Dílenská praxe

A4	7. Alfanumerický displej		
Tenk Jakub		1/12	Známka:
16. 2. 2022	Datum odevzdání:	16. 3. 2022	Odevzdáno:

Zadání:

Zpracujte program v programovacím jazyce C ovládající alfanumerický displej tak, aby obsahoval nejméně tyto funkce:

1) volbu druhu displeje (7segmentový/14segmentový)
2) zobrazení vhodně zvolené množiny znaků pro každý typ displeje
3) vhodně zvolená datová a programová struktura

Propojení PC a alfanumerického displeje:

0x300 P1 (OUT)		**0x301 P2 (OUT)**	
Jméno pinu	**Číslo bitu**	**Jméno pinu**	**Číslo bitu**
DATA	0	CLK	0
	1		1
	2		2
	3		3
	4		4
	5		5
	6		6
	7		7

Vývojový diagram:

Příloha 1 – vývojový diagram

Výpis programu:

Příloha 2 – výpis programu

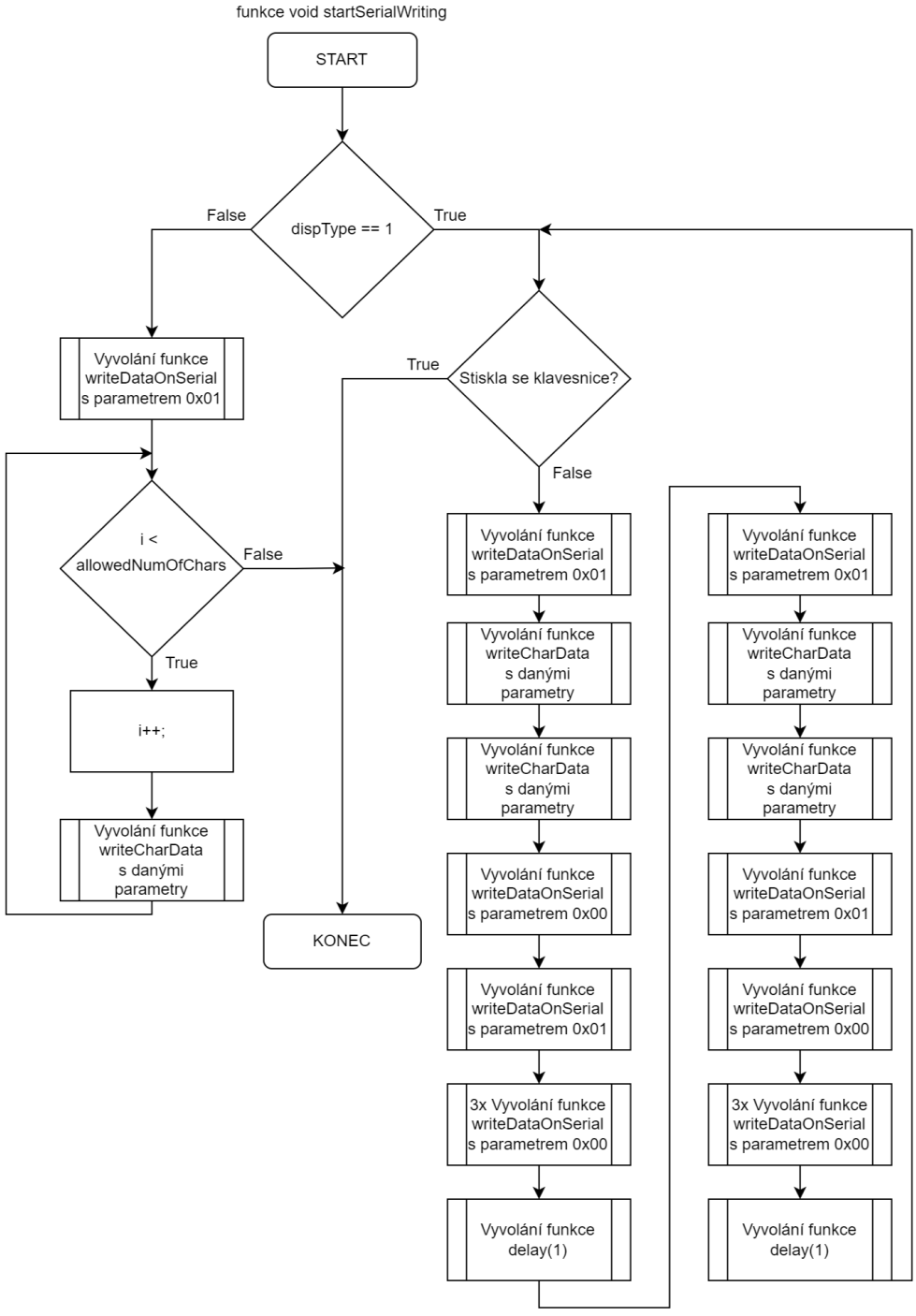
Závěr:

Program je sestaven dle zadání a měl by být funkční. Se zpracováním této úlohy nebyl žádný problém.

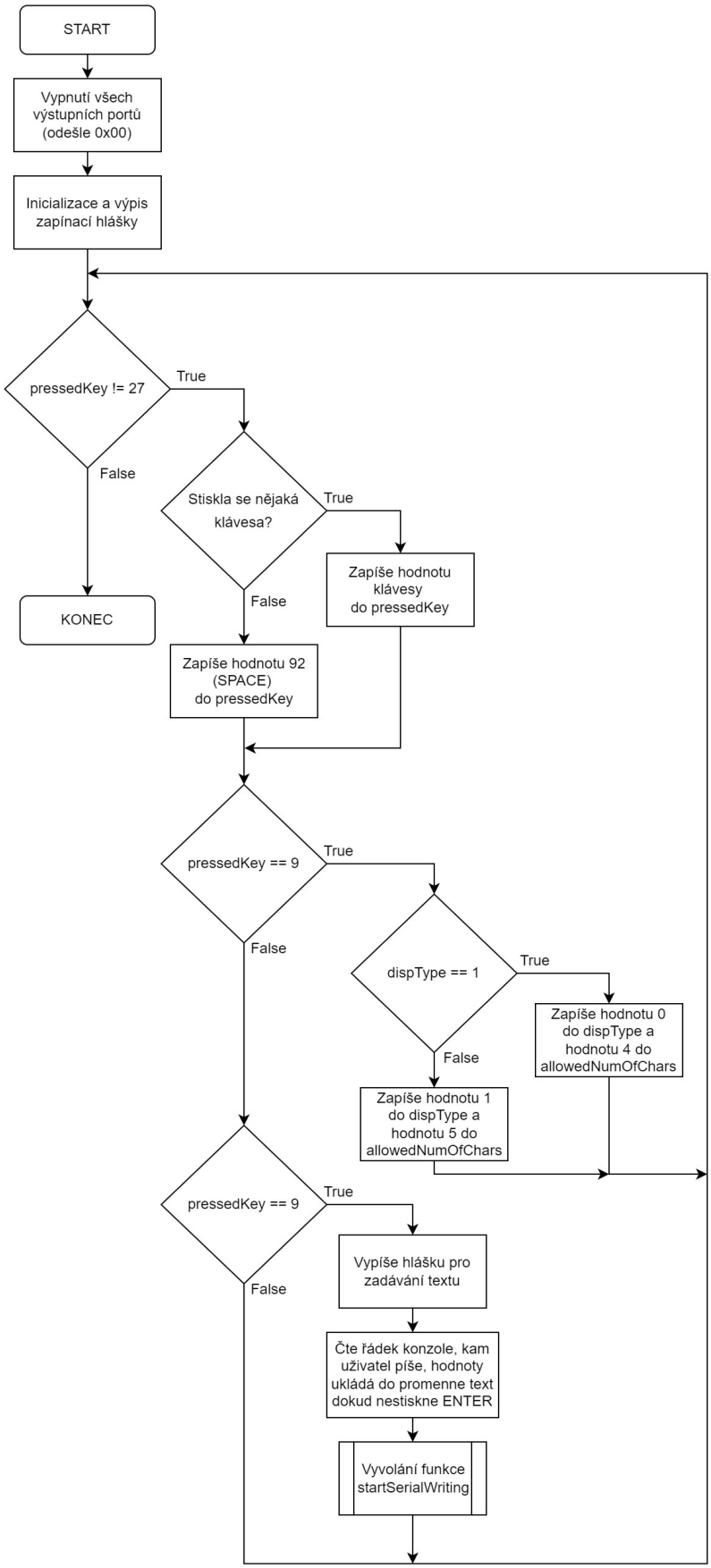
Přílohy:

- Příloha 1 – 3 strany
- Příloha 2 – 7 stran

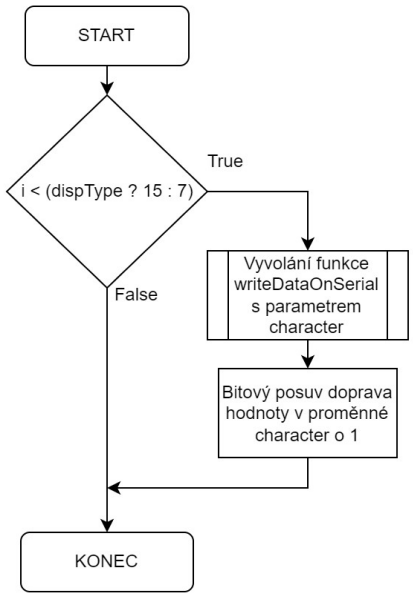
Příloha 1:


funkce void startSerialWriting
START
dispType == 1
False
True
Vyvolání funkce writeDataOnSerial s parametrem 0x01
i < allowedNumOfChars
False
True
i++;
Vyvolání funkce writeCharData s danými parametry
KONEC
Stiskla se klavesnice?
True
False
Vyvolání funkce writeDataOnSerial s parametrem 0x01
Vyvolání funkce writeCharData s danými parametry
Vyvolání funkce writeCharData s danými parametry
Vyvolání funkce writeDataOnSerial s parametrem 0x00
Vyvolání funkce writeDataOnSerial s parametrem 0x01
3x Vyvolání funkce writeDataOnSerial s parametrem 0x00
Vyvolání funkce delay(1)
Vyvolání funkce writeDataOnSerial s parametrem 0x01
Vyvolání funkce writeCharData s danými parametry
Vyvolání funkce writeCharData s danými parametry
Vyvolání funkce writeDataOnSerial s parametrem 0x01
Vyvolání funkce writeDataOnSerial s parametrem 0x00
3x Vyvolání funkce writeDataOnSerial s parametrem 0x00
Vyvolání funkce delay(1)

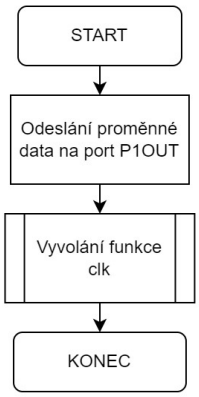
Hlavní program (funkce int main)



