



**Промислени електронни системи**

Плевен 5800, ул. "Осогово" No 27

тел.: 064/870-170, тел./факс: 064/870-172, GSM 0888 646 100

e-mail: [office@vema-bg.com](mailto:office@vema-bg.com) <http://www.vema-bg.com>

---

---

## ПРОГРАМИРУЕМИ КОНТРОЛЕРИ

VPC106

VPC108A4

VPC1410

VPC1812

VPC2416

VPC3224L

VPC4030

VPC3224A63

VPC4030A84

## РЪКОВОДСТВО ЗА ПОТРЕБИТЕЛЯ

---

---

## I. Въведение

Програмируемите логически контролери *VPC* притежават галванично разделяне от външната електрическа среда чрез оптоизолирани дискретни входни линии 24V/10mA, и оптоизолирани изходни линии 24V/2A.

Програмните модели и структурата на тези контролери е идентична, което създава удобство на потребителя.

Количеството на входните и изходни линии е с различен брой, съгласно таблицата в раздел Технически характеристики.

Контролерът *VPC3224A63*, освен дискретните притежава и 6 аналогови входни линии 0-20mA/0-10V и 3 изходни аналогови линии 0-20mA/0-10V, а *VPC108A4* е снабден с 4 аналогови входни линии 0-20mA/0-10V.

Контролерът *VPC4030A84* е от разширяем тип, като разширителите са с цифрови или аналогови входове и изходи.

Всички контролери притежават енергонезависима памет тип EEPROM за съхраняване на параметрите и FLASH памет за потребителската програма.

*VPC* контролерите всички типове са с вградена потребителска конзола, с която могат да се задават технологичните параметри в работен режим или да се въвежда и редактира потребителска програма в програмиращ режим.

Има възможност за въвеждане на програмата от персонален компютър по комуникационния интерфейс RS232C. Програмният продукт **VPC Host Interface** позволява създаване на потребителски програми както в мнемоничен вид, така и в графична релейно-контакторна среда.

Програмният език на *VPC* програмируемите контролери е с подсказваща мнемоника и богатство на команди, позволяващи бързото изграждане на гъвкави потребителски програми.

## II. Технически характеристики

Тип контролер	VPC106	VPC108A4	VPC1410	VPC1812	VPC2416	VPC3224L	VPC3224A63	VPC4030	VPC4030A84
Дискретни входове	10	10	14	18	24	32	32	40	40
Дискретни изходи	6	8	10	12	16	24	24	30	30
Аналогови входове		4					6		8*
Аналогови изходи							3		4*

\*С разширителен модул VPC84IO

### -Дискретни входове

- Входно напрежение - от 18 до 30 VDC
- Входен ток - 10 mA при 24V

### -Дискретни изходи

- Максимален изходен ток - 2A
- Макс. напрежение, приложено към изход - 30V
- Макс. напрежение на задействан изход - 1.4V при 2A

### -Аналогови входове

- 0-20mA/0-10V

### - Аналогови изходи

- 0-20mA/0-10V

### - Таймери от 00.00 до 99.99 s

- 32 броя (вж.Забележка)

### - Броячи до 9999

- 32 броя (вж.Забележка)

### - Помощни регистри (дискретни)

-96 броя

### - Помощни регистри (числови)

-78 броя

### - Капацитет на програмната памет

- 2048 реда

### - Максимално време на цикъл (за 2048 команди)

- 7 ms

### - Потребителска конзола

- вградена

- 
- Вид индикация:
    - LCD дисплей
      - 4 реда с по 16 символа ( за **VPC108A4** и **VPC3224A63**)
      - 2 реда с по 16 символа (всички останали модели)
    - светодиода      - за задействани входни и изходни линии  
(за моделите от **VPC2416** нагоре)
  
  - Интерфейс за връзка с РС      - RS232 C
  - Режими на работа:
    - работен с мониторинг      - **RUN & MONITOR**
    - редактиращ с търсене      - **PROGRAM**
  - Захранващо напрежение      - от 100 до 250 VAC
  - Работна температура      - от 0 до 50 °C
  - Температура на съхранение      - от -10 до 60 °C

### III. Регистри и операнди

- цифрови входни линии, обозначени с  $i00, \dots, i39$ ;
- цифрови изходни линии -  $o00, \dots, o29$ ;
- таймери' с времеконстанта между 00.00 и 99.99 s -  $t00, \dots, t31$ ;
- броячи' до 9999 -  $c00, \dots, c31$ ;

*' Забележка. Общият брой на таймерите и броячите е 32. Всеки един номер на регистър 0, ..., 31 може да бъде назначен или за таймер или за брояч. Вж. В Редактиращ режим PROGRAM как се извършва преназначаването от брояч към таймер и обратно.*

- помощни регистри (дискретни):
  - потребителски -  $r00, \dots, r87$ ;
  - системни:
    - $r88(RUN)$  - винаги "on", при преминаването му в "off" се прекратява режим **RUN** на контролера;
    - $r89(Err)$  - "on" сигнализира аварийно състояние на потребителската програма;
    - $r90(CON)$  - "on" ако контролерът е свързан към друго устройство по RS232C (*напр. РС с VPC Host Interface*);
    - $r91(VPCIO)$  - "on" ако контролерът е свързан към разширителен модул (*напр. VPC84IO*);
    - $r92$  - инвертира се всеки програмен цикъл;
    - $r93$  - цикъл с период 0.01 s;
    - $r94$  - цикъл с период 0.1 s;
    - $r95$  - цикъл с период 1 s.

