po spuštění programu postupujte podle bodů 1-7.

1. V režimu přiřazení parametrů kontrolujte aktuálně načítaný počet.
2. Pomocí tlačka S1 načítejte do tří.
3. Pomocí tlačítka S1 načítejte do pěti.
4. Pomocí tlačítka S1 načítejte do osmi.
5. Změňte směr čítání vypínačem S5 (Dir=1) a načítejte pomocí tlačítka S1 postupně do dvou.
6. Potom opět načítejte do osmi (Dir=0).
7. Resetujte čítač i výstup (I2=1).

Všimněte si:

Dir=0

Když vnitřní hodnota čítače dosáhne nastavené hodnoty, sepne nebo vypne výstup.

Při dalších impulsích na vstupu Cnt se dále zvyšuje vnitřní hodnota čítače.

Dir = 1

Když vnitřní hodnota čítače dosáhne nastavené hodnoty sepne nebo vypne výstup.

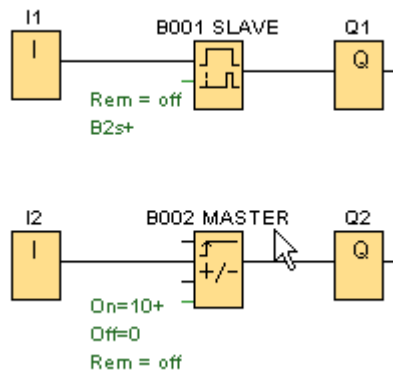
Odčítat jde do nuly.

Pro odčítání musí být Dir=1 a Cnt=1. Přesněji řečeno Dir musí být = 1 v okamžiku, kdy na vstupu Cnt je náběžná hrana signálu.

### 8.1.1. Nastavení parametru jiným, již nakonfigurovaným blokem

Obr. 91

Čas zpožděného zapnutí (slave) je určován momentálním stavem čítače (master). Čítač ve funkci master můžeme v programu použít zároveň i k jiným úkolům.

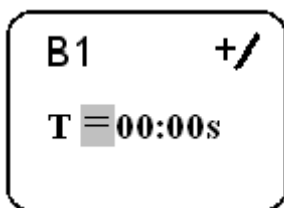


Obr. 91. Master, slave

#### Ruční programování

Nastavení B1 jako slave.

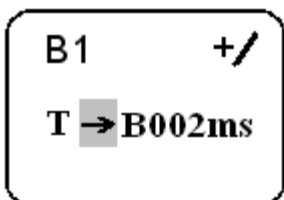
**Důležitá podmínka:** V tomto okamžiku musí být v programu vložený čítač, jinak následující operace nejsou umožněny.



Obr. 92

Při nastavování parametru B1

Přesuneme kurzor na pozici znaku = a stisknutím kurzorové klávesy nahoru nebo dolů změníme rovnítko (obr. 92) na šipku (obr. 93). Časová základna bude nahrazena číslem posledního bloku na který je dán odkaz a jeho časovou základnou. Zde blok B002<sup>16</sup> s časovou základnou ms. Když chceme vybrat jiný blok, posuneme kurzor klávesou kurzor vpravo na písmeno B a kurzorovou klávesou nahoru / dolů vybereme požadovaný blok. Stejným postupem můžeme změnit časovou základnu bloku z milisekund na sekundy, obr. 94.



Obr. 93

16 Číslo B002 je při ručním programování skutečně v tomto tvaru

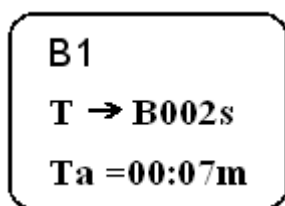
Změníme časovou základnu na sekundy.



*Obr. 94*

Na čítači (B2) bylo načítáno číslo 7, potom u B1 aktuální čas  $T_a = 7$  sekund.

Na obr. 95 je zobrazený displej B1 po doběhnutí času. LOGO! je v režimu přiřazení parametrů.



*Obr. 95*

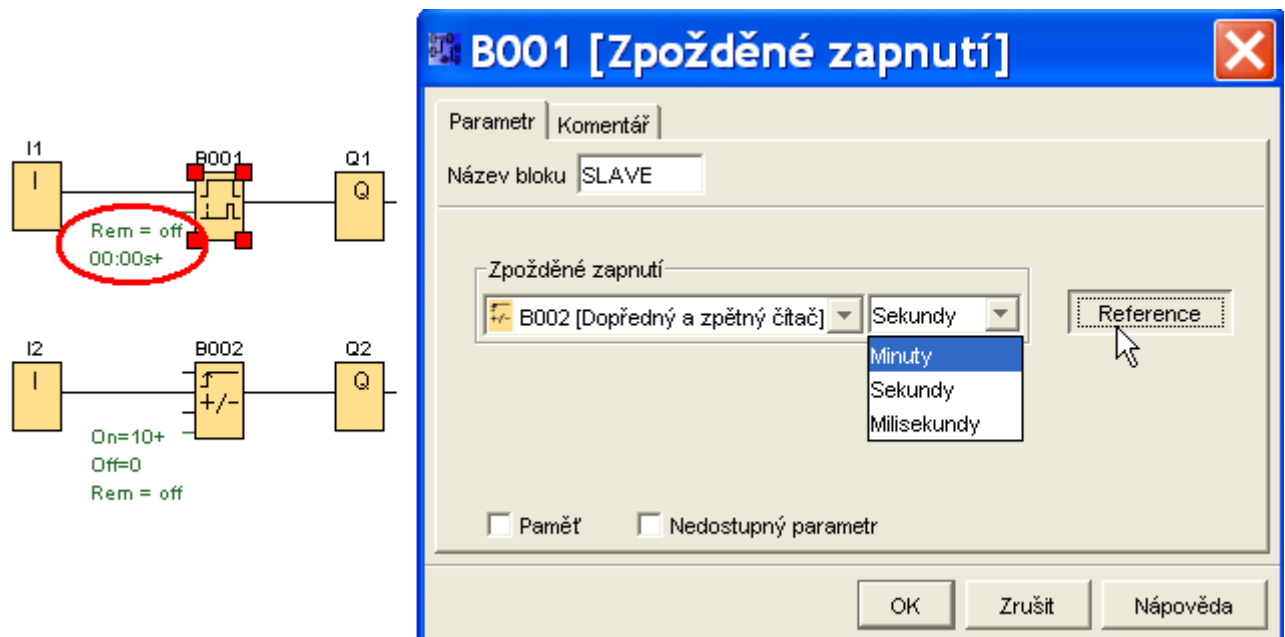
#### Nastavitelná časová základna.

