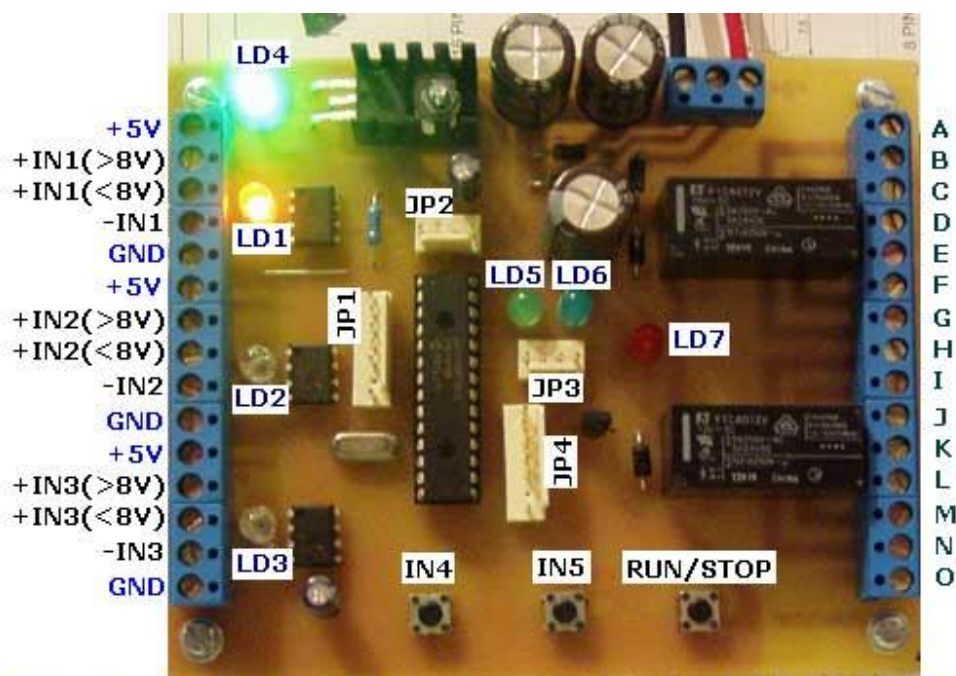


SW24x3 – programovatelné relé

Základní vlastnosti

1. Napájení modulu z externího zdroje 24VDC.
2. Tři externí galvanicky oddělené potenciálové vstupy (rozsah 3 až 26V) s možností zapojit je jako bezpotenciálové pro ovládání kontaktem (tlačítka, kontakty relé).
3. Dvě výstupní relé, každé s dvojicí přepínacích kontaktů (kontakty 230VAC/5A).
4. Vlastní BASIC interpret – jednoduché editování, programování a odlaďování programů bez nutnosti programování FLASH.
5. Odladěný program je možné uložit do vnitřní EEPROM - spustí se po zapnutí napájení.
6. Čtveřice analogových vstupů, 2 x UART (JP4, JP3), 2 x tlačítka s univerzální funkcí.
7. Tlačítka RUN/STOP pro zastavování a spouštění nahraného programu.



Modře zvýrazněné - galvanicky spojeno s napájením modulu (+5V a GND)

+5V (proudové omezení na max.10mA při zkratu s GND)

Černě zvýrazněné

Vstupy +IN a -IN jsou galvanicky oddělené od modulu.

