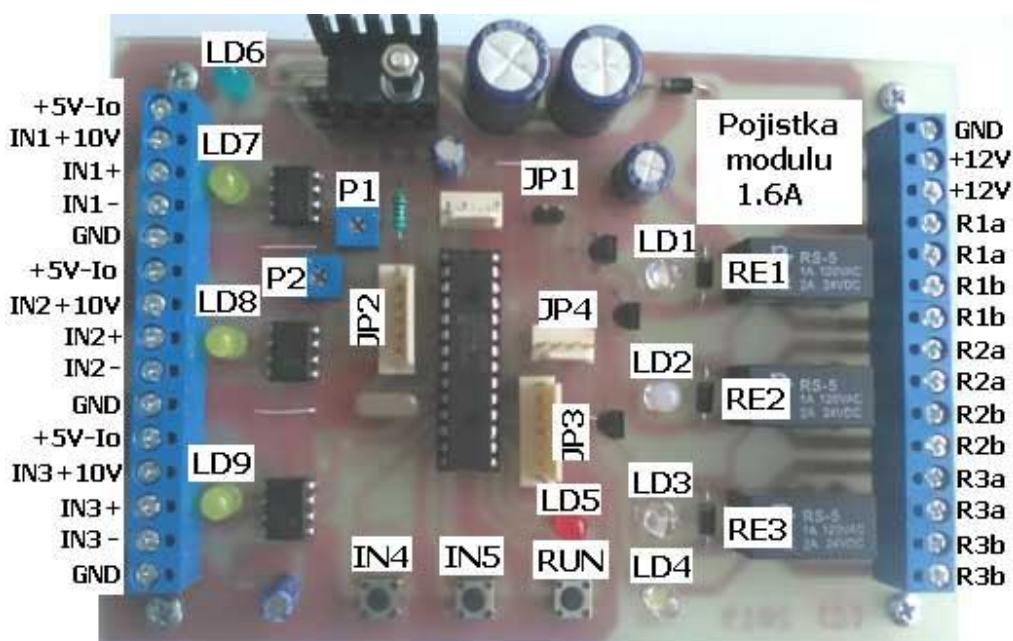


MKS3x3 – programovatelné relé

Základní vlastnosti

- 1 Napájení modulu z externího zdroje v rozsahu 9 až 24VDC.
- 2 Tři externí galvanicky oddělené potenciálové vstupy (rozsah 3 až 28V) s možností zapojit je jako bezpotenciálové pro ovládání kontaktem (tlačítka, kontakty relé).
- 3 Tři výstupní relé, každé s dvojicí přepínacích kontaktů (kontakty max. 120VAC/2A).
- 4 Ovládání z počítače – funkce jako USB relé.
- 5 Autonomní režim - vlastní BASIC interpret – jednoduché editování, programování a odlaďování programů bez nutnosti programování FLASH.
- 6 Odlaďený program je možné uložit do vnitřní EEPROM - spustí se po zapnutí napájení.
- 7 Čtveřice analogových vstupů, 2 trimry, 1 jumper s využitím v programu (v podmínkách větvení programu IF ... THEN).
- 8 2 x UART (JP3, JP4), 2 x tlačítka s univerzální funkcí.
- 9 Tlačítka RUN/STOP pro zastavování a spouštění nahraného programu.



Legenda:

Svorkovnice vstupů:

Vstupy **+5V-Io** jsou vyvedená napájení z modulu proudově limitovaná na maximální proud 9mA (rezistory 560R). Bez zatížení je na svorkách +5V.