Platné rozsahy časové základny, když  $T$  = skutečná hodnota již naprogramované funkce.

Časová základna	Max. hodnota	Znamená	Přesnost
ms	999990	Počet ms	$\pm 10\text{ms}$
s	5999	Počet s	$\pm 1\text{s}$
m	5999	Počet min	$\pm 1\text{min}$

*Tab. 20. Časová základna pro řízení master, slave*

## Nastavení pomocí software

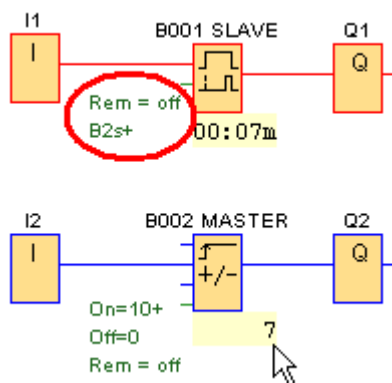


**Obr. 96.** Nastavení referenčního bloku

Všimněme si červeně zakroužkovaných údajů o nastaveném času u B001 před (obr. 96) a po (obr. 97) konfiguraci referenčního bloku.

Na obr. 96: **00:00s+**

Na obr. 97: **B2+**

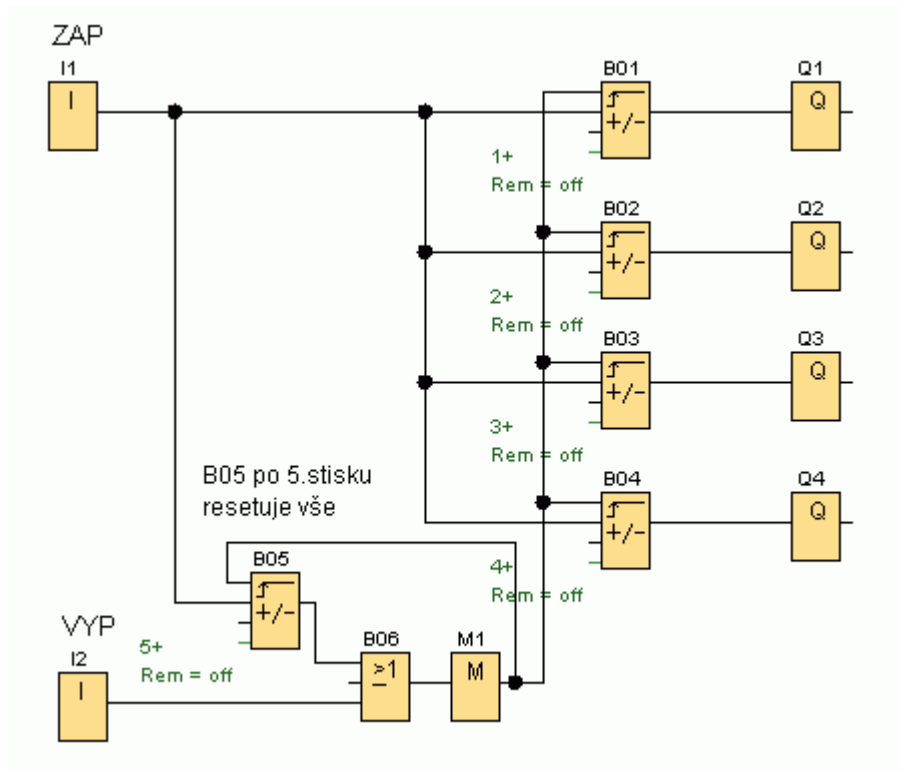


**Obr. 97.** Stav obvodu po sepnutí Q1

### 8.1.1. Řešené příklady

**Příklad 40.** Postupné spínání více výstupů, vypnutí dosaženým počtem

První až čtvrtý signál na I1 sepnou postupně Q1 až Q4. Pátý signál na I1 vypne všechny výstupy a resetuje načítané hodnoty. Následným signálem na I1 začne předchozí cyklus opět od začátku. Signál na I2 vypne výstupy a resetuje čítače kdykoliv.