Всеки от тези дискретни регистри ( $i, o, t, c, r$ ) може да бъде операнд на командите от езика за програмиране (вж. *Команди и мнемониката им*). Инструкциите за изпълнение (край на веригите) от типовете: *OUT, OUT NOT* или *LR* могат да имат за операнд само изходна линия или потребителски регистър.

- аналогови входни линии, обозначени с  $a00, \dots, a07$ ;
- аналогови изходни линии -  $d00, \dots, d03$ ;
- помощни регистри (числови) -  $p00, \dots, p77$ ;
- общи числови регистри -  $p00, \dots, p75$  - стойности от  $0000, \dots, 9999$ , които могат да се модифицират от потребителя в МОНИТОРЕН РЕЖИМ и програмно да се прехвърлят стойностите им към/от стойностите на таймерите/броячите;
- специални числови регистри :
  - $p76$ (*паралелен изход*) - стойността му се предава чрез първите 12 изхода, ако е употребен някъде в програмата.  $o00$  е младши, а  $o11$  - старши бит, като двата най-старши бита на  $p76$  се игнорират;
  - $p77$ (*паралелен вход*) - числовия еквивалент на първите 12 входа,  $i00$  - младши бит,  $i11$  - старши бит;

Всеки от тези числови регистри ( $a, d, p$ ) може да бъде операнд за сравняване или копиране на командите от езика за програмиране (вж. *Команди и мнемониката им*). Инструкциите за изпълнение  $OUT$  и  $OUT NOT (=OUT^*)$  върху тези регистри (без  $a00-a07$  и  $p77$ ) се нуждаят от втори операнд и служат за прехвърляне на стойностите на двата операнда.

При отпадане на захранването се запазват стойностите и състоянието на всички броячи, таймери, числови регистри, както и на  $r72-r87$ .

## IV. Команди и мнемониката им

Командите и операндите (регистрите) се визуализират с подходяща мнемоника за LCD индикация, като смисълът им е следният:

- “ $END$ ” - край на програмата;
- “ $LD$ ”/” $LD NOT$ ” (LoaD/LoaD NOT) - начало на верига с нормално отворен/затворен контакт (или числово сравнение, когато се използват числовите регистри);
- “ $AND$ ”/” $AND NOT$ ” - последователно свързване на нормално отворен/затворен контакт (или числово сравнение);
- “ $OR$ ”/” $OR NOT$ ” - паралелно свързване на нормално отворен/затворен контакт (или числово сравнение);
- “ $OUT$ ”/” $OUT NOT$ ” - изход/инверсен изход (край на верига) или инструкция за прехвърляне на стойност от един регистър в друг при изпълнение на условията във веригата за него;
- “ $LR$ ” (Latching Relay) - изход тригер тип R-S. Изходът от този тип се включва и изключва по две независими вериги, като приоритетна е нулиращата верига, т.е. ако е изпълнено условието за нулиращата верига, изходът “ $LR$ ” няма да се задейства при никакви условия;
- “ $TIM00$  до  $TIM31$ ” (Timer) - изход таймер. Този изход се задейства, когато изтече времето, зададено като числова стойност при условие, че веригата за него е с изпълнени условия и не прекъсва през това време. Прекъсване на веригата установява таймера в изходно положение;
- “ $CNT00$  до  $CNT31$ ” (Counter) - изход брояч. Изходът от този тип се включва след отброяване на зададеното число импулси по броящия вход и се нулира след изпълнение на условието/условията по нулиращия вход;

- 
- “*IL*” (InterLock) - начало на разклонение. Тази команда създава възможност за осъществяване на разклонение със серия изходи, чийто вериги ще могат да ги задействат при изпълнение на собствените си условия и условията до тази команда;
  - “*ILC*” (InterLock Clear) - край на разклонение;
  - “*JMP*” (JuMP) - условен преход. Инструкциите, затворени в областта от командата “*JMP*” до “*JPE*” се изпълняват само, ако текущото състояние на веригата до “*JMP*” е с изпълнение условия за включено “*on*”;
  - “*JPE*” (JumP End) - край на прехода. Командата “*JMP*” изисква задължително употребата на “*JPE*”, с което се постига прескачане на веригите между тях при неизпълнено условие на веригата до “*JMP*”;
  - “*AND LD*” (AND LoaD) - последователно въвеждане на група контакти, когато между тях има и паралелни клонове;
  - “*OR LD*” (OR LoaD) - паралелно въвеждане на група контакти;
  - “*??????*” - непозната команда.

## V. Режими на работа

Контролерите *VPC* работят в два основни режима - работен с мониторинг (**RUN & MONITOR**) и редактиращ (**PROGRAM**) с възможности за търсене на изходни вериги.

Работният режим е основния режим на контролера, тъй като в този режим се изпълнява потребителската програма и съответното управление на елементите, свързани към входовете и изходите му.

Редактиращият режим е спомагателен и е основно предназначен за използване от специалисти с необходимата квалификация за потребителско програмиране, например за незначителни промени в потребителската програма директно в контролера без да се ползва персонален компютър и специализираната програма *VPC Host Interface*.

Превключването между режимите се осъществява чрез последователно натискане на бутона “*Esc*”, докато на дисплея се появи съобщение, че може да се премине към другия режим - “*За преход към RUN натиснете <Ins>*” или “*За изход от RUN натиснете <Ins>*”. Преходът от **PROGRAM** в **RUN** е възможен само ако потребителската програма е коректна, в противен случай контролерът указва каква грешка е открита и остава в **PROGRAM** (вж. *Програмни грешки*).