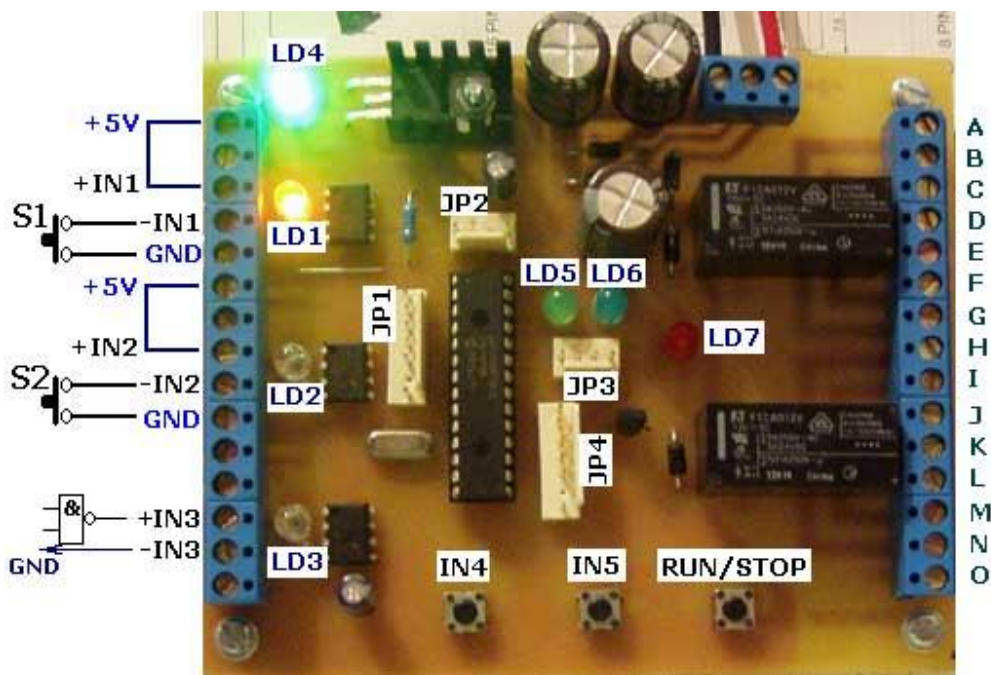
Legenda k obrázku: LD1, LD2, LD3 ... pro vstupy IN1, IN2 a IN3.

LD4 ... zelená POWER LED (přítomné napájení).

LD5 ... zelená – program běží (RUN).

LD6 ... modrá – sepnutý pomocný výstup R2 (v programu R2=1).

LD7 ... červená – sepnutý výstup R1 (obě relé sepnuta, v programu R1=1).



Příklad propojení pro ovládání dvěma kontakty (tlačítka, relé) a jedním externím signálem (TTL).

Vstupy IN1 a IN2 jsou ovládané tlačítka S1 a S2 (stisknuto = LED svítí , v programu vyhodnoceno jako stav log. jedniček).

Vstup IN3 externím TTL signálem (mezi +IN3 a -IN3 minimálně 3V = LD3 svítí , stav log.1 v programu).

Srdcem zařízení je moderní mikroprocesor zajišťující všechny nezbytné funkce včetně uložení všech potřebných funkcí a nastavení do vnitřní paměti EEPROM, jejíž obsah se nezmění ani po vypnutí či odpojení modulu od napájení.

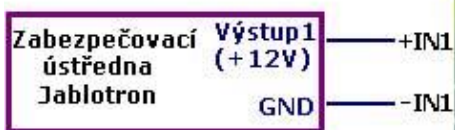
Na konektor JP1 je možné připojit až 4 analogové vstupy (potenciometry, čidla), v programu přístup prostřednictvím příkazů s proměnnými P0 až P9. Rozsah vstupního napětí 0 až 5V. Napájení potenciometrů (čidel) je možné rovněž přímo z desky (pin 1 konektoru JP1). Analogový vstup AN0 na pinu 2, AN1 na pinu 3, AN2 na pinu 4, AN3 na pinu 5, pin 6 je neosazen (klíč), GND je na pinu 7.

JP2 slouží k nahrávání nového firmware.

JP3 – druhý UART (UART2) a univerzální TTL výstup (R3).

Konektor JP4 slouží k připojení USB rozhraní pro ladění a práci s programem (USB převodník s galvanickým oddělením – v našem sortimentu např. USB-OPT3). Jinak UART č. 1.

Příklad ovládání externím signálem +12V



1. Popis ovládacích příkazů a parametrů komunikace

Ovládání: 9600bps, 8 bitů bez parity, 1 nebo 2 stop-bity.

Psaní programu je možné v jakémkoliv textovém terminálu s výstupem na USB porty.
(například: http://www.selfcontrol.cz/TM_comtester_V25.exe)

Komunikace s modulem (zadávání příkazů a programových řádků) je možná pouze ve stavu STOP. Při běhu programu jsou vstupy a výstupy přesměrované na samotný prováděný program.

