Dílenská praxe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A4** | 6. Maticový displej | | | |
| Macháček Daniel | |  | 1/7 | Známka: |
| 1.2.2016 | | Datum odevzdání: | 22.2.2017 | Odevzdáno: |

Zadání:

Zpracujte program v programovacím jazyce JSA MCS51 ovládající určený připojený maticový displej a klávesnici tak, aby obsahoval nejméně tyto funkce:

1. Stisknuté tlačítko klávesnice se uloží do paměti MCS51, minimálně 5 kláves, maximálně 15 kláves
2. Každé klávese bude přiřazen vhodný zobrazovaný symbol
3. Rozpoznání stavu vkládání znaku a stavu přehrávání vložených znaků ovládaných pomocí maticové klávesnice
4. Přepínání mezi těmito stavy
5. Využití všech HW možností přípravku MCS51

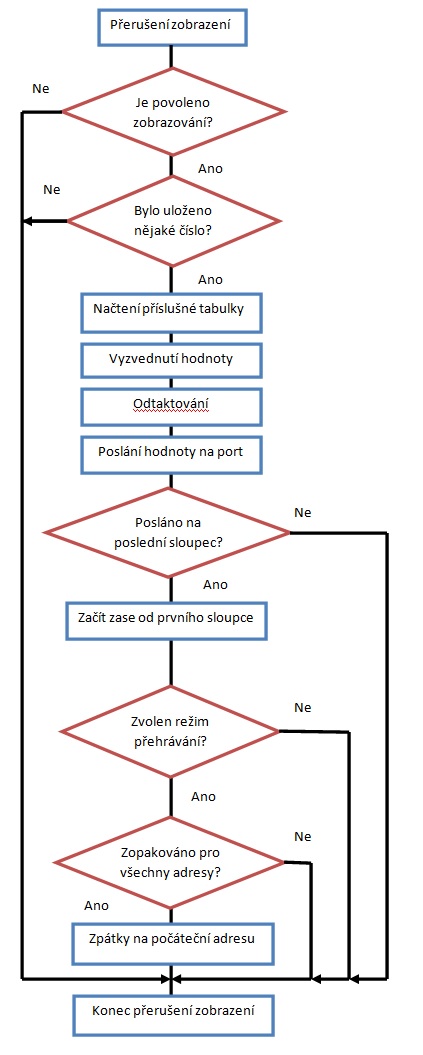
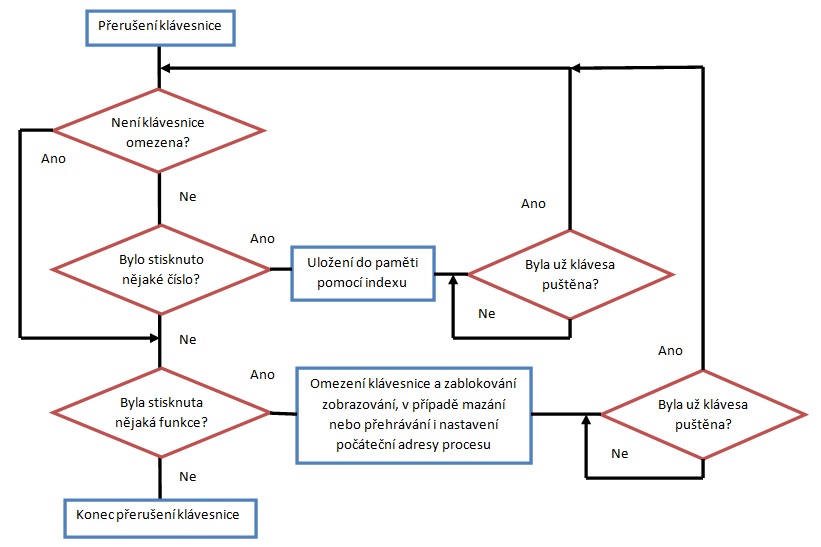
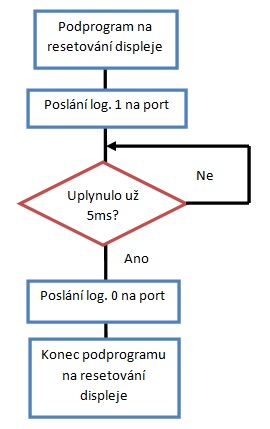
Závěr:

Tento program nebyl testován přímo na přípravku, ale věnoval jsem tvorbě algoritmu mnoho času a věřím, že je program plně funkční. Původní verze programu využívala pouze jednoho přerušení a byla značně kratší než finální verze, ovšem, z důvodu ochrany „přeplnění“ paměti mým drobným vylepšením, jsem musel do programu zakomponovat o něco více podmínek, než bylo původně nutné. To vedlo ke značnému znepřehlednění programu, ale na svojí jednoduchosti neztratil. Nakonec jsem byl donucen využít i tedy druhého přerušení z důvodu zabezpečení ovládání klávesnice a následného ukládání znaků do paměti.

Přílohy:

* Výpis programu – 6 stran
* Vývojové diagramy – 2 strany

**Vývojové diagramy k JSA:**



**Výpis programu JSA:**

org 0 ;Začátek programu

jmp setintr ;Skočí na výchozí konfiguraci

org 0Bh ;Adresa prvního přerušení 0Bh

jmp intr ;Skočí na rozsvěcení displeje

org 1Bh ;Adresa prvního přerušení 1Bh

jmp klav ;Skočí na ovládání klávesnice

cislo1: DB 1111111b, 1111011b, 1111101b, 0000000b, 1111111b

cislo2: DB 0111001b, 0011110b, 0101110b, 0110110b, 0111001b

cislo3: DB 1001001b, 0111110b, 0111110b, 0110110b, 1001001b

cislo4: DB 1110011b, 1110101b, 1110110b, 0000011b, 1110111b

cislo5: DB 1010000b, 0110110b, 0110110b, 0110110b, 1000110b

cislo6: DB 1000001b, 0110110b, 0110110b, 0110110b, 1001101b

cislo7: DB 1111110b, 1111110b, 0000110b, 1111010b, 1111100b

cislo8: DB 1001001b, 0110110b, 0110110b, 0110110b, 1001001b

cislo9: DB 0111001b, 0110110b, 0110110b, 0110110b, 1000001b

znakx: DB 0011100b, 1101011b, 1110111b, 1101011b, 0011100b

casintr1 EQU 40535 ;Čas pro přerušení 1

casintr2 EQU 30535 ;Čas pro přerušení 2

setintr:

mov DPTR, #8000h

movc A, @A+DPTR

mov TMOD, #00010001b ;Zapíná časovače v módu 1

mov IE, #10001010b ;Povoluje využití přerušovacího systému

mov TH0, #high(casintr1) ;Nastaví horní část časovače hodotou casintr1

mov TL0, #low(casintr1) ;Nastaví spodní část časovače hodotou casintr1

mov TH1, #high(casintr2) ;Nastaví horní část časovače hodotou casintr2

mov TL1, #low(casintr2) ;Nastaví spodní část časovače hodotou casintr2

mov R0, #0Ch ;výchozí adresa pro ukládání

mov R2, #1d ;Podmínková hodnota

mov R3, #0d ;Vyčistění registru pro posílání na port

mov R4, #1d ;Podmínková hodnota

mov R5, #1d ;Podmínková hodnota

mov R6, #250d ;Hodnota pro aktivaci resetu

mov R7, #5d ;Hodnota pro aktivaci resetu

call clear ;Resetování sloupců displeje

setb TR0 ;Spustí časovač 0

setb TR1 ;Spustí časovač 1

main:

jmp $ ;Zacyklení programu

clear:

setb P3.7 ;Nastartování pomocí clocku

zp:

djnz R6, zp ;Počkání nejméně 5ms pomocí smyček

mov R6, #250d ;z důvodu nedostatku časovačů

djnz R7, zp

clr P3.7 ;Dokončení taktu

ret ;Ukončení podprogramu

klav:

cjne R2, #1d, nahr ;Test zda-li není v provozu přehrávání hodnot

cjne R5, #1d, prehr ;Test zda-li nedochází k čištění paměti

clr P1.5 ;Na první sloupec klávenice pošle nulu

setb P1.6 ;Na druhý sloupec klávenice pošle jedničku

setb P1.7 ;Na třetí sloupec klávenice pošle jedničku

jnb P1.3, cislo1 ;Pokud na prvním řádku bude nula, skočí na návěští cislo1

jnb P1.2, cislo4 ;Pokud na druhém řádku bude nula, skočí na návěští cislo4

jnb P1.1, cislo7 ;Pokud na třetím řádku bude nula, skočí na návěští cislo7

prehr:

clr P1.5 ;Na první sloupec klávenice pošle nulu

setb P1.6 ;Na druhý sloupec klávenice pošle jedničku

setb P1.7 ;Na třetí sloupec klávenice pošle jedničku

jnb P1.0, preh ;Pokud na čtvrtém řádku bude nula, skočí na návěští preh

cjne R5, #1d, cisti ;Pokud nebyla zvolena žádná dostupná akce skočí na cisti

setb P1.5 ;Na prvnísloupec klávenice pošle jedničku

clr P1.6 ;Na druhý sloupec klávenice pošle nulu

setb P1.7 ;Na třetí sloupec klávenice pošle jedničku

jnb P1.3, cislo2 ;Pokud na prvním řádku bude nula, skočí na návěští cislo2

jnb P1.2, cislo5 ;Pokud na druhém řádku bude nula, skočí na návěští cislo5

jnb P1.1, cislo8 ;Pokud na třetím řádku bude nula, skočí na návěští cislo8

cisti:

setb P1.5 ;Na prvnísloupec klávenice pošle jedničku

clr P1.6 ;Na druhý sloupec klávenice pošle nulu

setb P1.7 ;Na třetí sloupec klávenice pošle jedničku

jnb P1.0, cislo0 ;Pokud na čtvrtém bude nula, skočí na návěští cislo0

cjne R5, #1d, Konklav ;Pokud nebyla zvolena žádná dostupná akce skočí na Konklav

setb P1.5 ;Na první sloupec klávenice pošle jedničku

setb P1.6 ;Na druhý sloupec klávenice pošle jedničku

clr P1.7 ;Na třetí sloupec klávenice pošle nulu

jnb P1.3, cislo3 ;Pokud na prvním řádku bude nula, skočí na návěští cislo3

jnb P1.2, cislo6 ;Pokud na druhém řádku bude nula, skočí na návěští cislo6

jnb P1.1, cislo9 ;Pokud na třetím řádku bude nula, skočí na návěští cislo9

nahr:

setb P1.5 ;Na první sloupec klávenice pošle jedničku

setb P1.6 ;Na druhý sloupec klávenice pošle jedničku

clr P1.7 ;Na třetí sloupec klávenice pošle nulu

jnb P1.0, nah ;Pokud na čtvrtém bude nula, skočí na návěští nah

Konklav:

mov TH1, #high(casintr2) ;Nastaví horní část časovače hodotou casintr2

mov TL1, #low(casintr2) ;Nastaví spodní část časovače hodotou casintr2

reti ;Ukončení přerušení klávesnice

c123:

jb P1.3, klav ;Pokud není klávesa stisknutá, skočí na návěští klav

jmp c123 ;Skočí na návěští c123

c456:

jb P1.2, klav ;Pokud není klávesa stisknutá, skočí na návěští klav

jmp c456 ;Skočí na návěští c456

c789:

jb P1.1, klav ;Pokud není klávesa stisknutá, skočí na návěští klav

jmp c789 ;Skočí na návěští c789

cp0:

jb P1.0, klav ;Pokud není klávesa stisknutá, skočí na návěští klav

jmp cp0 ;Skočí na návěští cp0

cislo1:

mov @R0, #1d ;Při stisknutí tl.1 uloží na aktuální adresu R0 její index

inc R0 ;Posune se na další adresu

cjne R0, #15h, c123 ;Pokud není u konce vymezené paměti skáče na test stisknuté klávesy

mov @R0, #10d ;Pokud ano, uloží na konečnou adresu nevyšší index

mov R5, #2d ;Zamezí nahrávat další znaky do paměti

jmp c123 ;Skočí na na test stisknuté klávesy

cislo2:

mov @R0, #2d

inc R0

cjne R0, #15h, c123

mov @R0, #10d

mov R5, #2d

jmp c123

cislo3:

mov @R0, #3d

inc R0

cjne R0, #15h, c123

mov @R0, #10d

mov R5, #2d

jmp c123

cislo4:

mov @R0, #4d

inc R0

cjne R0, #15h, c456

mov @R0, #10d

mov R5, #2d

jmp c456

cislo5:

mov @R0, #5d

inc R0

cjne R0, #15h, c456

mov @R0, #10d

mov R5, #2d

jmp c456

cislo6:

mov @R0, #6d

inc R0

cjne R0, #15h, c456

mov @R0, #10d

mov R5, #2d

jmp c456

cislo7:

mov @R0, #7d

inc R0

cjne R0, #15h, c789

mov @R0, #10d

mov R5, #2d

jmp c789

cislo8:

mov @R0, #8d

inc R0

cjne R0, #15h, c789

mov @R0, #10d

mov R5, #2d

jmp c789

cislo9:

mov @R0, #9d

inc R0

cjne R0, #15h, c789

mov @R0, #10d

mov R5, #2d

jmp c789

cislo0: ;0 - čístí paměť

mov R4, #2d ;Zamezí přehrávání z paměti

mov R2, #3d ;Zamezí používání klávesnice

mov R0, #0Ch ;Nastaví počáteční adresu na 0Ch

maz:

mov @R0, #0d ;Do příslučné adresy R0 zapíše 0

inc R0 ;Posune se o adresu dál

cjne R0, #16h, maz ;Opakuje dokud nebylo vše vyčištěno

mov R0, #0Ch ;Vrací se na počáteční adresu

mov R2, #1d ;Povoluje používání klávesnice

mov R4, #1d ;Povoluje přehrávání z databáze

mov R5, #1d ;Odstraňuje varovný symbol

ljmp cp0 ;Dlouhý skok na kontrolu stisknuté klávesy

preh:

mov R2, #2d ;Zamezí používání klávesnice

mov R5, #1d ;Odstraňuje varovný symbol

mov R0, #0Ch ;Nastaví počáteční adresu na 0Ch

ljmp cp0 ;Dlouhý skok na kontrolu stisknuté klávesy

nah:

mov R2, #1d ;Povoluje používání klávesnice

ljmp cp0 ;Dlouhý skok na kontrolu stisknuté klávesy

intr:

mov TH0, #high(casintr1) ;Nastaví horní část časovače hodotou casintr1

mov TL0, #low(casintr1) ;Nastaví spodní část časovače hodotou casintr1

cjne R4, #1d, Konintr ;Zamezení zobrazování - čištění paměti - skok na Konintr

c1:

cjne @R0, #1d, c2 ;Porovná index v příslušné adrese, pokud není roven 1d pokračuje skokem na c2

mov DPTR,#cislo1 ;Načtení tabulky čísla 1

jmp Vypis ;Skok na zobrazení uloženého čísla

c2:

cjne @R0, #2d, c3

mov DPTR,#cislo2

jmp Vypis

c3:

cjne @R0, #3d, c4

mov DPTR,#cislo3

jmp Vypis

c4:

cjne @R0, #4d, c5

mov DPTR,#cislo4

jmp Vypis

c5:

cjne @R0, #5d, c6

mov DPTR,#cislo5

jmp Vypis

c6:

cjne @R0, #6d, c7

mov DPTR,#cislo6

jmp Vypis

c7:

cjne @R0, #7d, c8

mov DPTR,#cislo7

jmp Vypis

c8:

cjne @R0, #8d, c9

mov DPTR,#cislo8

jmp Vypis

c9:

cjne @R0, #9d, c0

mov DPTR,#cislo9

jmp Vypis

c0:

cjne @R0, #10d, Konintr

mov DPTR,#znakx

jmp Vypis

Vypis:

mov A, R3 ;Ukazatel tabulky přesune do A

movc A, @A+DPTR ;Načte hodnotu z tabulky dle ukazatele do A

setb P3.7 ;Takt - 1

mov P3, A ;Poslání načtené hodnoty

clr P3.7 ;Takt - 0

inc R3 ;Posunutí ukazatele

cjne R3, #5d, Konintr ;Pokud ukazatel došel na konec tabulky

mov R3, #0d ;je zde vynulován

cjne R5, #1d, Konintr ;Kontrola zda je povoleno přehrávání

cjne R2, #2d, Konintr ;Kontrola zda je povoleno přehrávání

inc R0 ;Posunutí adresy pro přehrávání

cjne R0, #15h, Konintr ;Pokud adresa dosáhla konce paměti

mov R0, #0Ch ;je navrácena na začátek

Konintr:

reti ;Ukončení přerušení zobrazování

end