Obr. 98. K příkladu 40

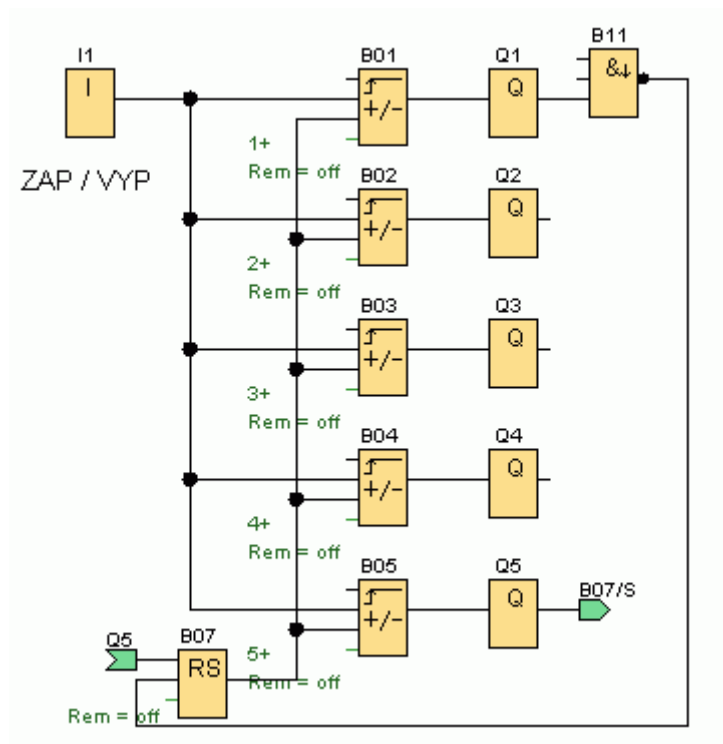
**Příklad 41.** Využití vstupu Dir

Cyklus: ZAP v pořadí Q1-Q5, VYP v pořadí Q5-Q1

Ovládacím tlačítkem S1 (I1) se postupně spínají a vypínají vývody Q1 – Q5.

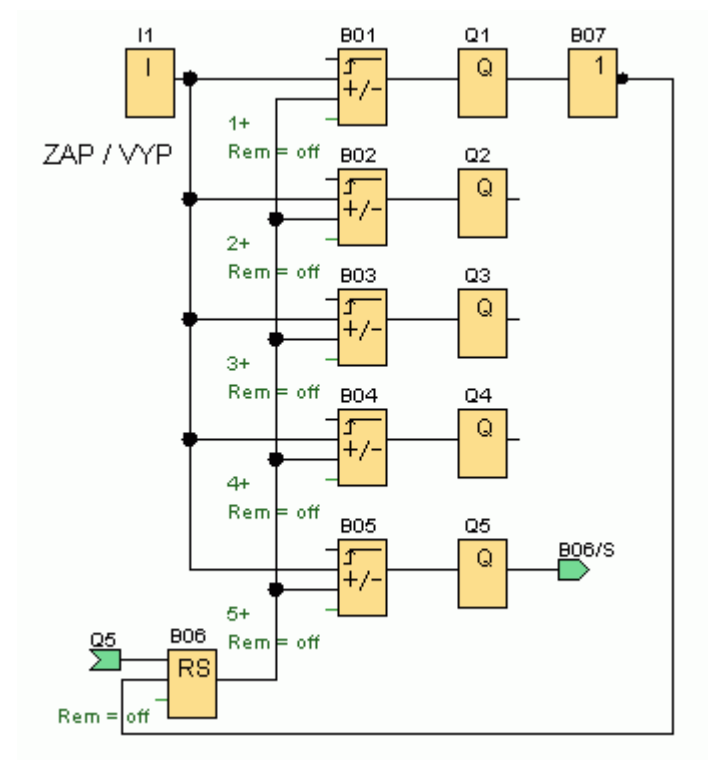
1. signál na I1 sepne Q1.
2. signál na I1 sepne Q2.
3. signál na I1 sepne Q3.
4. signál na I1 sepne Q4.
5. signál na I1 sepne Q5.
6. signál na I1 vypne Q5.
7. signál na I1 vypne Q4.
8. signál na I1 vypne Q3.
9. signál na I1 vypne Q2.
10. signál na I1 vypne Q1.
11. signál na I1 sepne Q1.
12. signál na I1 sepne Q2 ... atd.





Obr. 99. K příkladu 41

Možné řešení pro starší verze LOGO!, které ještě neznají blok NAND s detekcí sestupné hrany.



Obr. 100. K příkladu 41

### 8.1.2. Příklady na procvičení

#### **Příklad 42.** Postupné spínání

Zapínací tlačítko na I1.

Vypínací tlačítko na I2.

První náběžná hrana na I1 sepne výstup Q1.

Třetí náběžná hrana na I1 sepne výstup Q2.

Pátá náběžná hrana na I1 sepne výstup Q3.

Sedmá náběžná hrana na I1 sepne výstup Q4.

*Další náběžná hrany na I1 nemají vliv na stav výstupů.*

Signál na I2 vypíná obvod kdykoliv.

#### **Příklad 43.** Cyklus: Sepnutí a vypnutí výstupu po dosažení určeného počtu

Navrhněte obvod, který splňuje tyto požadavky.

Třetí náběžná hrana na I1 sepne výstup.

Pátá náběžná hrana na I1 vypne výstup.

Další třetí náběžná hrana na I1 sepne výstup ..... atd.

#### **Příklad 44.** Zablkování funkce dosaženým počtem

Navrhněte obvod, který splňuje tyto požadavky.

Třetí náběžná hrana na I1 sepne výstup Q1.

Pátá náběžná hrana na I1 vypne výstup Q1.

Šestá náběžná hrana na I1 zablokuje další ovládání obvodu.

Pro zrušení zablkování použijte vstup I2.

#### **Příklad 45.** Zablkování funkce počtem pracovních cyklů

Navrhněte obvod, který splňuje tyto požadavky:

Po třetím ukončeném pracovním cyklu se automaticky zablokuje další spouštění zařízení Q1.

Zablkování je signalizováno žárovkou na Q2.

Signál na I3 zruší zablkování obvodu.

**Začátek pracovního cyklu.** Signál na I1 sepne Q1.

**Konec pracovního cyklu.** Signál na I2 vypne Q1.

#### **Příklad 46.** Čítání vzestupných i sestupných hran

Navrhněte obvod, který splňuje tyto požadavky.