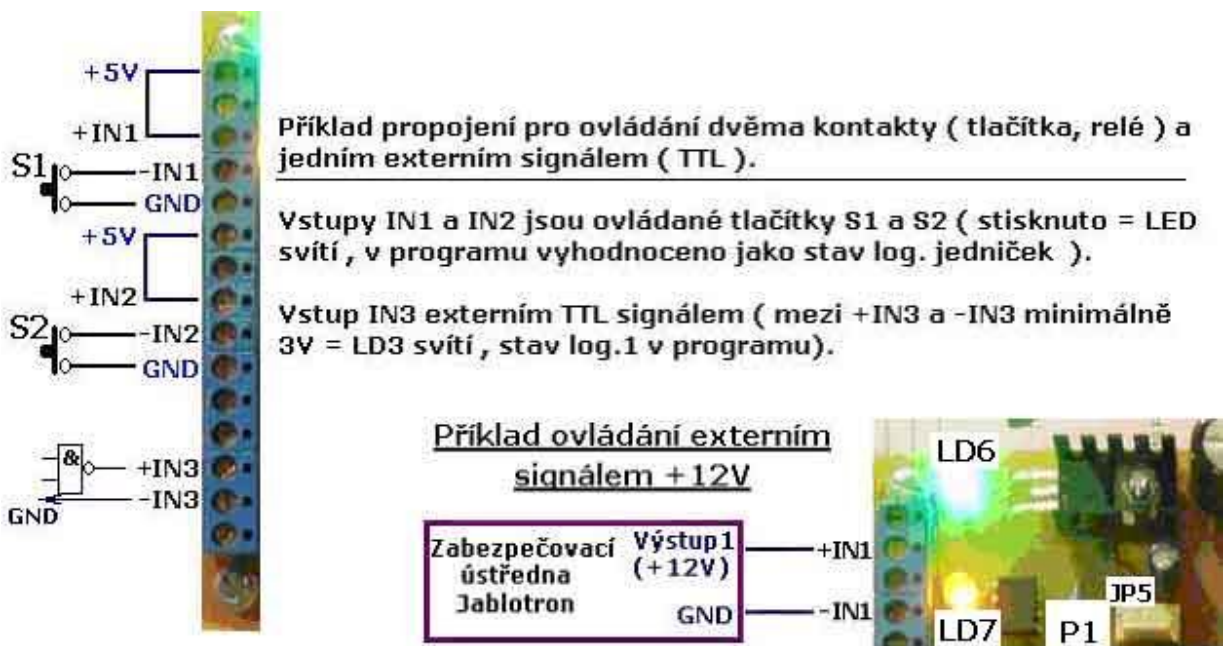
Vstupy **INx+10V**, **INx+** a **INx-** jsou galvanicky oddělené (optrony) od modulu.

INx+10V jsou určeny pro vstupní napětí nad 10V.

INx+ jsou určeny pro vstupní napětí do 10V.

- LD1 ... zelená LED svítí, pokud je sepnuté relé RE1
- LD2 ... zelená LED svítí, pokud je sepnuté relé RE2
- LD3 ... zelená LED svítí, pokud je sepnuté relé RE3
- LD4 ... bílá LED svítí nebo bliká, pokud modul vysílá (TXD) – další využití viz sw.
- LD5 ... červená LED svítí – program běží (RUN) – další využití viz sw.
- LD6 ... modrá POWER LED (přítomné napájení).
- LD7 ... žlutá LED svítí – IN1 aktivní (mezi vstupem IN1+ a IN1- je napětí).
- LD8 ... žlutá LED svítí – IN2 aktivní (mezi vstupem IN2+ a IN2- je napětí).
- LD9 ... žlutá LED svítí – IN3 aktivní (mezi vstupem IN3+ a IN3- je napětí).

Připojení vstupů:



Konektor JP2: 4 analogové vstupy (na prvních 2 jsou připojeny interní trimry P1 a P2 a je možné využít těchto trimrů-výstupů ve spojení s jiným modulem, zbylé 2 analogové vstupy je možné využít ke snímání dalších 2 externích potenciometrů nebo měření 2 napětí – maximální vstupní napětí je 5V – nepřekračujte. Pro vyšší rozsah zapojte odporový dělič).