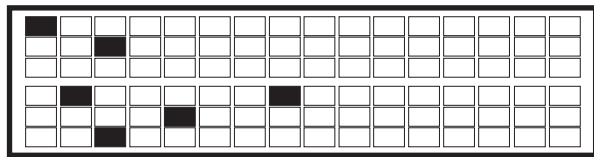
Контролерът притежава възможност и за дистанционен достъп до потребителската програма в редактиращ режим, ако контролерът е свързан по интерфейсен кабел RS232C към персонален компютър със стартирана комуникационна програма *VPC Host Interface*.

### V.a. Режим RUN & MONITOR

В режим **RUN & MONITOR** контролерът изпълнява последователно инструкциите на потребителската програма започвайки от инструкцията на адрес “0000” (начало на програмен цикъл) до първата срещната инструкция “*END*” (край на програмен цикъл). Този режим се визуализира на четири последователни прозореца на дисплея, което дава възможност на потребителя за контрол над регистрите и програмата на контролера. Последователното придвижване от прозорец в прозорец се осъществява чрез бутона [Esc]:

- Първи прозорец на режим **RUN & MONITOR**.

Той изобразява състоянието на входните и изходните линии на VPC контролерите, като първите два реда изобразяват входовете, а последните два - изходите, подредени в последователен ред. Плътен квадрат означава включено ("on"), а празен - изключено ("off") състояние. В примера от фигурата входовете i01, i07, i20, i34 и изходите o00, o18 са "on", а всички останали входове и изходи са "off".



Изходи: o00 ... o15  
o16 ... o29

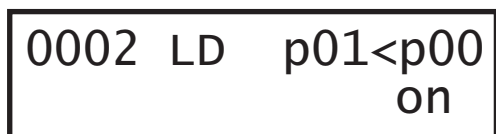
Входове: i00 ... i15  
i16 ... i31  
i32 ... i39

първи прозорец на **RUN&MONITOR**

- Втори прозорец на режим RUN & MONITOR. В този прозорец се изобразява мониторинният подрежим на контролера, в който потребителят може да наблюдава състоянието на всеки регистър. Чрез бутона [←] се избира (поставя в мигащ режим) типа на регистъра за наблюдение, номера му и причислената му константа (за таймерите и броячите). Избраният елемент може да се променя чрез стрелките [←] и [→] и по такъв начин на дисплея да се визуализира произволен регистър на контролера. Текущото състояние на така избрания регистър се индикира вдясно от него - "on" за включено и "off" за изключено, а за таймерите и броячите се индикира и текущата им стойност. В този подрежим по горепосочената процедура е възможна и смяна на времеконстантите на таймерите и константите на броячите.

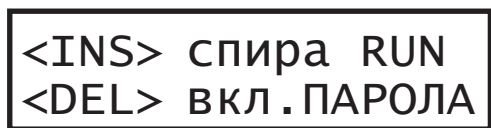


- Трети прозорец на режим RUN & MONITOR. На практика това е редактиращият режим (вж. *Редактиращ режим PROGRAM*) на контролера, но без възможността потребителят да променя програмата, тъй като в момента тя се изпълнява. Последните три символа на втория ред отразяват моментната стойност на операнда:

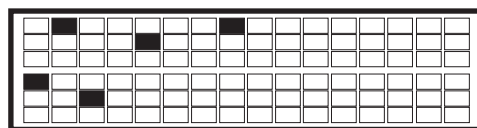


програмата може да се разглежда чрез бутоните [↑] и [□], както и да се търсят изходни вериги (Вж. **SEARCH** в *Редактиращ режим PROGRAM*)

- Четвърти прозорец на режим RUN & MONITOR. Този последен прозорец служи за изход от работния режим на контролера и преход към редактиращ режим чрез натискане на бутона [Ins]. Потребителят може да се върне към първия прозорец на **RUN & MONITOR** чрез бутона [Esc]:



последен прозорец на **RUN & MONITOR**

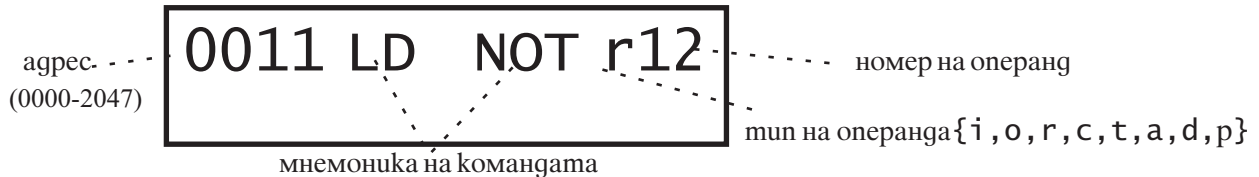


първи прозорец на **RUN & MONITOR**

С цел да се защити програмата от нежелателни промени, потребителят може да избере 'ПАРОЛА' чрез бутона [Del] и оттук нататък излизането от режим **RUN & MONITOR** става само след избиране паролата 'числото след VPC' - например: '4030' от четвъртия прозорец.