Zapínací tlačítko na I1.

Vypínací tlačítko na I2.

První náběžná hrana signálu na I1 sepne výstup Q1.

První sestupná hrana signálu na I1 sepne výstup Q2 .

Druhá náběžná hrana signálu na I1 sepne výstup Q3.

Druhá sestupná hrana signálu na I1 sepne výstup Q4..

Další sestupná / náběžná hrana na I1 nemá vliv na stav vývodů Q1 – Q4.

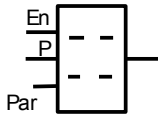
Signál na I2 vypne obvod.

#### **Příklad 47.** Jen určené pořadí a počet sepne výstup

Výstup sepne, když stisknutím tlačítka na vstupu I1 přivedeme signál jedenkrát na I1 a stisknutím tlačítka na vstupu I2 přivedeme signál jedenkrát na I2. V tomto pořadí a počtu.

Všechny ostatní možnosti (počet nebo pořadí) vedou k zablkování programu. Délka signálů na vstupech nemá vliv na funkci programu.

## 8.2. Textové zprávy



**Obr. 101**

Blok slouží k zobrazení předdefinovaného textu, nebo parametru zvoleného bloku. K dispozici máme 10 bloků textových zpráv.

Vstup En (Enable *vysl. enejbl* = povolení): Náběžná hrana na vstupu En spustí zobrazení textu zprávy.

Výstup: Je sepnutý tak dlouho, dokud je zpráva zobrazena.

a) Parametrový vstup P

P Určuje prioritu zprávy. ( 0 nejnižší, 30 nejvyšší.).

Když se při běhu programu sepne více textových zpráv najednou, je na displeji zobrazena zpráva s nejvyšší prioritou. Pro listování mezi dalšími zprávami použijeme následující postupy.

- Na modulu LOGO! kurzorovými klávesami nahoru / dolů.
- V programu LogoSoftComfort kliknutím myši na záložku s číslem zprávy v poli TEXTOVÁ ZPRÁVA Číslo záložky je priorita zprávy.

**Aktivní zpráva má číslo priority červené.**

**Neaktivní zpráva má číslo priority modré.**

Quit: Potvrzení textu zprávy

Quit = off. Potvrzení textu zprávy zakázáno. (**Výchozí stav.**)

Quit = on. Potvrzení textu zprávy povoleno. Do potvrzení je zpráva zobrazena na displeji a výstup bloku je sepnutý i když En=0.

- Na modulu LOGO! se potvrzení zprávy provede klávesou OK.
- V programu LogoSoftComfort se potvrzení zprávy provede kliknutím myši na aktivní políčko POTVRZRNÍ v okně textové zprávy.

b) Parametrový vstup Par: Zde se nastavují další parametry textové zprávy.

Text: Vlastní text zprávy **se vkládá bez diakritiky.**

Par: Parametr nebo skutečná hodnota jiné, již naprogramované funkce.

### Popis funkce:

Při přechodu stavu vstupu En z 0 na 1 bude zobrazen zadaný text.

Potvrzení zprávy zakázáno: Při přechodu vstupu En z 1 do 0 bude zpráva smazána.

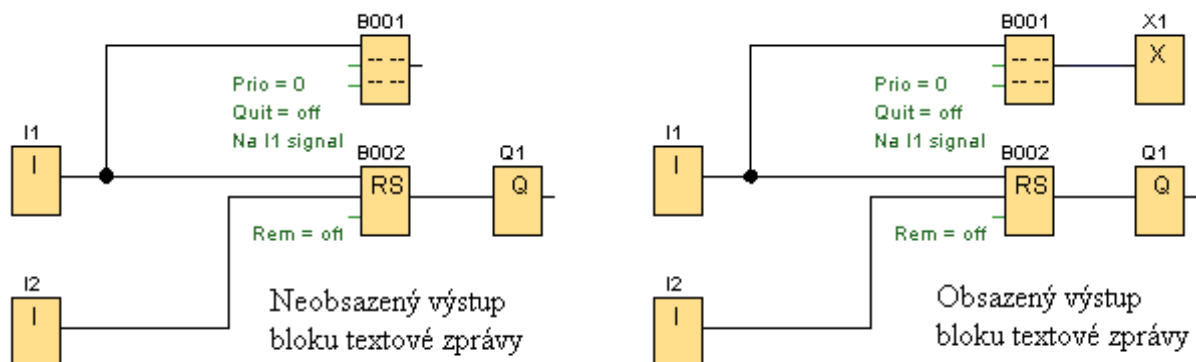
Potvrzení zprávy povoleno: Při přechodu vstupu En z 1 do 0 bude zpráva zobrazena, dokud nebude potvrzena. Zprávu je možno potvrdit, jen když na vstupu En není signál.

Mezi standardním zobrazením na displeji LOGO! a zobrazením textových zpráv se přepíná kurzorovými klávesami nahoru / dolů.

Signál z výstupu bloku textové zprávy je možné použít v programu jako každý jiný signál.

### Ukázka 1. Neobsazený výstup bloku textové zprávy

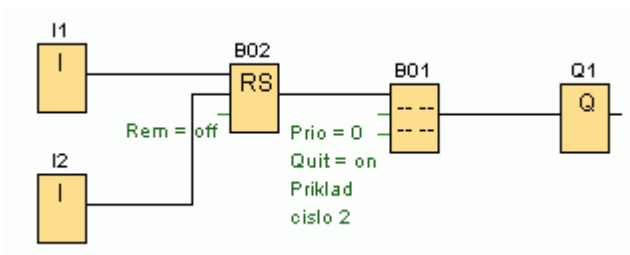
Tento program nepůjde přenést do LOGO! (o ručním programování nemluvě!). Blok textové zprávy musí být na výstupu napojen alespoň na virtuální výstup.