K analogovým vstupům v programu přístup prostřednictvím příkazů s proměnnými P0 až P9. Rozsah vstupního napětí 0 až 5V.

Popis pinů JP2 (odshora dolů):

- Pin 1 ... napájení potenciometrů (čidel), proudově omezeno rezistorem 100R.
- Pin 2 ... výstup trimru P1 – nepřipojujte jiný zdroj napětí nebo jiný potenciometr.
- Pin 3 ... výstup trimru P2 – rovněž nepřipojujte jiný výstup - je možné pouze snímat..
- Pin 4 ... analogový vstup AN2 (v programu viz AN2).
- Pin 4 ... analogový vstup AN3 (v programu viz AN3).
- Pin 6 ... neosazen (klíč).
- Pin 7 ... GND

JP3 ... UART1 – komunikace s PC (USB adaptér) nebo s jiným modulem, 9600bps

JP4 – UART2 - 9600bps, univerzální TTL rozhraní.

JP1 – jumper, využití v programu v podmínkách IF ... THEN

IN4, IN5 – tlačítka pro využití v programu nebo v PC komunikaci s modulem.

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>

1.1. Základní příkazy pro editaci a nahrávání programu – pouze ve STOP režimu

LIST <CR> ... vypíše obsah paměti z RAM, viz naposledy nahraný program (pro lepší srozumitelnost CR dále nebude psáno, ale bude vyžadováno – všechny příkazy musí končit znakem 0x0D).

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

R1=1 ... sepne relé RE1.

R2=1 ... sepne relé RE2.

R3=1 ... sepne relé RE3.

R1=0 ... vypne relé RE1.

R2=0 ... vypne relé RE2.

R3=0 ... vypne relé RE3.

(více informací o ovládacích příkazech – viz níže).

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. Při zastavení se vypnou všechna relé (stopstav)

Pokud program běží, svítí červená 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 – V2.0 *** <CR> <LF>

H? ... vrátí stav všech vstupů a výstupů v pořadí (IN1, IN2, IN3, IN4, IN5, JP1, RE1, RE2, RE3, LD5) ve tvaru: *0000000000 <CR> <LF>

Tímto příkazem je možné číst stav v době, kdy program neběží, tj. modul je využíván v režimu USB relé (nastavení výstupů a relé přímými příkazy – viz 1.1.).

V autonomním režimu jsou vstupy a výstupy včetně UART1 a UART2 přesměrovány na samotný program a po ukončení programu nebo jeho zastavení (tlačítko RUN/STOP) jsou všechny výstupy nulovány (relé vypnuta).

POZN: <CR> je 0x0D a <LF> 0x0A

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 EEPROM žádný program, nebude po zapnutí modulu nic spuštěno (LD5 zhasnuta).

EELoad ... kdykoliv, pokud není modul v režimu RUN, 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.

2.1. Rozdělení programových příkazů

1. Nepodmíněné skoky.
2. Podmíněné skoky s relačními operátory.
3. Přiřazovací příkazy pro výstupy/relé R1 až R3 a pomocné bity R4 až R8, dále R0 a R9.
4. Přiřazení číselné hodnoty nebo vstupu proměnným P0 až P9.
5. Nastavení řídicích registrů M, X a K číselnou hodnotou nebo proměnnou.
6. Čtení řídicích registrů M, X, K.
7. Aritmetické příkazy s proměnnými P0 až P9.
8. Výpis textů a obsahu proměnných na porty UART1 a UART2.

2.2. 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.

2.3. Podmíněný skok a relační operátory

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 řádce 110 splněna, bude proveden skok na řádek 320, jinak bude vykonán řádek následující.

Typy podmínek:

1. Testování vstupů IN1 až IN3, tlačítek IN4 a IN5, jumperu JP1.
2. Testování výstupů R1 až R3 a pomocných bitů R4 až R8 a R0 či R9.
3. Porovnávání proměnných P0 až P9 mezi sebou.