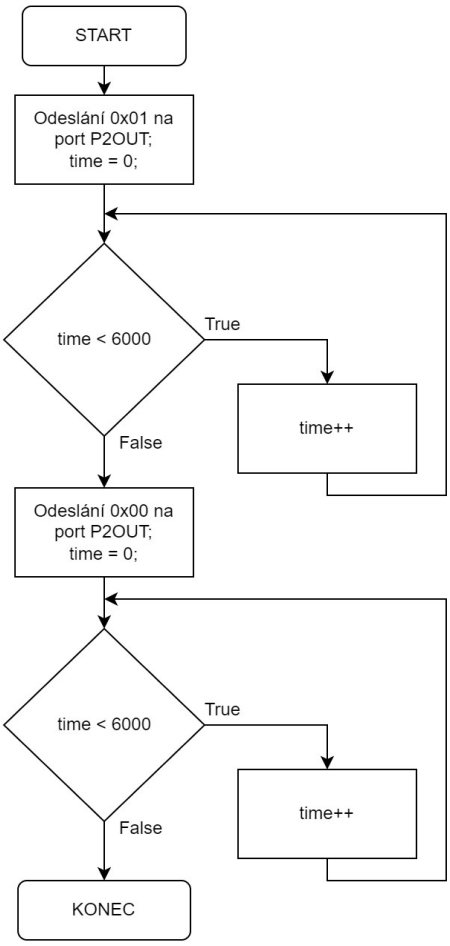
funkce void startSerialWriting



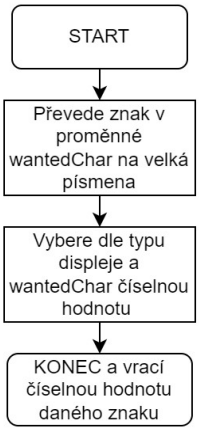
funkce void writeDataOnSerial



funkce void clk



funkce int charToHex



Příloha 2:

```
/*
   Dokumentace zapojeni:
        P1 (0x300):  0 bit = DATA
        P2 (0x301):  0 bit = CLK
*/

#include <stdio.h>
#include <conio.h>
#include <time.h>
#include <ctype.h>
#include <dos.h>

#define P1OUT 0x300
#define P2OUT 0x301

char text[1024]; // Pole pro zadavany text (128 bajtu)

// Prelozi pozadovany znak na hex. cislo, pro zadany displej (7/14 segmentovy)
int charToHex(int dispType, char wantedChar)
{
    wantedChar = toupper(wantedChar); // Prevede znak na velka pismena (male nezobrazujeme)

    if (dispType) {
        switch (wantedChar)
        {
            case 'A':
                return(0x1137);
            case 'B':
                return(0x054F);
            case 'C':
                return(0x0039);
            case 'D':
                return(0x044F);
            case 'E':
                return(0x1139);
            case 'F':
                return(0x1131);
            case 'G':
                return(0x013D);
            case 'H':
                return(0x1136);
            case 'I':
                return(0x0449);
            case 'J':
                return(0x001E);
```

```
case 'K':
    return(0x12B0);
case 'L':
    return(0x0038);
case 'M':
    return(0x20B6);
case 'N':
    return(0x2236);
case 'O':
    return(0x003F);
case 'P':
    return(0x1133);
case 'Q':
    return(0x023F);
case 'R':
    return(0x1333);
case 'S':
    return(0x112D);
case 'T':
    return(0x0441);
case 'U':
    return(0x003E);
case 'V':
    return(0x2206);
case 'W':
    return(0x0A36);
case 'X':
    return(0x2A80);
case 'Y':
    return(0x2480);
case 'Z':
    return(0x0889);
case '0':
    return(0x003F);
case '1':
    return(0x0006);
case '2':
    return(0x111B);
case '3':
    return(0x110F);
case '4':
    return(0x1126);
case '5':
    return(0x112D);
case '6':
    return(0x113C);
```

```
            case '7':
                return(0x1981);
            case '8':
                return(0x113F);
            case '9':
                return(0x1127);
            default:
                return(0x0000);
        }
    }
    else
    {
        switch (wantedChar)
        {
            case '0':
                return(0x3F);
            case '1':
                return(0x06);
            case '2':
                return(0x5B);
            case '3':
                return(0x4F);
            case '4':
                return(0x66);
            case '5':
                return(0x6D);
            case '6':
                return(0x7D);
            case '7':
                return(0x07);
            case '8':
                return(0x7F);
            case '9':
                return(0x6F);
            default:
                return(0x00);
        }
    }
}

// Hodiny pro seriovy displej
void clk()
{
    int time = 0;

    outportb(P2OUT, 0x01);
```

```
    // hack-fix (misto pouziti klasicke funkce delay, zpozdime program pomoci cyklu for)
    for (time = 0; time < 6000; time++)
    {}

    outportb(P2OUT, 0x00);

    for (time = 0; time < 6000; time++)
    {}
}

// Slouzi pro odesilani bitu na seriovy displej
void writeDataOnSerial(int data)
{
    outportb(P1OUT, data);
    clk();
}

// Odesilani samotneho znaku na seriovku
void writeCharData(int character, int dispType)
{
    for (int i = 0; i < (dispType ? 15 : 7); i++)
    {
        writeDataOnSerial(character);   // Odeslani
        character >>= 1;    // Bitovy posun (posunuti na dalsi bit znaku)

    }
}

// Odesilani dat pres seriovou komunikaci