Obr. 102. Ukázka 1

### Ukázka 2. Vypnutí Q zablokováno do potvrzení textové zprávy

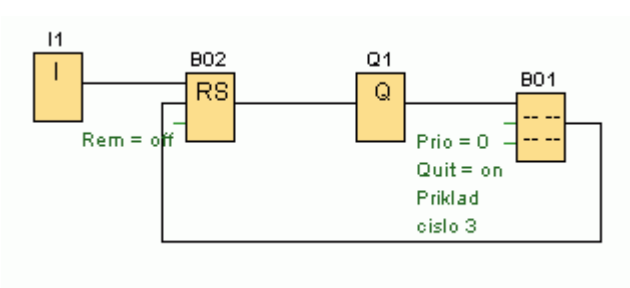
Po přivedení signálu na I1 se zpráva zobrazí a sepne Q1. Po přivedení signálu na vstup I2 se resetuje B02. B01 má povoleno potvrzení textové zprávy (Quit = on). Do potvrzení zprávy je zobrazený text a sepnutý výstup Q1.



Obr. 103. Ukázka 2

### Ukázka 3. Opětovné zapnutí Q zablokováno do potvrzení textové zprávy

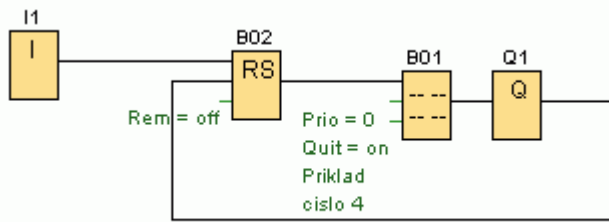
Po přivedení signálu na I1 sepne B02, zobrazí se zpráva, která rekurzí vazbou resetuje B02. Výstup Q1 vypne a textová zpráva půjde teprve nyní potvrdit. Do doby potvrzení zprávy je blokováno sepnutí výstupu Q1.



Obr. 104. Ukázka 3

**Ukázka 4.** Výstup Q vypne až po potvrzení textové zprávy

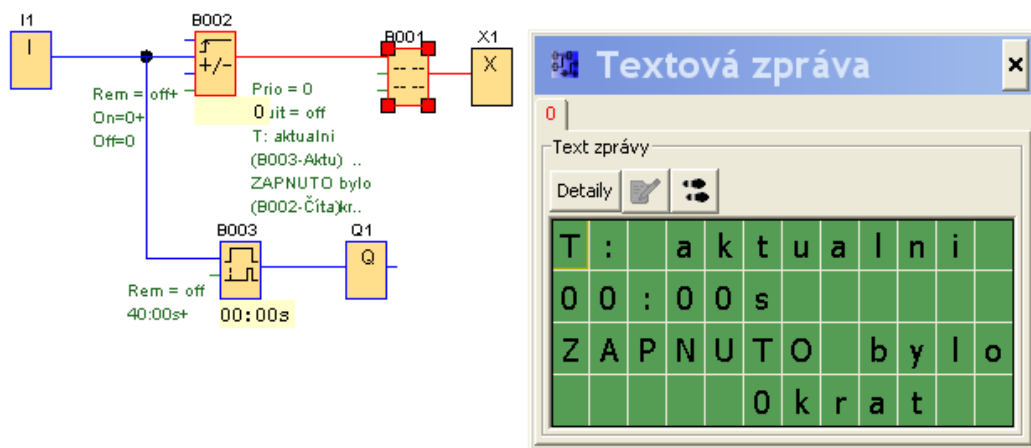
Po přivedení signálu na I1 sepne B02, zobrazí se zpráva a sepne Q1. Rekurzivní vazbou se resetuje B02, ale Q1 se vypne až po potvrzení textové zprávy.



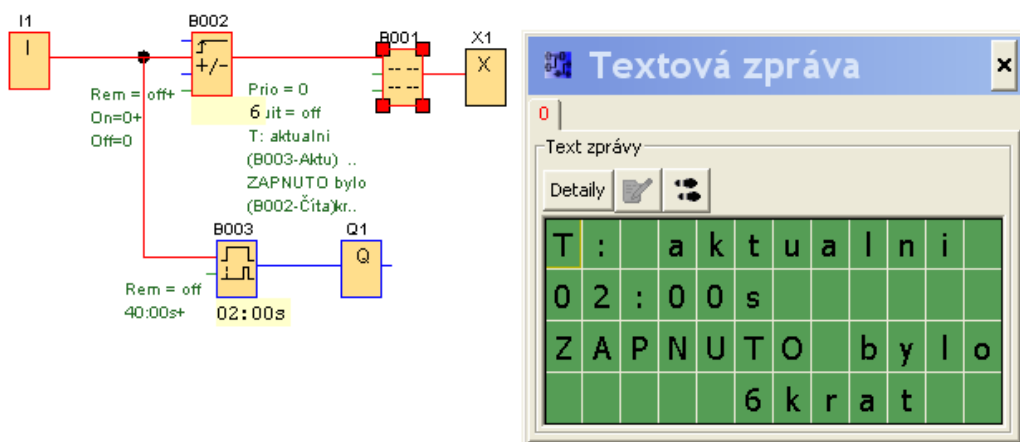
*Obr. 105. Ukázka 4*

**Ukázka 5.** Zobrazení přednastaveného textu spolu s parametry jiných bloků

Signál na I1 sepne B01 a zobrazí se předdefinovaný text spolu s určitými parametry B02, B03. Proč se ihned po spuštění programu (obr.106) zobrazí textová zpráva?