V režimu RUN modul komunikuje s porty prostřednictvím příkazů PRINT (pro výpis) nebo načítání vstupních znaků. Pokud program běží, není jej možné editovat.

Všechny příkazy jsou zpracovány až po „odentrování“ – po odeslání <CR>

Základní příkazy pro editaci a nahrávání programu:

LIST <CR> ... vypíše obsah paměti z RAM (naposledy nahraný program).

CLS ... vymaže program v RAM pro ladění programu.

RUN ... spustí program z RAM (pakliže není v paměti žádný program, modul zůstane ve STOP režimu). Stejnou funkci jako příkaz RUN má tlačítko RUN/STOP na desce – jedním stiskem tlačítka se program spustí, dalším stiskem zastaví.

Pokud program běží, svítí zelená LED LD5.

Program se sám ukončí/zastaví, pokud je zpracován poslední řádek bez skoku zpět (uzavřená smyčka), tj. pokud není na posledním řádku příkaz nepodmíněného skoku nebo podmíněného, jehož podmínka nebyla splněna.

DEL číslo řádku ... vymaže daný řádek programu z RAM.

Příklad: DEL 3100 <CR> ... vymaže řádek 3100

HELP ... vypíše verzi firmware, např. *** SW24x3 – V1.0 ***

Jakýkoliv ASCII řetězec začínající číselnými znaky (číslo od 1 do 65535) je číslem řádku s příkazy a tento řetězec bude uložen do programu.

Příklad: 100 IF P1<P2 THEN 200 <CR>

EESAVE ... uloží program z RAM do vnitřní EEPROM. Po zapnutí modulu bude spuštěn.

Pokud nebude v paměti žádný program, nebude po zapnutí modulu nic spuštěno (LD5 zhasnuta).

EELoad ... kdykoliv je možné tímto příkazem zkopírovat program z EEPROM do RAM (po zapnutí napájení modulu před spuštěním programu je to provedeno automaticky).

2. Popis programových příkazů

Každý příkaz je ASCII řetězec začínající číslem řádku (znaky 30H až 39H) v rozsahu 1 až 65535. Následuje mezera (znak 20H) nebo více mezer ukončených příkazem. Konec řádku je ukončen znakem <CR> (0DH). Znak <LF> (0AH) není nutně vyžadován. Při zpětném výpisu programu (příkazem LIST) bude každý řádek vždy zakončen oběma znaky v pořadí **0DH 0AH**.

Znaky malé i velké abecedy mají stejný význam a jsou zaměnitelné – v příkladech programů bude pro přehlednost zvoleno psaní příkazů velkými písmeny).

K dispozici je 10 proměnných s identifikátory P0 až P9, přičemž P0 a P9 mají specifickou funkci časovačů (časovače setin vteřiny). Každých 10ms je stav P0 i P9 snížen o 1 směrem k nule. Bude-li například nastaveno P9=1000, pak za 10 vteřin bude proměnná P9 rovna nule. V podmíněných skocích je tak možné využít přesného časování.

Příkazy se rozdělují na:

- Nepodmíněné skoky.
- Podmíněné skoky s relačními operátory.
- Přiřazovací příkazy pro výstupy/relé R1 až R3 a pomocné bity R4 až R8.
- Přiřazení číselné hodnoty nebo vstupu proměnným P0 až P9.
- Aritmetické příkazy s proměnnými P0 až P9.
- Výpis textů a obsahu proměnných na porty UART1 a UART2.

2.1.Nepodmíněný skok

100 GOTO 20 <CR> ... nepodmíněný skok na řádek 20 (pokud řádek 20 v programu chybí, program nebude spuštěn – zelená LED LD5 po stisku RUN jenom blikne).

POZN: Překlad a kontrola cílových adres jsou provedeny těsně před spuštěním programu.

2.2.Podmíněný skok

110 IF <podmínka> THEN 320 <CR> ... nepodmíněný skok na řádek 320 (pokud řádek 320 v programu chybí, program nebude spuštěn – zelená LED LD5 po stisku RUN jenom blikne).

Je-li podmínka na řádku 110 splněna, bude proveden skok na řádek 320, jinak bude vykonán řádek následující.