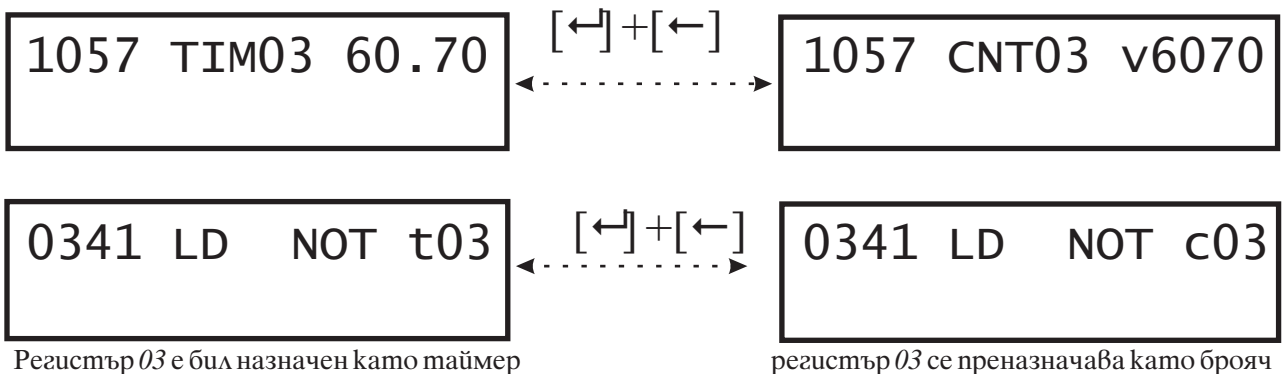
## V.b. Редактиращ режим PROGRAM

Този режим служи за въвеждане и коригиране на потребителската програма чрез вградената конзола на VPC контролерите. На дисплея се извеждат командите на програмата във следния формат: текущ адрес на команда, мнемоника на командата и нейния операнд (, ако такъв съществува):

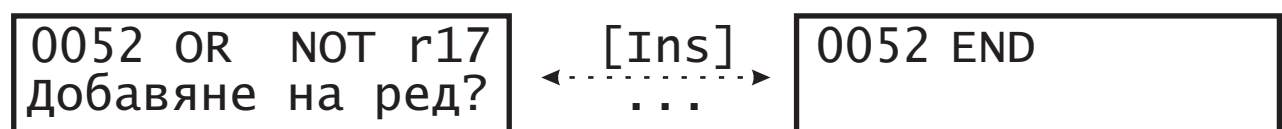


Потребителят може да разглежда програмата със стрелките: [↑] намалява текущия адрес, а [↓] го увеличава. Стрелките са многоскоростни, което позволява бързо достигане до произволен адрес на програмата. Комбинацията задържане на [↑] и [←] поставя на дисплея началото на програмата (адрес 0000), а комбинацията [↓] и [←] - края на програмата, т.е. първото срещане на командата 'END'. Текущият програмен рег може да се редактира като чрез [←] се избира кой компонент на командата (мнемоника, тип операнд, номер на операнд, константа) да се промени. Този компонент се поставя в мигащ режим, а съдържанието му може да се променя чрез стрелките [←] и [→]. Промененото съдържание се записва с бутона [↵]. В края на програмата задължително трябва да присъства командата 'END' за край на програмния цикъл.

При използването на броячите и таймерите като операнди на команди трябва да се помни, че един номер 00,...,31 може да е или само за брояч или за таймер. Преназначаването на даден регистър, напр. t03 от таймер към брояч c03, или обратно, се извършва като на програмния рег, където той участва се натисне и задържи [←] и се натисне еднократно [←] :



За вмъкване преди или изтриване на текущия програмен рег се използват бутоните [Ins] и [Del]. След еднократното им натискане контролерът очаква като потвърждение повторното им натискане. На дисплея се появява съобщение "Добавяне..." или "Изтриване..." с мигащо многоточие докато трае съответната операция и бутониерата е блокирана. При вмъкване новата команда на текущия програмен рег е 'END':



на адрес 0052 се вмъква инструкцията 'END', а 'OR NOT r17' преминава на адрес 0053

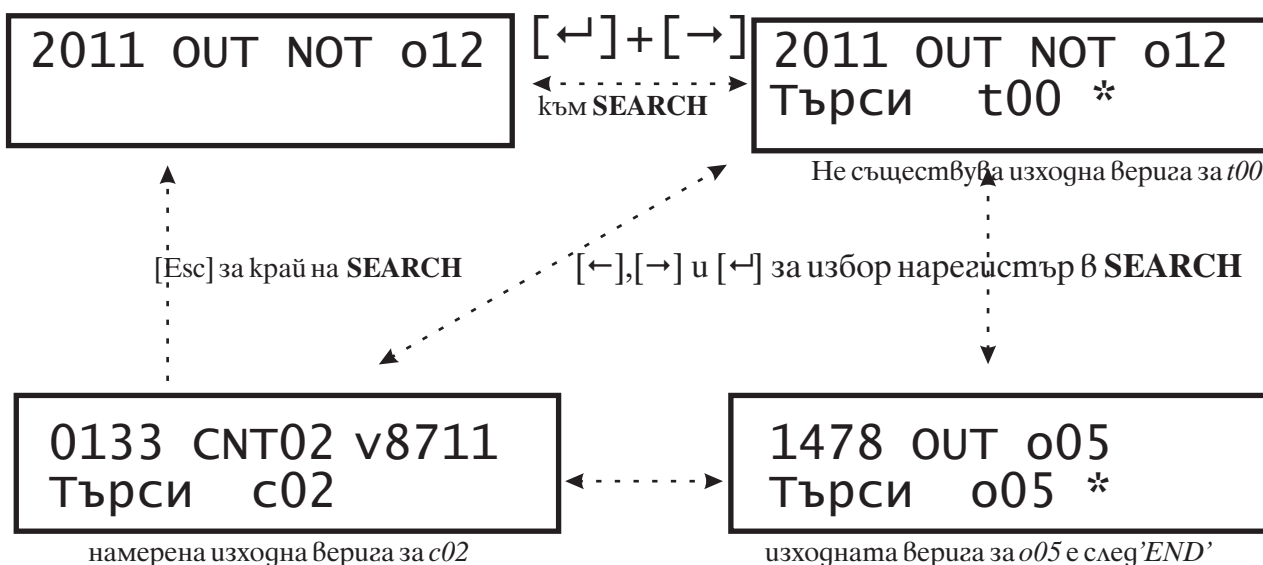
0611 LD NOT i08  
Изтриване на ред

[De] ]  
...  
←

0611 OR r11

инструкцията на на адрес 0611 се изтрива и на нейно място преминава тази от адрес 0612