*Obr. 106. Ukázka 5*

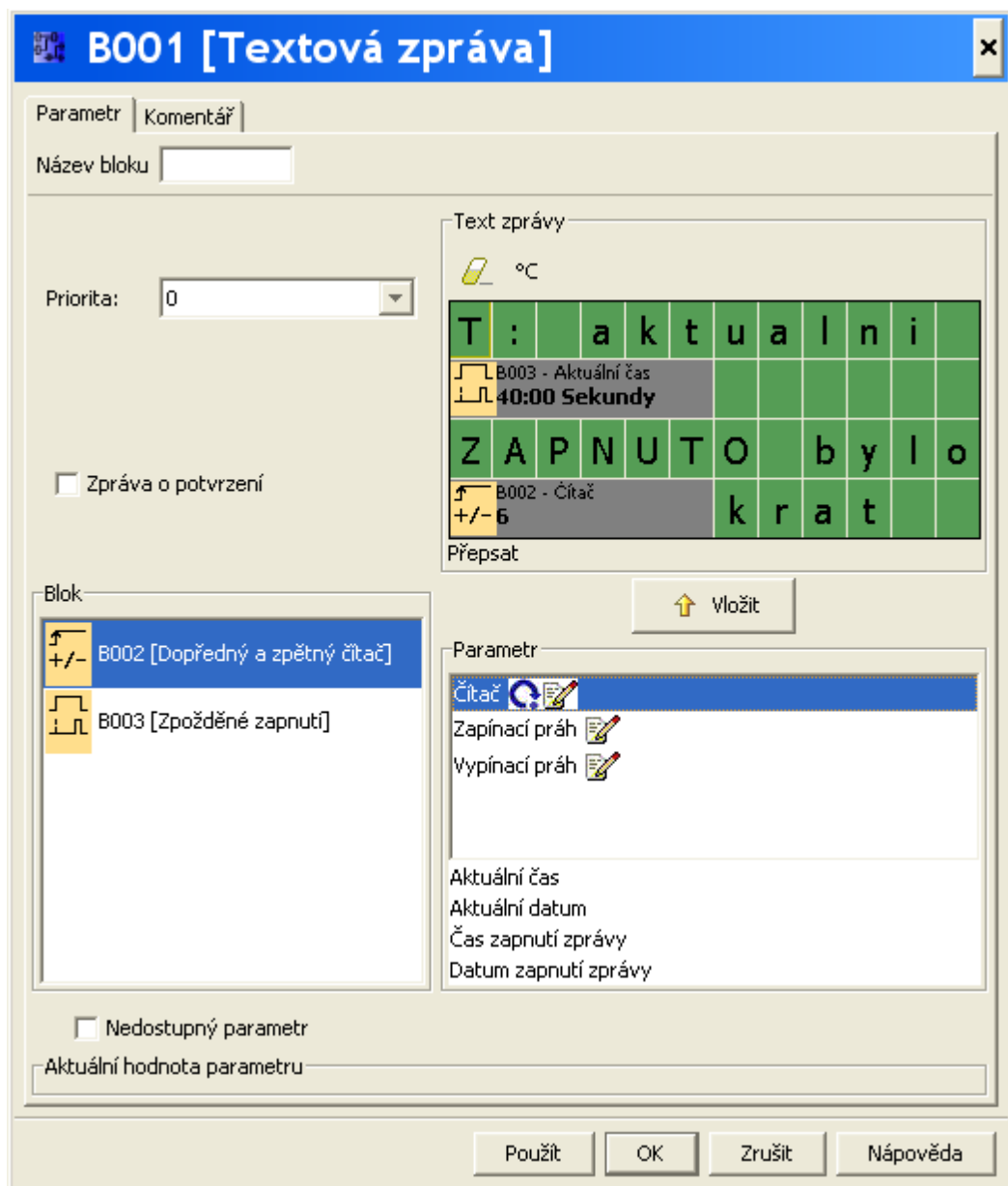


*Obr. 107. Ukázka 5*

### Konfigurace bloku Textové zprávy softwarově

Pravým tlačítkem myši otevře nabídku VLASTNOSTI BLOKU a vyvoláme dialog, obr. 107. V poli BLOK vidíme všechny bloky programu, jejichž parametry je možné zobrazit pomocí textových zpráv. Vybereme levým tlačítkem myši blok a jeho parametry se objeví v poli PARAMETR. Preferovaný parametr přesuneme táhnutím, nebo vložíme tlačítkem VLOŽIT do pole TEXT ZPRÁVY. Jednu řádku je možné použít pro jeden sledovaný parametr, který může být doplněn dalším textem.

Vhodnou kombinací textu a parametrů bloků můžeme informovat obsluhu o běhu programu a stavu zařízení.



