Dílenská praxe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A4** | Maticová displej | | | |
| Dobeš Daniel | |  | 1/1 | Známka: |
| 28. 11. 2018 | | Datum odevzdání: | 9. 1. 2019 | Odevzdáno: |

Zadání:

Zpracujte program v programovacím jazyce JSA ATMega128 ovládající určený připojený maticový displej a klávesnici tak, aby obsahoval nejméně tyto funkce:

1. stisknuté tlačítko klávesnice se uloží do paměti modulu MB\_ATmega128, minimálně 5 kláves, maximálně 15 kláves
2. každé klávese bude přiřazen vhodný zobrazovaný symbol
3. rozpoznání stavu vkládání znaků a stavu přehrávání vložených znaků ovládaných pomocí maticové klávesnice
4. přepínaní mezi těmito režimy
5. využití všech vhodných HW možností přípravku MB-ATmega128.

Stavový diagram:

Viz příloha

Postup pro uživatele:

1. Uživatel vybere pomocí symbolů „\*“ nebo „#“ režim zobrazování nebo režim vkládání znaků
2. Při stisku „#“ se na displeji rozsvítí „X“ označení, že klávesy, které zmáčkne se uloží do paměti
3. Při stisku „\*“ se na displeji začne postupně střídat posledních 5 kláves, které byl zmáčknuty.

Výpis programu:

Viz příloha

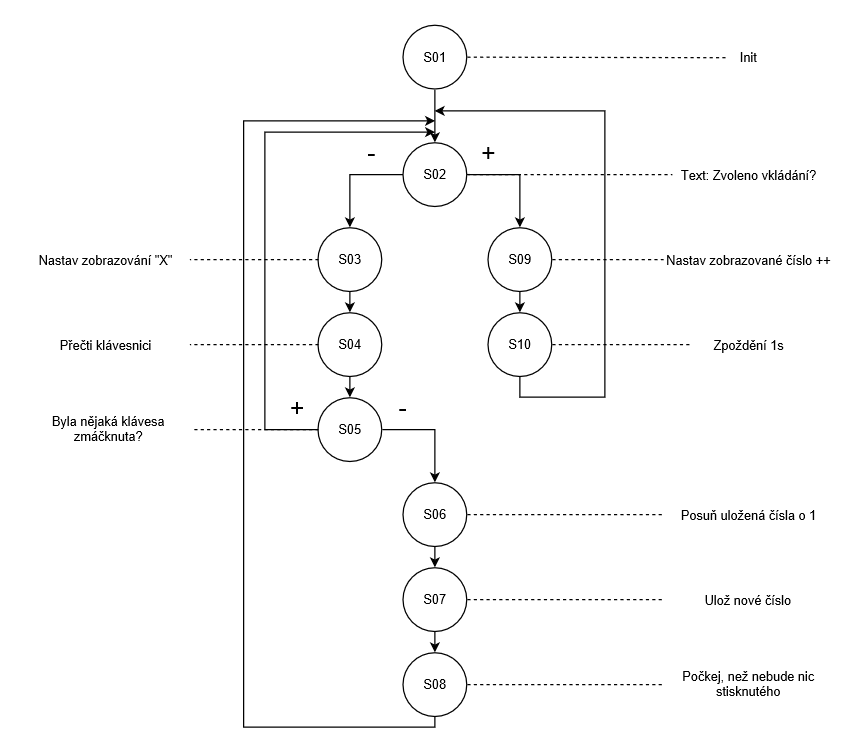
Závěr:

Při psaní kódu jsem se snažil vymyslet zajímavý způsob, jak vyřešit možnost, když uživatel zmáčkne jiný počet kláves než přímo určitý počet. Nejvhodnější mi přišlo zobrazovat posledních 5 kláves postupně od té která byl zaznamenána naposledy. Pro způsob rozlišení čtení a vkládání znaků jsem zvolil znak „X“, který svítí během režimu pro vkládání znaků na displeji.