Редактирацията режим **PROGRAM** притежава и подрежим **SEARCH** за търсене на изходни вериги в потребителската програма. **SEARCH** се извиква чрез задържане на [←] и натискане на бутона [→]. При това на втория ред на дисплея се изписва 'Търсене на t00 \*'. Отново с бутона [←] и стрелките [←] и [→] се избира произволен невходен регистър - изходна линия, таймер, брояч или вътрешно реле. За всеки така избран регистър на първият ред на дисплея се изписва програмния ред, на който завършва изходната верига за избрания регистър, или този регистър се маркира с "\*" на втория ред, ако такава верига не съществува в рамките на програмния цикъл. За изход от **SEARCH** се натиска бутона [Esc]:



## VI. Програмни грешки

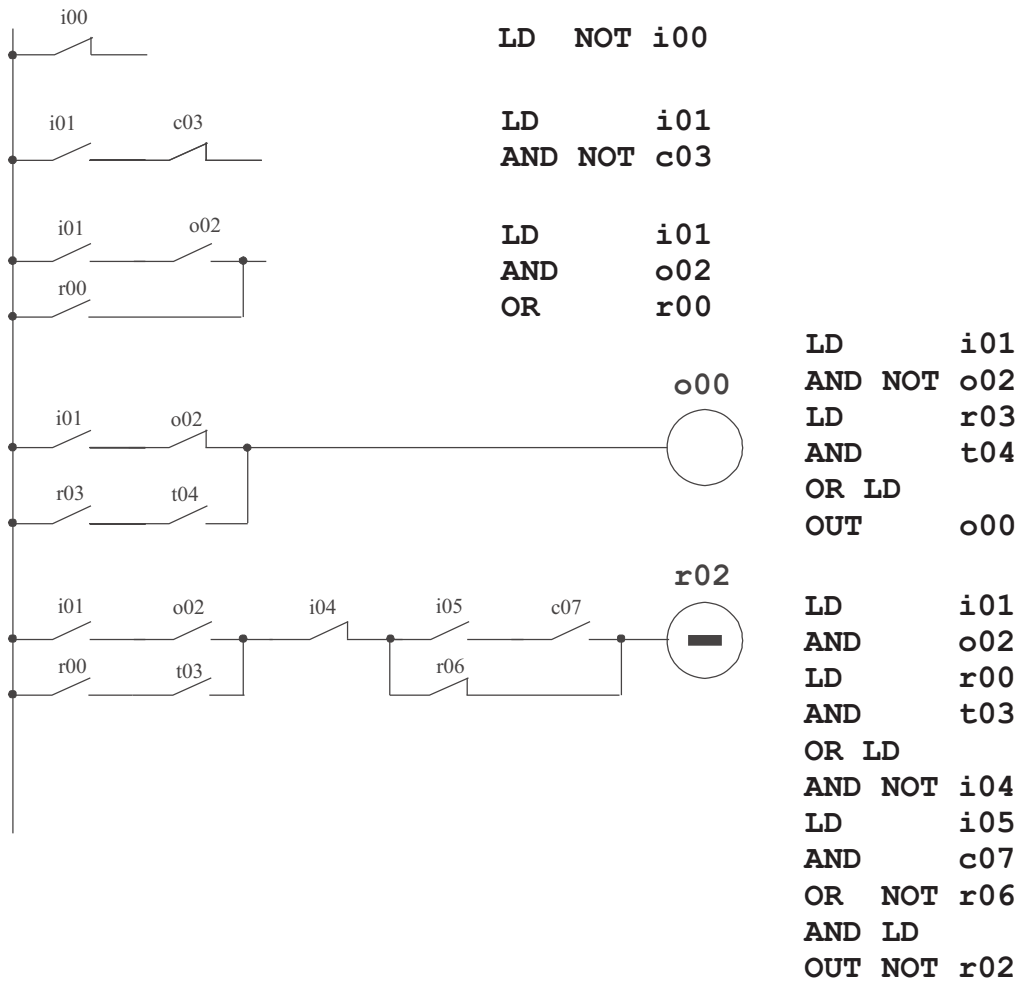
След редактиране на програмата и преди всяко нейно изпълнение при преход към работен режим (**RUN & MONITOR**), се извършва проверка на програмата и при наличие на грешки, те се визуализират, а контролерът преминава (остава) в редактиращ режим. Програмните грешки, които се обработват от VPC контролерите са:

- "Грешка с EEPROM!" - грешка с EEPROM;
- "Грешка по RS232!" - грешка при комуникация с РС в комуникационен режим;
- "Дублиран изход!" - повече от една изходна верига за даден регистър;
- "Неправилен код!" - синтактична грешка, непозната инструкция;
- "Грешен операнд!" - неправилен за текущата инструкция операнд;
- "Грешна стойност!" - неправилна стойност на таймер/брояч или индекс на регистър;
- "Липсващ 'END'!" - липсваща инструкция "END" за край на програмата;
- "Липсващ IL (JMP)" - командата "IL"/"JPE" няма предишна съответстваща "ILC"/"JMP" ;
- "Пресичане IL-JMP" - преплитане на командите "IL"- "ILC" и "JP"- "JPE".