Obr. 107. Konfigurace bloku textových zpráv softwarově

## Ukázka 6. Priorita

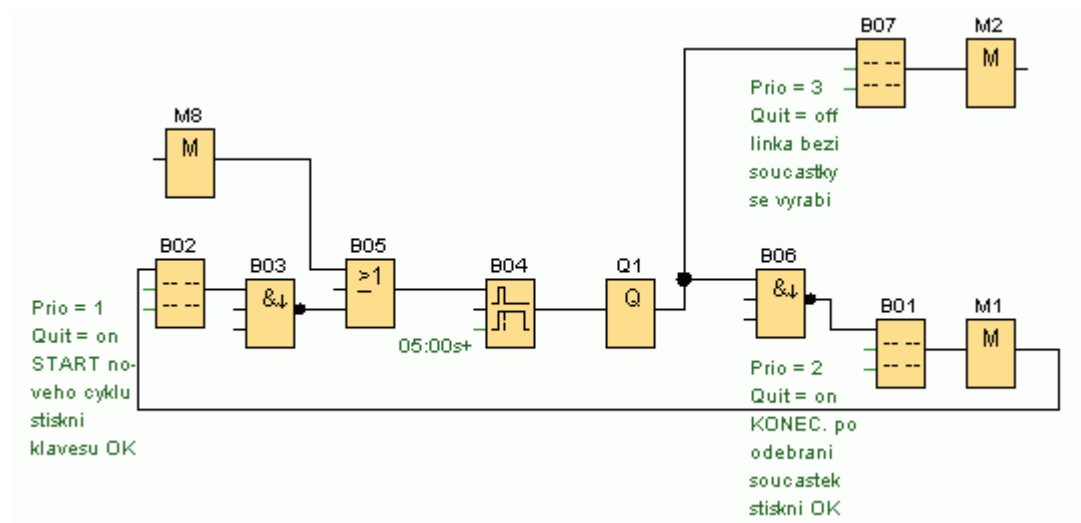
Současný stav programu podle obr. 108.

Výrobní linka (Q1) se spouští na nastavený čas zapnutím LOGO!. Dokud linka běží, je zobrazena zpráva č.1 LINKA BEZI SOUCASTKY SE VYRABI. Po uplynutí nastaveného času (zde 5 sekund) linka vypne a objeví se zpráva č.2 KONEC. po odebrání součástek stiskni OK a zpráva č.3. START nového cyklu stiskni klavesu OK. Zprávy č.2 a č.3 jsou s potvrzením. Potvrzením zpráv obsluha dává najevo, že požadované operace byly provedeny.

Do potvrzení zpráv je další běh programu zablokován.

**Všimněme si, že bloky textových zpráv BO1 a B02 jsou sepnuté současně, pořadí zobrazení na displeji je určeno prioritou.**

Pomocí LogoSoftComfort doplňte zprávu č.1 (B07) o zobrazení aktuálního času běhu linky.



Obr. 108. Ukázka 6

## 8.2.1. Příklady na procvičení

**Příklad 48.** Aktuální informace o stavu výstupu

Jestliže I1 = 0, potom výstup Q1 = 0.

Informuje o tom textová zpráva nevyžadující potvrzení VYSTUP Q1 ROZEPNUT.

Jestliže I1 = 1, potom výstup Q1 = 1.

Informuje o tom textová zpráva nevyžadující potvrzení VYSTUP Q1 SEPNU.

**Příklad 49.** Aktuální informace, priorita a blokování pomocí nepotvrzené zprávy

Třífázový motor je spínán pomocí stykače.

Ovládání motoru je pomocí tlačítek. Motor je chráněn před přetížením tepelnou ochranou.

Motor půjde spustit po odstranění poruchy a potvrzení zprávy č.4.

Doplňte o textové zprávy.

č.1 nevyžadující potvrzení: MOTOR STOJI.

č.2 nevyžadující potvrzení: MOTOR BEZI.

č.3 nevyžadující potvrzení: MOTOR PORUCHA.

č.4 vyžadující potvrzení: PORUCHA ODSTRANENA, STISKNI OK.

**Příklad 50.** Textová zpráva, aktuální informace, kombinace textu a parametru

Signál na I1 sepne výstup Q1 se zpožděním 10 sekund.

Signál na I2 vypne výstup Q1.

Doplňte o textové zprávy, které budou informovat obsluhu o:

- Vypnutém výstupu.
- Nastaveném a aktuálním času zpožděného zapnutí výstupu podle obr. 109.
- Zapnutém výstupu.



*Obr. 109. K příkladu 50*



## 9. Závěrečné příklady

### Příklad 51. Blokování dosaženým počtem nebo časem

Po třetím ukončeném pracovním cyklu, nebo nepřerušeném běhu zařízení delším než je nastavený čas (pro ladění nastavte na 5 sekund), se zablokuje další spouštění zařízení.

Blokování je signalizováno sepnutím Q2 pokud uplynul čas.

Blokování je signalizováno sepnutím Q3 pokud uplynul počet pracovních cyklů.

Začátek pracovního cyklu: Tlačítko na vstupu I1 sepne výstup Q1.

Konec pracovního cyklu: Tlačítko na I2 vypne výstup Q1.

Opakovaný signál na vstupech nesmí mít negativní vliv na funkci obvodu.

### Příklad 52. Cyklus – po opětovném zapnutí program pokračuje z místa vypnutí

Signál na vstupu I1, CYKLUS ZAP.