Typy podmínek:

- Testování vstupů IN1 až IN3, tlačítek IN4 a IN5.
- Testování výstupů R1 až R3 a pomocných bitů R4 až R8
- Porovnávání proměnných P0 až P9 mezi sebou.

Testování vstupů

@A=1 ... Pokud je vstup IN1 aktivní (v log. 1, svítí LED LD1), je podmínka splněna.
@A=0 ... Pokud je vstup IN1 neaktivní (LED LD1 zhasnuta), podmínka je splněna.
@A=R... Na vstupu IN1 došlo k pulsu z log. 0 do log.1 nebo k sepnutí kontaktu/tlačítka.
Příznakový bit je čtením podmínky vynulován, aby nedocházelo k jeho akumulování.

@B=1 ... Pokud je vstup IN2 aktivní (v log. 1, svítí LED LD2), je podmínka splněna.
@B=0 ... Vstup IN2 neaktivní (LED LD2 zhasnuta), podmínka splněna.
@B=R... Na vstupu IN2 došlo k pulsu z log. 0 do log.1 nebo k sepnutí kontaktu/tlačítka.
Příznakový bit je čtením podmínky vynulován, aby nedocházelo k jeho akumulování.

@C=1 ... Pokud je vstup IN3 aktivní (v log. 1, svítí LED LD3), je podmínka splněna.
@C=0 ... Vstup IN3 neaktivní (LED LD3 zhasnuta), podmínka splněna.
@C=R... Na vstupu IN3 došlo k pulsu z log. 0 do log.1 nebo k sepnutí kontaktu/tlačítka.
Příznakový bit je čtením podmínky vynulován, aby nedocházelo k jeho akumulování.

@D=1 ... Pokud je tlačítko IN4 drženo/tištěno, podmínka je splněna.
@D=0 ... Tlačítko IN4 je uvolněno, podmínka splněna.
@D=R... Tlačítko IN4 bylo stisknuto/sepnuto, podmínka je splněna. Příznakový bit pro IN4 je čtením této podmínky vynulován, aby nedocházelo k jeho akumulování.

@E=1 ... Pokud je tlačítko IN5 je drženo/tištěno, podmínka je splněna.
@E=0 ... Tlačítko IN5 je uvolněno, podmínka splněna.
@E=R... Tlačítko IN5 bylo stisknuto/sepnuto, podmínka je splněna. Příznakový bit pro IN5 je čtením této podmínky vynulován, aby nedocházelo k jeho akumulování.

Testování výstupů

@1=1 ... Pokud je relé R1 sepnuto (svítí červená LED LD7), je podmínka splněna.
@1=0 ... Pokud je relé R1 vypnuto, podmínka je splněna.
@1=R... Relé R1 cvaklo nebo právě seplo, podmínka je splněna. Příznakový bit pro R1 je čtením této podmínky vynulován, aby nedocházelo k jeho akumulování.

@2=1 ... Pokud je výstup R2 sepnutý (svítí modrá LED LD6), podmínka je splněna.
@2=0 ... Pokud je výstup R2 vypnutý, podmínka je splněna.
@2=R... Puls na výstupu R2. Příznakový bit pro R2 je čtením této podmínky vynulován, aby nedocházelo k jeho akumulování.

@3=1 ... Výstup R3 je sepnutý (TTL úrovně na pinu 1 JP3, pin4 je GND), podmínka splněna.
@3=0 ... Pokud je výstup R3 vypnutý, podmínka je splněna.
@3=R... Pokud nastal puls na výstupu R3, podmínka je splněna.
Příznakový bit pro R3 je čtením této podmínky vynulován, aby nedocházelo k jeho akumulování.

Testování pomocných bitů (příznaků) – binární proměnné R4 až R8

@4=1 ... Pokud je proměnná R4 nastavena, podmínka je splněna.

@4=0 ... Pokud je proměnná R4 vynulována, podmínka je splněna.

@4=R... Podmínka splněna, pokud právě došlo ke změně stavu proměnné z 0 do 1.

Příznakový bit pro R4 je čtením této podmínky vynulován, aby nedocházelo k jeho akumulování.