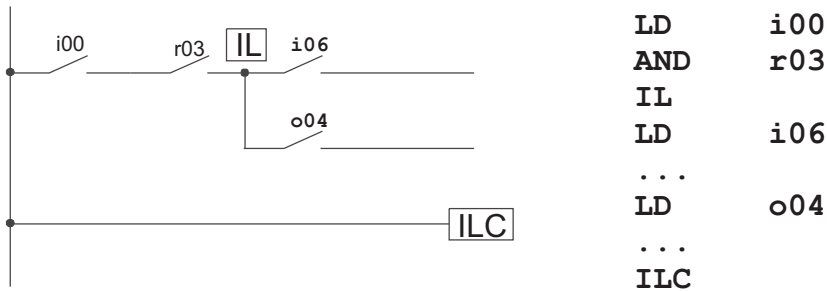


## VII. Примери на релейно-контакторна аналогия

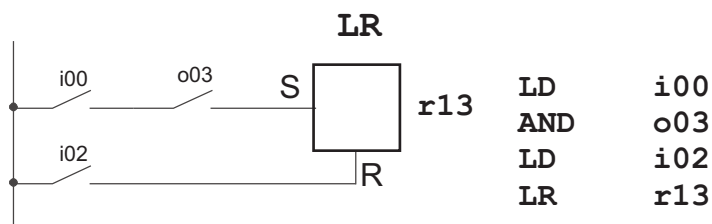
Инструкции LD; LD NOT; AND; AND NOT; OR; OR NOT; OR LD; AND LD; OUT; OUT NOT:



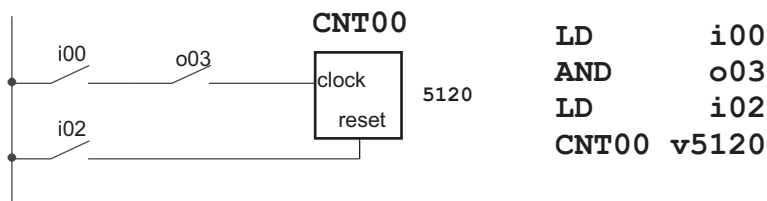
Инструкция IL (INTERLOCK) и ILC (INTERLOCK CLEAR) начало на разклонение и край на разклонение:



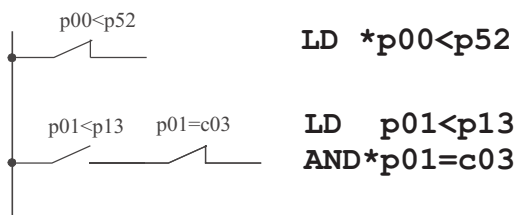
Инструкция LR (Latching Relay) реле със задържане, еквивалентно на RS тригер:



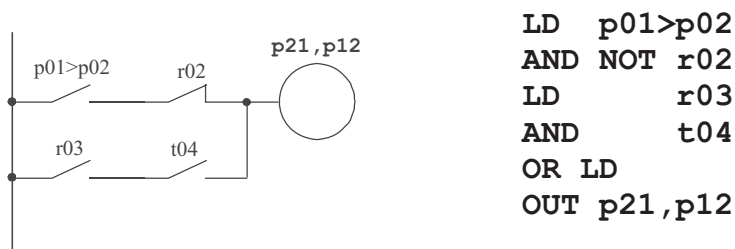
Инструкции CNT (Counter) брояч:



Инструкции за числово сравнение, изискващи два числови операнда:  
LD; LD NOT (=LD \*); AND; AND NOT (=AND\*); OR; OR NOT (=OR\*):



Прехвърляне на стойност от един регистър (*p12*) в друг (*p21*), ако е изпълнено някакво логическо условие с командите OUT и OUT NOT (=OUT\*):

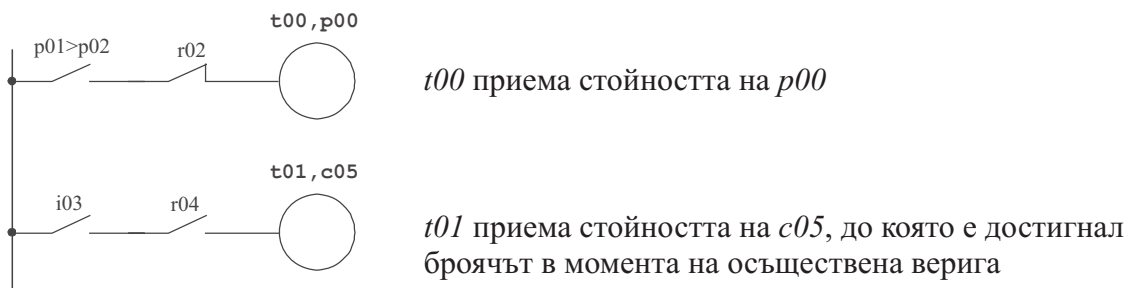


Прехвърлянето на числови стойности с командите OUT и OUT\* се извършва само, ако входът на командата е логическа единица, т.е. веригата е осъществена независимо по кой клон.