Testování vstupů

@A=1 ... Pokud je vstup IN1 aktivní (v log. 1, svítí LED LD7), je podmínka splněna.

@A=0 ... Pokud je vstup IN1 neaktivní (LED LD7 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 LD8), je podmínka splněna.

@B=0 ... Vstup IN2 neaktivní (LED LD8 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 LD9), je podmínka splněna.

@C=0 ... Vstup IN3 neaktivní (LED LD9 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.

@J=1 ... Pokud je jumper JP1 nezapojen (volný), podmínka je splněna.

@J=0 ... Je-li JP1 spojen (zkratovací propojka), podmínka je splněna.

Testování výstupů

@1=1 ... Pokud je relé RE1 sepnuto (svítí LD1), je podmínka splněna.

@1=0 ... Pokud je relé RE1 vypnuto, podmínka je splněna.

@1=R... Relé RE1 cvaklo nebo právě seplo, podmínka je splněna. Příznakový bit pro RE1 je čtením této podmínky vynulován, aby nedocházelo k jeho akumulování.

@2=1 ... Pokud je relé RE2 sepnuto (svítí LD2), podmínka je splněna.

@2=0 ... Pokud je relé RE2 vypnuto, podmínka je splněna.

@2=R... Relé RE2 cvaklo nebo právě seplo, podmínka je splněna. Příznakový bit pro RE2 je čtením této podmínky vynulován, aby nedocházelo k jeho akumulování.

@3=1 ... Pokud je relé RE3 sepnuto (svítí LD3), podmínka je splněna.

@3=0 ... Pokud je relé RE3 vypnuto, podmínka je splněna.

@3=R... Relé RE3 cvaklo nebo právě seplo, podmínka je splněna. Příznakový bit pro RE3 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.

2.4. Přiřazovací příkazy pro výstupy a pomocné bity

100 R0=1 <CR> ... rozsvítí LD5 (RUN/STOP – LD5 tímto ztrácí význam indikace běžícího programu, LED LD5 zhasne zastavením programu, tj. například stiskem tlačítka RUN).

110 R1=1 <CR> ... sepne relé RE1 (svítí LD1).

120 R2=1 <CR> ... sepne relé RE2 (svítí LD2).

130 R3=1 <CR> ... sepne relé RE3 (svítí LD3)..

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

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

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

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

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

190 R9=1 <CR> ... přepne výstup TXD (UART1) na obyčejný I/O výstup a nastaví jej do logické 1 (bílá LED LD4 zhasne)..

200 R0=0 <CR> ... zhasne LD5 (RUN/STOP – LD5 tímto ztrácí význam indikace běžícího programu).

210 R1=0 <CR> ... vypne relé RE1 (LD1 zhasne).

220 R2=0 <CR> ... vypne relé RE2 (LD2 zhasne).

230 R3=0 <CR> ... vypne relé RE3 (LD3 zhasne).

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

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

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

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

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

290 R9=0 <CR> ... přepne výstup TXD (UART1) na obyčejný I/O výstup a nastaví jej do logické 0 (bílá LED LD4 svítí)..

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

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

320 R\$=T <CR> ... současně invertuje výstupy R1 až R3 a pomocné bity R4 až R8.

400 R0=T <CR> ... přepne LD5 (RUN/STOP – LD5 tímto ztrácí význam indikace běžícího programu, LED LD5 zhasne zastavením programu, tj. například stiskem tlačítka RUN).

410 R1=T <CR> ... přepne (změní stav) relé RE1 (svítí LD1).

420 R2=T <CR> ... přepne relé RE2 (svítí LD2).

430 R3=T <CR> ... přepne relé RE3 (svítí LD3)..

440 R4=T <CR> ... invertuje (Toggle) pomocný bit R4.

450 R5=T <CR> ... invertuje pomocný bit R5.

460 R6=T <CR> ... invertuje pomocný bit R6.

470 R7=T <CR> ... invertuje pomocný bit R7.

480 R8=T <CR> ... invertuje pomocný bit R8.

490 R9=T <CR> ... přepne výstup TXD (UART1) na obyčejný I/O výstup a invertuje jej (bílá LED LD4 změní stav - přepne).

2.5. Přiřazovací příkazy pro proměnné P0 až P9

Přiřazení přímých číselných hodnot a 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 stavu trimrů P1 a P2 a analogových vstupů AN2 a AN3

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

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).