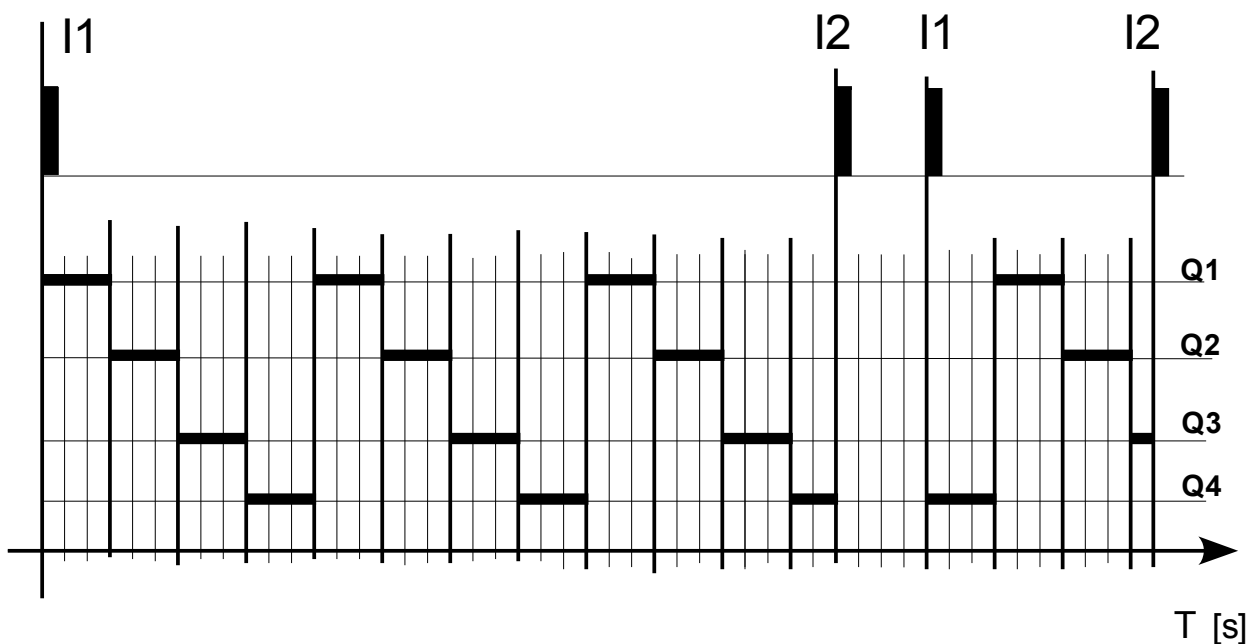
Signál na vstupu I2, CYKLUS VYP.

Výstupy spínají a vypínají postupně v pořadí Q1, Q2, Q3, Q4, Q1, Q2 ... atd

Každý výstup je sepnutý 3 sekundy a potom vypne.

Cyklus můžeme kdykoliv ukončit signálem na I2.

Při příštím spuštění pokračuje cyklus z místa vypnutí, viz obr.110.



Obr. 110. K příkladu 52

Po úspěšném odladění předchozí části doplňte.

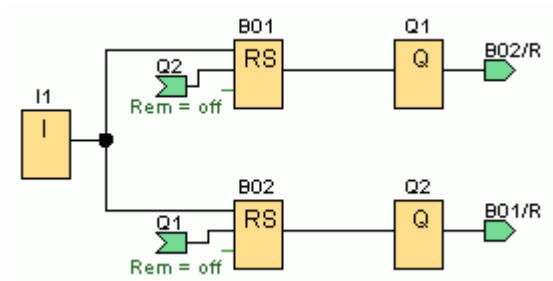
Jeden signál na I2 způsobí, že při následném spuštění pokračuje cyklus z místa vypnutí.

Dva po sobě jdoucí signály na I2 způsobí, že při následném spuštění začíná cyklus sepnutím výstupu Q1.

**Příklad 53.** Výstupy spínají a vypínají střídavě 10 sekund

Popis programu podle obr.111.

Na vstupu I1 je připojen vypínač. Po přivedení trvalého signálu na vstup I1 se střídavě spínají a vypínají výstupy Q1 a Q2. Po zrušení signálu se výstupy vypnou.



**Obr. 111.** K příkladu 53

Upravte zapojení na obr.111 takto.

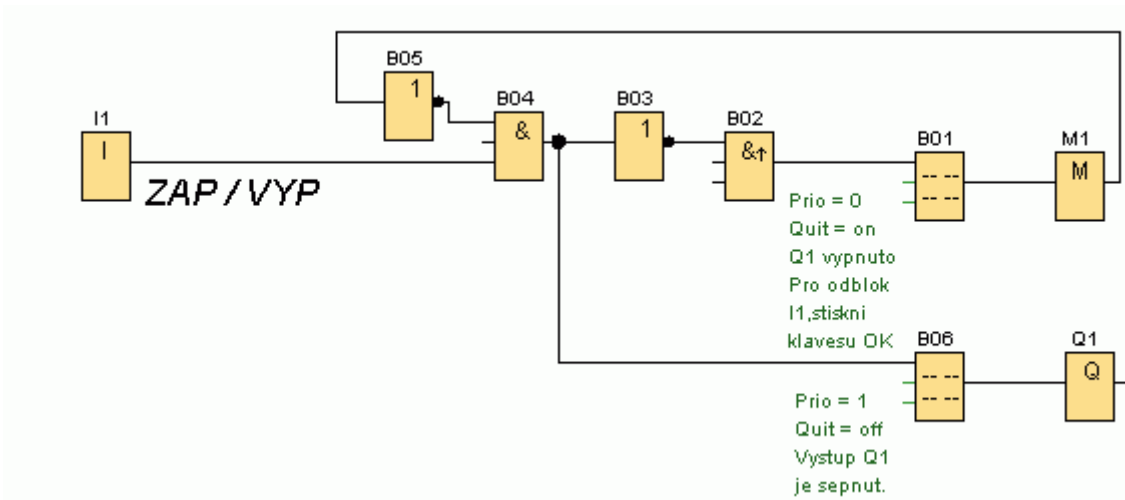
Program se spustí zapínacím tlačítkem na vstupu I1.

Program poběží 10 sekund a potom se automaticky ukončí.

Textové zprávy budou vhodně informovat o probíhající činnosti programu.

**Příklad 54.** Sepnutí výstupu blokováno textovou zprávou

1. Popište funkci programu podle obr. 112. Na vstupu I1 je momentálně připojený vypínač.



**Obr. 112.** K příkladu 54

2. Upravte program tak, aby na vstupu I1 bylo tlačítko.

První stisknutí tlačítka sepne Q1, následující stisknutí tlačítka vypne Q1. Zobrazování a blokování textovými zprávami zůstane zachováno.

3. Vyřešte úkol:
- a) S použitím libovolných bloků.
  - b) Jen pomocí bloků GF, nepočítaje bloky textových zpráv.