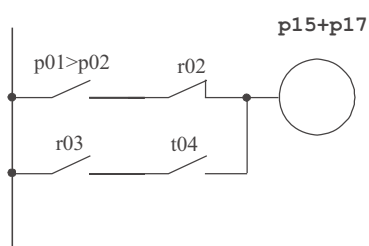
Прехвърлянето се извършва от втория към първия операнд (съдържанието на *p12* се прехвърля като съдържание на *p21*).

Когато като първи операнд е посочен таймер или брояч, прехвърлянето е към зададената му стойност. В случай, че вторият операнд е таймер или брояч, взема се неговата *текуща* стойност за прехвърляне.

Например:



Инструкции за аритметични действия на числови стойности с командите OUT и OUT\* .

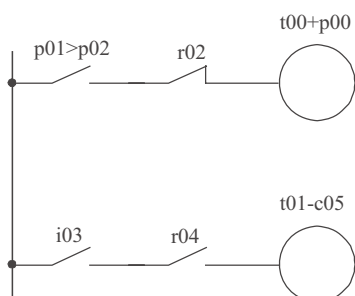


```
LD p01>p02
AND NOT r02
LD r03
AND t04
OR LD
OUT p15+p17
```

Сумирането се извършва, като към съдържанието на първия операнд се добавя това на втория операнд (съдържанието на  $p15$  се сумира с това на  $p17$  и резултата остава като съдържание на  $p15$ ).

Когато като първи операнд е посочен таймер или брояч, сумата става негова нова *зададена* стойност. В случай, че вторият операнд е таймер или брояч, се взима неговата *текуща* стойност за сумиране.

Например:



```
LD p01>p02
AND NOT r02
OUT t00+p00
```

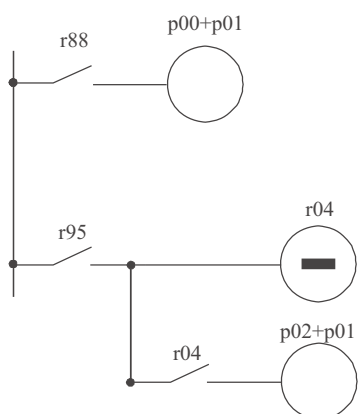
$t00$  приема стойността на сумата от собствената си зададена стойност и стойността на  $p00$

```
LD i03
AND r04
OUT t01-c05
```

$t01$  приема стойността на разликата от собствената си зададена стойност и стойността до която е достигнал  $c05$  до момента и тази разлика става новата му зададена стойност

Трябва да се има предвид, че аритметичната операция се извършва *непрекъснато*, докато условието е налице. Ето защо внимателно трябва да се подбира условието, особено за случаите, когато аритметичната операция трябва да се извърши еднократно.

Например:



```
LD r88
OUT p00+p01
```

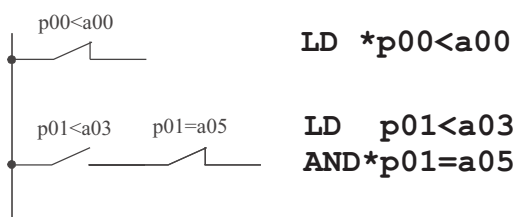
$p00$  ще нараства със стойността на  $p01$  всеки програмен цикъл

```
LD r95
IL
OUT NOT r04
LD r04
OUT p02+p01
ILC
```

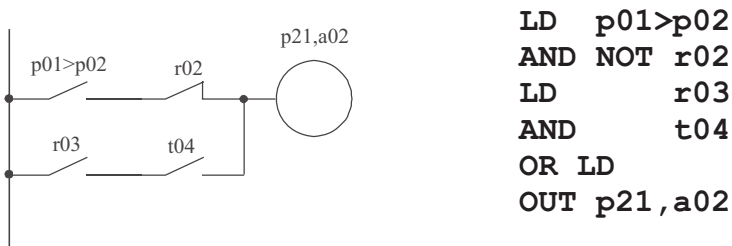
$p02$  ще нараства със стойността на  $p01$  всяка секунда

## Работа с аналоговите параметри:

Аналоговите входове се интерпретират по същия начин, както числовите параметри.



Прехвърляне на стойност от аналогов вход ( $a02$ ) в регистър ( $p21$ ), ако е изпълнено някакво логическо условие с командите OUT и OUT NOT (OUT\*):

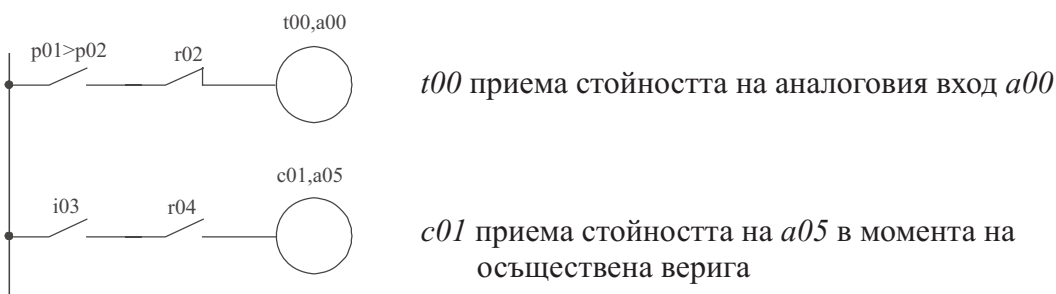


Прехвърлянето на числови стойности с командите OUT и OUT\* се извършва само, ако входът на командата е логическа единица, т.е. веригата е осъществена независимо по кой клон.