Podobně platí pro proměnné R5 až R8.

POZN: Všechny příznakové bity R pro vstupy IN1 až IN5, výstupy R1 až R3 a proměnné R4 až R8 jsou při spuštění programu vynulovány.

Testování aritmetických proměnných P0 až P9 - relace

P4=P1 ... Pokud jsou hodnoty proměnných P4 a P1 stejné, podmínka je splněna.

P2<P8 ... Pokud je obsah proměnné P2 menší než P8, podmínka je splněna.

P3>P7 ... Pokud je obsah proměnné P3 větší než P7, podmínka je splněna.

POZN: Všechny proměnné jsou typu WORD (Unsigned Integer), rozsah 0 až 65535.

2.3. Přiřazovací příkazy pro výstupy a pomocné bity

120 R1=1 <CR> ... sepne relé Re1 (svítí červená LED LD7).

130 R2=1 <CR> ... rozsvítí modrou LED LD6.

140 R3=1 <CR> ... nastav výstup R3 do log. 1.

150 R4=1 <CR> ... nastav pomocný bit R4.

160 R5=1 <CR> ... nastav pomocný bit R5.

170 R6=1 <CR> ... nastav pomocný bit R6.

180 R7=1 <CR> ... nastav pomocný bit R7.

190 R8=1 <CR> ... nastav pomocný bit R8.

200 R1=0 <CR> ... vypne relé Re1 (LD7 zhasne).

210 R2=0 <CR> ... zhasne modrou LED LD6.

220 R3=0 <CR> ... nuluj výstup R3.

230 R4=0 <CR> ... nuluj pomocný bit R4.

240 R5=0 <CR> ... nuluj pomocný bit R5.

250 R6=0 <CR> ... nuluj pomocný bit R6.

260 R7=0 <CR> ... nuluj pomocný bit R7.

270 R8=0 <CR> ... nuluj pomocný bit R8.

280 R\$=0 <CR> ... vypne/nuluje všechny výstupy R1 až R3 a pomocné bity R4 až R8.

290 R\$=1 <CR> ... zapne/nastaví všechny výstupy R1 až R3 a pomocné bity R4 až R8.

2.4. Přirázovací příkazy pro proměnné P0 až P9

Přirazení přímých číselných hodnot/konstant proměnným

- 300 P1=100 <CR>** ... přiřazení konstanty 100 proměnné P1.
- 305 P4=0 <CR>** ... nuluje proměnnou P4.
- 310 P5=24500** ... přiřazení číselné hodnoty 25400 proměnné P5.
- 315 P9=200** ... nastaví proměnnou/časovač P9 (po uplynutí 2 vteřin bude P9 vynulována).

Nepřímá přiřazení proměnným

- 320 P8=P7 <CR>** ... hodnotu P7 zkopíruj do P8, P7 beze změny.
- 330 P1=P2 <CR>** ... hodnotu P2 zkopíruj do P1, P2 beze změny.
- 340 P0=P1 <CR>** ... hodnotu P1 zkopíruj do časovače P0, P1 beze změny.
- 350 P3=P9 <CR>** ... zkopíruj časovač P9 do proměnné P3.

Načtení analogových vstupů AN0 až AN3

- 400 P1=#0 <CR>** ... načtení zdigitalizovaného analogového vstupu AN0 do proměnné P1 (číselný rozsah 0 až 127, 0 odpovídá 0V, 127 odpovídá 5V na vstupu AN0, 7bitový rozsah).
- 401 P1=#1 <CR>** ... načtení zdigitalizovaného analogového vstupu AN1 do proměnné P1
- 402 P8=#2 <CR>** ... načtení zdigitalizovaného analogového vstupu AN2 do proměnné P8
- 403 P2=#3 <CR>** ... načtení zdigitalizovaného analogového vstupu AN3 do proměnné P2

Načtení znaků z přijímacích bufferů UART1 (JP4) a UART2 (JP3)

- 410 P4=%1 <CR>** ... načte jeden znak ze vstupního bufferu UART1 do proměnné P4. (Je-li po provedení příkazu P4 rovna 0, nebyl v bufferu žádný přijatý znak. Po provedení příkazu je znak ve vstupním bufferu UART1 odstraněn. Znak je číslo – maximální hodnota 255.).
- 411 P1=%2 <CR>** ... načte jeden znak ze vstupního bufferu UART2 do proměnné P1. (Je-li po provedení příkazu P1 rovna 0, nebyl v bufferu žádný přijatý znak. Po provedení příkazu je znak ve vstupním bufferu UART2 odstraněn).