2.6. 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.

2.7. Výpis 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.

2.8. Nastavení řídicích registrů K1 až K3

Přiřazení přímých číselných hodnot a konstant řídicím registrům K1 až K3

100 K1=10 <CR> ... přiřazení konstanty 10 do MKO1 (filtr impulsů na vstupu IN1 – 10 setin, tj. 0.1 vteřiny, impulsy po sobě následují v kratším intervalu než 0.1 vteřiny budou vyhodnoceny jako jeden souvislý impuls).

200 K2=2000 <CR> ... zapíše 2000 do MKO2 (filtr impulsů na IN2) – impulsy kratší než 2 vteřiny budou vyhodnoceny jako jeden souvislý impuls.

300 K3=2 ... zapíše 2 do MKO3 (filtr impulsů na IN3) – impulsy kratší než 20ms budou vyhodnoceny jako jeden souvislý impuls.

POZN: Po zapnutí programu jsou MKO1 až MKO3 nastaveny na 17 (filtrační čas 170ms).

Nepřímá přiřazení proměnných do řídicích registrů K1 až K3

400 K1=P7 <CR> ... hodnotu proměnné P7 zkopíruj do K1 (MKO1), P7 beze změny.

410 K2=P2 <CR> ... hodnotu proměnné P2 zkopíruj do K2 (MKO2), P2 beze změny.

420 K3=P1 <CR> ... hodnotu proměnné P1 zkopíruj do K3 (MKO3), P1 beze změny.

2.9. Nastavení řídicích registrů M1 až M5

Přiřazení přímých číselných hodnot a konstant řídicím registrům M1 až M5

100 M1=100 <CR> ... přiřazení konstanty 100 do časovače M1 vstupu IN1.

200 M2=0 <CR> ... nulování časovače M2 vstupu IN2 (M1 se automaticky nuluje po zahájení pulsu na vstupu IN2 a probíhá časování, dokud je IN2 sepnutý a po odeznění impulsu zůstává v časovači M2 stav – doba impulsu IN2 v setinách, např. 120 odpovídá vteřinovému pulsu + konstantě K2=20 MKO2, která určuje dočasování).

300 M5=0 ... nulování časovače M5 pro tlačítko IN5 (čtením registru M5 lze snadno měřit dobu držení tlačítka).

Nepřímá přiřazení proměnných do řídicích registrů M1 až M5

400 M1=P7 <CR> ... hodnotu proměnné P7 zkopíruj do M1, P7 beze změny.

410 M3=P2 <CR> ... hodnotu proměnné P2 zkopíruj do M3, P2 beze změny.

420 M4=P1 <CR> ... hodnotu proměnné P1 zkopíruj do M4, P1 beze změny.

2.10. Nastavení řídicích registrů X1 až X9

Přiřazení přímých číselných hodnot a konstant řídicím registrům X1 až X9

100 X1=100 <CR> ... přiřazení konstanty 100 do řídicího registru X1.

200 X2=0 <CR> ... nulování registru X2 vstupu IN2 (X2 se automaticky naplní po odeznění impulsu na vstupu IN2, tj. časovač M2 se po ukončení impulsu přepíše do registru X2).

300 X5=0 ... nulování registru X5 (v programové smyčce je pak snadné čekat na stisk a uvolnění tlačítka IN5 a dále vyhodnocovat dobu stisku, tj. držení tlačítka až po jeho uvolnění).