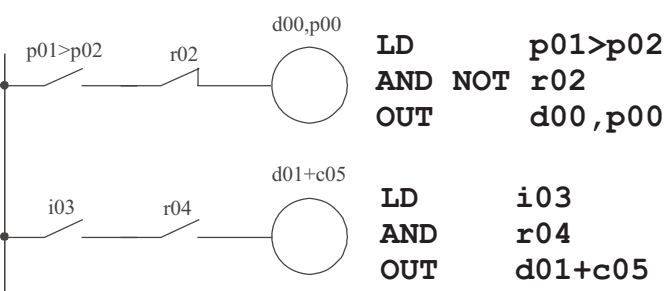
Прехвърлянето се извършва от втория към първия операнд (съдържанието на  $a02$  се прехвърля като съдържание на  $p21$ ).

Когато като първи операнд е посочен таймер или брояч, прехвърлянето е към *зададената* му стойност.

Например:



Инструкции за работа с аналоговите изходи:

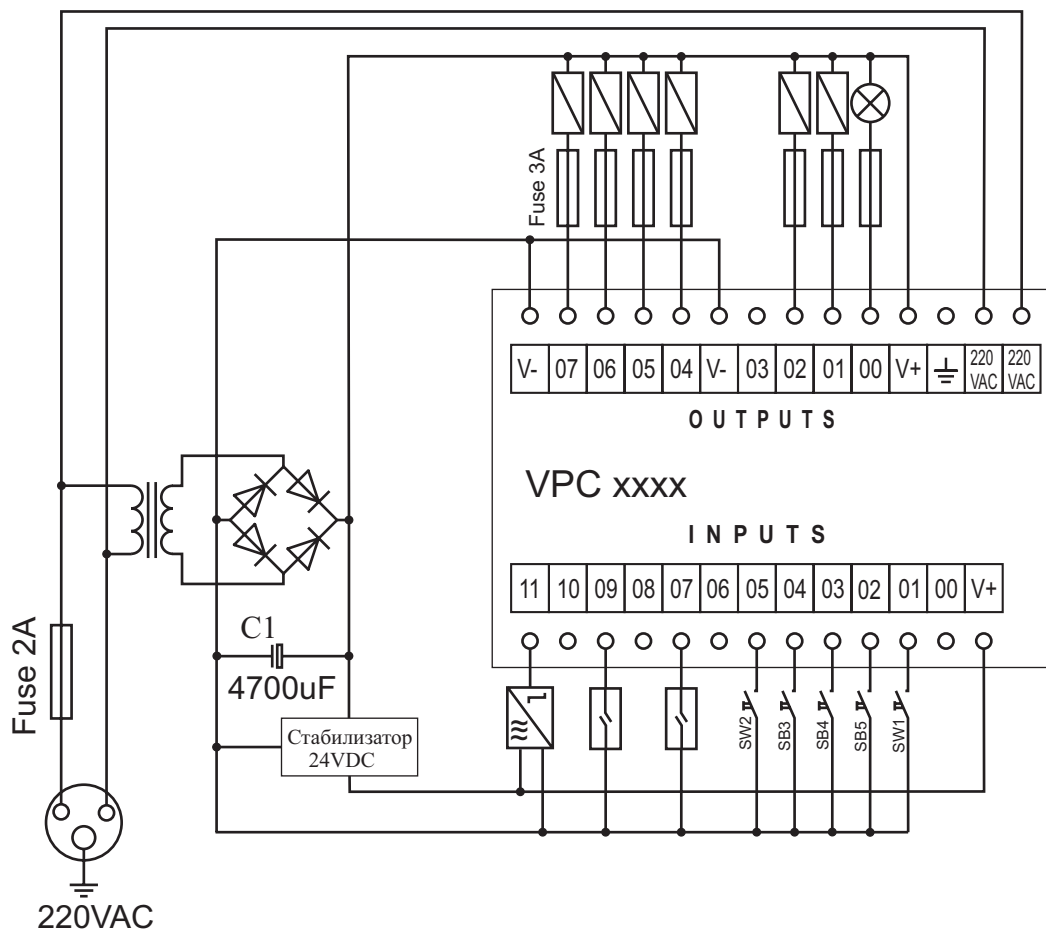


При изпълнение на условието  $p01 > p02$  и незадействано реле  $r02$  аналоговият изход  $d00$  приема стойността на параметъра  $p00$

При изпълнение на условието задействан вход  $i03$  и реле  $r04$  аналоговият изход  $d01$  приема сумата от собствената си стойност и стойността, до която е достигнал до момента броячът  $c05$

## VIII. Начин на свързване на контролерите VPC

Входовете и изходите стандартно се изпълняват с **NPN** логика. Това означава, че активните нива са  $0V$ .



Примерна схема за свързване на датчици и бутони към входовете и електромагнитни клапани и други товари към изходите на контролер от типа **VPC**.

Когато нестабилизираното напрежение от токоизправителя има вероятност да превиши допустимото за електронни датчици е необходимо да се стабилизира с подходящ стабилизатор на  $24VDC$ , като в този случай се свързва общата входна шина  $V+$  към стабилизирания източник, а общата изходна шина  $V+$  към нестабилизирания източник.

Всички **VPC** контролери имат достатъчно мощни изходни транзистори и могат да се свързват директни товари с ток на консумация до  $2 ADC$  с активен или индуктивен характер. Желателно е да се включват между изходите и товарите бързи предпазители за да се избегне дефектиране при евентуални къси съединения.