2.5. Aritmetické příkazy pro proměnné P0 až P9

Aritmetické příkazy s jednou proměnnou

- 500 **P5++** ... inkrementace obsahu proměnné P5.
- 510 **P1--** ... dekrementace obsahu proměnné P1.
- 520 **P2+3** ... obsah proměnné P2 zvýš o 3.
- 530 **P4-100** ... od proměnné P4 odečti 100.

Aritmetické příkazy s více proměnnými

- 540 **P1=P4+P1** ... k proměnné P1 přičti P4. Obsah proměnné P4 neměň.
- 550 **P6=P7-P8** ... od P7 odečti P8 a výsledek ulož do P6, přičemž P7 a P8 beze změny.
- 560 **P3=P1*P2** ... vynásob P1 s P2 a spodních 16 bitů ulož do P3, horních 16 bitů ulož do proměnné o 1 vyšší, v tomto případě do P4, přičemž P1 a P2 beze změny.
- 570 **P1=P2/P3** ... vyděl P2 / P3 a výsledek 16-bitového dělení ulož do P1.

POZN: Mezi proměnnými, znaménky a indexy nesmí být žádný jiný znak ani mezera, jinak bude příkaz špatně vyhodnocen.

Například:

- 580 **P1=P2 +P3** ... bude vyhodnoceno jako přiřazení proměnné P2 do P1 místo součtu P2 s P3.

2.6. Vypis textů a obsahu proměnných – příkaz PRINT

- 600 **PRINT1“obecný text“** ... odešle text v uvozovkách na port UART1 (9600bps).
- 610 **PRINT2“** text ** “** ... odešle text v uvozovkách na port UART2 (9600bps).
- 620 **PRINT1LN“Ahoj“** ... odešle text v uvozovkách na port UART1 s odřádkováním (CR+LF).
- 630 **PRINT2LN“ text “** ... odešle text v uvozovkách na port UART2 se zakončením CR+LF.

- 640 **PRINT1(P4)** ... vypíše obsah proměnné P4 na portu UART1 (9600bps).
- 650 **PRINT2(P1)** ... vypíše obsah proměnné P1 na portu UART2 (9600bps).
- 660 **PRINT1LN(P1)** ... vypíše obsah proměnné P1 na portu UART1 se zakončením CR+LF.
- 670 **PRINT2LN(P9)** ... vypíše obsah proměnné P9 na portu UART2 se zakončením CR+LF.

3. Příklady programů

3.1 Relé Re1 sepne, pokud je alespoň jeden ze vstupů IN1 až IN5 aktivní

```
100 IF @A=1 THEN 200
110 IF @B=1 THEN 200
120 IF @C=1 THEN 200
130 IF @D=1 THEN 200
140 IF @E=1 THEN 200
150 R1=0
160 GOTO 100
190 R2=0
200 R1=1
210 GOTO 100
```

3.2 Stiskem tlačítka inkrementuj P1 a vypiš na UART1, odesláním znaku „B“ na UART1 zvyš P1 o 10 a vypiš zpětně obsah P1 na UART1

```
10 P1=100
20 P2=65
100 IF @D=R THEN 200
110 P3=%1
120 IF P3=P2 THEN 300
160 GOTO 100
200 P1++
210 GOTO 400
300 P1+10
400 PRINT1LN(P1)
410 GOTO 100
```

3.3 Hladinový spínač - zapínání relé Re1 a výstupu R2 podle napětí na výstupu AN0, je-li AN0 vyšší než 100, sepne Re1, jinak vypne

```
10 P1=100
20 P2=#0
30 IF P2>P1 THEN 100
40 IF P2<P1 THEN 200
50 GOTO 20
100 R1=1
110 R2=0
120 GOTO 20
200 R1=0
210 R2=1
220 GOTO 20
```