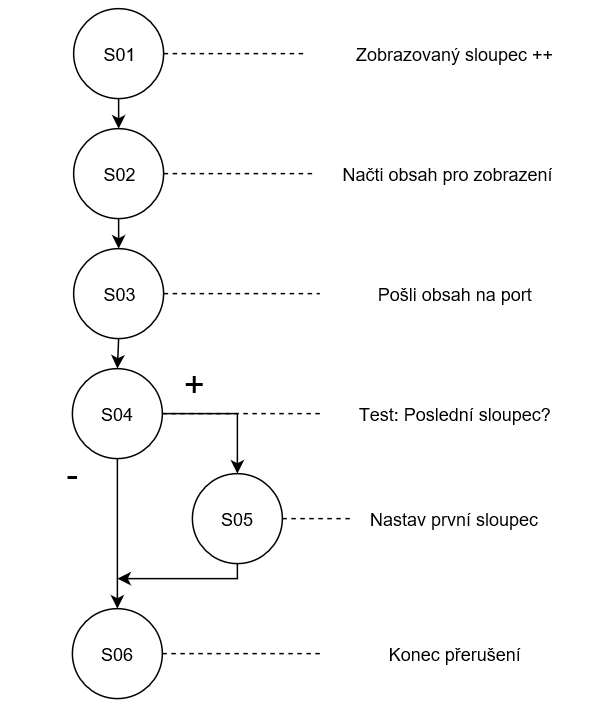
Přílohy:

* Stavové diagramy (1 strana)
* Výpis programu (8 stran)

**Stavové diagramy:**



Interrupt:



**Výpis programu:**

.Nolist

;Include File:

.Include "m128def.inc"

.List

.ORG 0x0000

JMP Initialization

.ORG 0x001E ;Timer0 interrupt - Displaying

JMP MatrixDisplay

.ORG 0x0046

.CSEG

;Definition register and expression

.DEF temp1 = R16

.DEF Key = R17

.DEF index1 = R18 ;cycle counter (delay) & displacement

.DEF index2 = R19 ;cycle counter (delay)

.DEF index3 = R20 ;cycle counter (delay)

.DEF index4 = R21 ;position of saved numbers

.DEF index5 = R22 ;position of displaying column

.EQU ddrKey = ddrB

.EQU portKey = PortB

.EQU pinKey = PinB

.EQU ddrDisplay = ddrA

.EQU portDisplay = PortA

.EQU Timer0Setting = 0b0000\_0110

.EQU TimskSetting = 0b0000\_0010

.EQU MAX = 0xFF

.EQU MIN = 0x00

.EQU FirstHalf = 0xF0

.EQU SecondHalf = 0x0F

.EQU Column1 = 0b1111\_1110

.EQU Column2 = 0b1111\_1101

.EQU Column3 = 0b1111\_1101

.EQU Column4 = 0b1111\_0111

.EQU One = 0b1110\_1110

.EQU Two = 0b1110\_1101

.EQU Three = 0b1110\_1011

.EQU Four = 0b1101\_1110

.EQU Five = 0b1101\_1101

.EQU Six = 0b1101\_1011

.EQU Seven = 0b1011\_1110

.EQU Eight = 0b1011\_1101

.EQU Nine = 0b1011\_1011

.EQU Star = 0b0111\_1110

.EQU Zero = 0b0111\_1101

.EQU Hash = 0b0111\_1011

Initialization:

;Reserve place for saved numbers

.DSEG

SavedNumbers:.BYTE 6 ;First is used for displaying X

SavedNumbersEnd:

.CSEG

;Stack pointer setting

LDI temp1, LOW(RAMEND)

OUT SPL, temp1

LDI temp1, HIGH(RAMEND)

OUT SPH, temp1

;Port setting

LDI temp1, SecondHalf

OUT ddrKey, temp1

LDI temp1, MAX

OUT ddrDisplay, temp1

;Reset - counter in Matrix display

SBI portDisplay, 7

CALL delay1ms ;long impuls reset

CBI portDisplay, 7

;Choose first adress with first saved number

LDI XL, LOW(SavedNumbers)

LDI XH, HIGH(SavedNumbers)

LDI index4, (SavedNumbersEnd-SavedNumbers-1)

;Save neutral number "X" (10)