void startSerialWriting(int allowedNumOfChars, int dispType)
{
    if (dispType)
    {
        printf("\n\rPro ukonceni odesilani textu na displej zmacknete libovolnou klavesu
krome ESC\n\r");

        while (!kbhit())
        {
            int i = 0;

            // Start bit
            writeDataOnSerial(0x01);

            // Odesle 1. a 3. znak (A1)
            writeCharData(charToHex(dispType, text[0]), dispType);
```

```
            writeCharData(charToHex(dispType, text[2]), dispType);

            // Aktivuje levy tranzistor
            writeDataOnSerial(0x00);
            writeDataOnSerial(0x01);

            // Vypln zbylich bitu (bit 33-35)
            for (i = 0; i < 3; i++)
            {
                writeDataOnSerial(0x00);
            }

            delay(1);

            // Start bit
            writeDataOnSerial(0x01);

            // Odesle 2. a 4. znak (A2)
            writeCharData(charToHex(dispType, text[1]), dispType);
            writeCharData(charToHex(dispType, text[3]), dispType);

            // Aktivuje pravy tranzistor
            writeDataOnSerial(0x01);
            writeDataOnSerial(0x00);

            // Vypln zbylich bitu (bit 33-35)
            for (i = 0; i < 3; i++)
            {
                writeDataOnSerial(0x00);
            }

            delay(1);
        }
    }
    else
    {
        // Start bit
        writeDataOnSerial(1);

        // Postupne odesle znaky
        for (int i = 0; i < allowedNumOfChars; i++)
        {
            writeCharData(charToHex(dispType, text[i]), dispType);
        }
    }
}
```

```
// Hlavni funkce programu
int main()
{
    // Vypnuti vseho na vystupnich portech
    outport(P1OUT, 0x00);
    outport(P2OUT, 0x00);

    char pressedKey;
    int allowedNumOfChars = 5; // maximalni pocet znaku displeje
    int dispType = 0; // 0 = 7segmentovy, 1 = 14segmentovy

    // Vymazani obrazovky + vypis zakladnich informaci k ovladani
    printf("\x1B[2J\x1B[H");
    printf("\n\r[############################################################]");
    printf("\n\r   Zapinani programu AlfaNum displej, vytvoril Jakub Tenk    ");
    printf("\n\r[------------------------------------------------------------]");
    printf("\n\r Zakladni ovladani: ");
    printf("\n\r  Stisk klavesy TAB  ->  zmena typu displeje 7/14 segmentovy ");
    printf("\n\r  Stisk klavesy `    ->  zadavani textu / odeslani textu     ");
    printf("\n\r  Stisk klavesy ESC  ->  vypnuti programu                    ");
    printf("\n\r[############################################################]\n\r");

    while (pressedKey != 27)
    {
     // Kdyz se smazkne klavesa, tak ji to precte, jinak dosadi znak SPACE
     pressedKey = kbhit() ? getch() : 92;

     switch (pressedKey)
     {
      case 9: // Klavesa TAB
       dispType = dispType ? 0 : 1;
       allowedNumOfChars = dispType ? 4 : 5;

       printf("\n\rZmenen typ displeje na: %s", dispType ? "14segmentovy" : "7segmentovy");
       printf("\n\rMaximalni delka textu zmenen na: %d\n\r", allowedNumOfChars);
       break;
      case 96: // Klavesa `
       printf("\n\rZadejte text s maximalni delkou %d znaku:\n\r", allowedNumOfChars);
       scanf("%s", &text);

       printf("\n\rOdesilani textu na displej...\n\r");
       startSerialWriting(allowedNumOfChars, dispType);
       printf("\n\rOdeslani textu probehlo uspesne!\n\r");
       break;
     }
    }
```

```
    printf("\n\r[#####################################]");
    printf("\n\r   Vypinani programu AlfaNum displej    ");
    printf("\n\r[#####################################]");

    delay(1000);

    return 0;
}
```