POZN: Rozdíl mezi M1 a X1 je v tom, že M1 je časovač, kdežto X1 se naplní až po odeznění impulsu.

Nepřímá přiřazení proměnných do řídicích registrů X1 až X5

400 X1=P7 <CR> ... hodnotu proměnné P7 zkopíruj do X1, P7 beze změny.

410 X3=P2 <CR> ... hodnotu proměnné P2 zkopíruj do X3, P2 beze změny.

420 X5=P1 <CR> ... hodnotu proměnné P1 zkopíruj do X5, P1 beze změny.

Speciální řídicí registry X6 až X9

Zvláštní význam mají řídicí registry X6 až X9.

X9 ... LED LD5 bude blikat podle hodnoty uložené v X9 (X9=0 ... funkce vypnuta).

X8 ... každých 10ms sníží o 1 (dekrementace), při dosažení 0 nahraje do X8 stav X9 a změní stav LD5 (invertuje výstup).

X7 ... LED LD4 bude blikat podle hodnoty uložené v X7 (X7=0 ... funkce vypnuta).

X6 ... každých 10ms sníží o 1 (dekrementace), při dosažení 0 nahraje do X6 stav X7 a změní stav LD4 (invertuje výstup). Výstup se ovšem přepne jedině tehdy, byl-li nastaven jako I/O.

500 X9=10 <CR> ... přiřazení konstanty 10 do X9 (LD5 bude blikat 5x za vteřinu).

510 X7=50 <CR> ... zapíše 50 do X7 (LD4 blikne 1x za vteřinu).

520 X9=0 <CR> ... vypne funkci blikání LD5.

POZN: Po zapnutí programu X1 až X9 vynulovány.

2.11. Čtení řídicích registrů K1 až K3, M1 až M5, X1 až X9

500 P2=M1 <CR> ... obsah M1 zapíše do P2, M1 beze změny.

510 P3=K1 <CR> ... obsah K1 zapíše do P3, K1 beze změny.

520 P5=X3 <CR> ... obsah X3 zapíše do P5, X3 beze změny.

2.12. Pozastavení programu na krátký čas

600 WAIT=100 <CR> ... zastav vykonávání programu na 1 vteřinu (100 setin).

610 WAIT=10 <CR> ... zastav program na 100ms.

620 WAIT(20) <CR> ... jiný formát zápisu, zastaví program na 200ms.

620 WAIT=P1 <CR> ... zastaví program na dobu určenou obsahem proměnné P1.

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 relé RE2 podle napětí na vstupu AN3, je-li AN3 vyšší než 100, sepne RE1, jinak vypne

```
10 P1=100
20 P2=#3
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
```

3.4 Měří šířku vstupního impulsu na IN1 (filtr K1) a pokud je impuls delší než limit P1 a kratší než limit P2 (natočení trimru P2 určujedobu sepnutí relé RE3), zapne reléový automat. Tutěž funkci plní držení tlačítka IN4. Dle natočení P1 bliká LD5, dle P2 bliká LD4. Tlačítko IN5 slouží k ověření funkce automatu.

```
10 K1=40
20 P1=2
30 P2=10
40 P3=10
50 R9=0
80 X1=0
90 X4=0
100 P4=#0
110 P4+12
120 P6=P4/P1
130 X9=P6
140 P4=P4*P2
150 P5=#1
160 P5+12
170 P6=P5/P1
180 X7=P6
190 P5=P5*P3
200 IF @E=R THEN 300
210 P6=X4
220 IF P6<P4 THEN 250
230 IF P6>P5 THEN 250
240 GOTO 300
250 P6=X1
260 IF P6<P4 THEN 100
270 IF P6>P5 THEN 100
300 R1=1
310 WAIT=50
320 R2=1
330 WAIT=50
340 R3=1
350 WAIT=P5
360 R3=0
370 WAIT=50
380 R2=0
390 WAIT=50
400 R1=0
410 GOTO 80
```

www.selfcontrol.cz