LDI temp1, 10

ST X, temp1

;Set variable to control column

LDI index5, 5

;First set Displaying Mode

CLT

;Timer0 & Timsk configuration

LDI temp1, Timer0Setting

OUT TCCR0, temp1

IN temp1, Timsk

ORI temp1, TimskSetting

OUT Timsk, temp1

;Global interrupt enabled

SEI

Program:

;Mod check (Reading/Displaying)

CALL MatrixKeypad

BRTS Reading

Displaying: ;T = 0

CALL NumberReplacement ;change displaying number

CALL Delay1s

RJMP ProgramEnd

Reading: ;T = 1

CALL DisplayX ;it shows that the reading mode is active

CALL MatrixKeypad

CPI Key, MAX

BREQ ProgramEnd

CALL DisplacementSavedNumbers ;Position saved number ++

LDI XL, LOW(SavedNumbers)

LDI XH, HIGH(SavedNumbers)

ST X, Key ;Store press key

NoKeyDetected: ;Waiting for no key

CALL MatrixKeypad

CPI Key, MAX

BRNE NoKeyDetected

ProgramEnd:

RJMP Program

MatrixDisplay: ;Every cca 1 ms - generate by interrupt - Timer 0

PUSH temp1 ;Backup

SBI PortDisplay, 7 ;Short impuls -> shift displaying column

NOP NOP NOP NOP ;Delay 200 microsec

CBI PortDisplay, 7 ;End of impuls

LPM temp1, Z+ ;Source for each number combination

OUT PortDisplay, temp1

DEC index5

BREQ SetPointerZ ;Reset Z pointer if last column have displayed

POP temp1

RETI ;Back from interrupt

NumberReplacement: ;Every 1 s

;Set next number to display

LD temp1, X+

DEC index4

BRNE NRend

LDI XL, LOW(SavedNumbers+1)

LDI XH, HIGH(SavedNumbers+1)

LDI index4, (SavedNumbersEnd-SavedNumbers-2)

NRend:

RET

SetPointerZ: ;Subprogram

LD temp1, X ;Load number which is displaying

N0: CPI temp1, 0 ;is saved number "0"?

BRNE N1 ;no, try next

LDI ZL, LOW(Number0\*2) ;yes, set Z pointer

LDI ZH, HIGH(Number0\*2)

RJMP SPZend

N1: CPI temp1, 1

BRNE N2

LDI ZL, LOW(Number1\*2)

LDI ZH, HIGH(Number1\*2)

RJMP SPZend

N2: CPI temp1, 2

BRNE N3

LDI ZL, LOW(Number2\*2)

LDI ZH, HIGH(Number2\*2)

RJMP SPZend

N3: CPI temp1, 3

BRNE N4

LDI ZL, LOW(Number3\*2)

LDI ZH, HIGH(Number3\*2)

RJMP SPZend

N4: CPI temp1, 4

BRNE N5

LDI ZL, LOW(Number4\*2)

LDI ZH, HIGH(Number4\*2)

RJMP SPZend

N5: CPI temp1, 5

BRNE N6

LDI ZL, LOW(Number5\*2)

LDI ZH, HIGH(Number5\*2)

RJMP SPZend

N6: CPI temp1, 6

BRNE N7

LDI ZL, LOW(Number6\*2)

LDI ZH, HIGH(Number6\*2)

RJMP SPZend

N7: CPI temp1, 7

BRNE N8

LDI ZL, LOW(Number7\*2)

LDI ZH, HIGH(Number7\*2)

RJMP SPZend

N8: CPI temp1, 8

BRNE N9

LDI ZL, LOW(Number8\*2)

LDI ZH, HIGH(Number8\*2)

RJMP SPZend

N9: CPI temp1, 9

BRNE NX

LDI ZL, LOW(Number9\*2)

LDI ZH, HIGH(Number9\*2)

RJMP SPZend

NX: CPI temp1, 10

BRNE SPZend

LDI ZL, LOW(NumberX\*2)

LDI ZL, LOW(NumberX\*2)

SPZend:

LPM temp1, Z+ ;To shift the first place (Number Tab)

LDI index5, 5 ;reinit "watchdog" of Z pointer

RET

MatrixKeypad:

SetColumn1:

LDI temp1, Column1 ;Choose column 1

OUT portKey, temp1

IN temp1, pinKey ;Read port

TestStar:

CPI temp1, Star

BRNE test7

LDI Key, 10

SET ;T=1

RJMP MKend

Test7:

CPI temp1, Seven

BRNE test4

LDI Key, 7

RJMP MKend

Test4:

CPI temp1, Four

BRNE test1

LDI Key, 4

RJMP MKend

Test1:

CPI temp1, One

BRNE SetColumn2

LDI Key, 1

RJMP MKend

SetColumn2:

LDI temp1, Column2 ;Choose column 2

OUT portKey, temp1

IN temp1, pinKey ;Read port

Test0:

CPI temp1, Zero

BRNE test8

LDI Key, 0

RJMP MKend

Test8:

CPI temp1, Eight

BRNE test4

LDI Key, 8

RJMP MKend

Test5:

CPI temp1, Five

BRNE test1

LDI Key, 5

RJMP MKend

Test2:

CPI temp1, Two

BRNE SetColumn3

LDI Key, 2

RJMP MKend

SetColumn3:

LDI temp1, Column3 ;Choose column 3

OUT portKey, temp1

IN temp1, pinKey ;Read port

TestHash:

CPI temp1, Hash

BRNE test9

LDI Key, 11

CLT ;T=0

RJMP MKend

Test9:

CPI temp1, Nine

BRNE test6

LDI Key, 9

RJMP MKend

Test6:

CPI temp1, Six

BRNE test3

LDI Key, 6

RJMP MKend

Test3:

CPI temp1, Three

BRNE NoKey

LDI Key, 3

RJMP MKend

NoKey:

LDI Key, MAX

MKend:

RET

DisplacementSavedNumbers:

LDI YL, LOW(SavedNumbersEnd)

LDI YH, HIGH(SavedNumbersEnd)

LDI XL, LOW(SavedNumbersEnd)

LDI XH, HIGH(SavedNumbersEnd)

LDI index1, (SavedNumbersEnd-SavedNumbers-1)

LD temp1, -X

DisplacementSavedNumbers\_Loop:

LD temp1, -X

ST -Y, temp1

DEC index1

BRNE DisplacementSavedNumbers\_Loop

RET

DisplayX:

LDI XL, LOW(SavedNumbers) ;adress with stored "X"

LDI XH, HIGH(SavedNumbers)

LDI temp1, 10 ;restore "X"

ST X, temp1

Delay1ms:

LDI index1, 0

LDI index2, 16

LDI index3, 1

RJMP Cycle

Delay1s:

LDI index1, 0

LDI index2, 0

LDI index3, 16

Cycle:

DEC index1

BRNE Cycle

DEC index2

BRNE Cycle

DEC index3

BRNE Cycle

RET

;Number table

;In every line must be even number, so first is 0x00, it´s skipped when the Z pointer is setting.

NumberX:.DB 0b00000000, 0b01011101, 0b01101011, 0b01110111, 0b01101011, 0b01111111

Number0:.DB 0b00000000, 0b01111111, 0b01000001, 0b00111110, 0b01000001, 0b01111111

Number1:.DB 0b00000000, 0b01111111, 0b00111101, 0b00000000, 0b00111111, 0b01111111

Number2:.DB 0b00000000, 0b01111111, 0b00011101, 0b00100110, 0b00111001, 0b01111111

Number3:.DB 0b00000000, 0b01111111, 0b01011101, 0b00110110, 0b01001001, 0b01111111

Number4:.DB 0b00000000, 0b01111111, 0b01000111, 0b00011001, 0b01011110, 0b01111111

Number5:.DB 0b00000000, 0b01111111, 0b01010000, 0b00111010, 0b01000110, 0b01111111

Number6:.DB 0b00000000, 0b01111111, 0b01000001, 0b00110110, 0b01001110, 0b01111111

Number7:.DB 0b00000000, 0b01111111, 0b00001110, 0b01110010, 0b01111100, 0b01111111

Number8:.DB 0b00000000, 0b01111111, 0b01001001, 0b00110110, 0b01001001, 0b01111111

Number9:.DB 0b00000000, 0b01111111, 0b01111001, 0b01110110, 0b01111110, 0b01111111