



Díleňská praxe

A4	07 Sériová komunikace		
Kahovec Petr		1/11	Známka:
15.2.2024	Datum odevzdání:	14.3.2024	Odevzdáno:

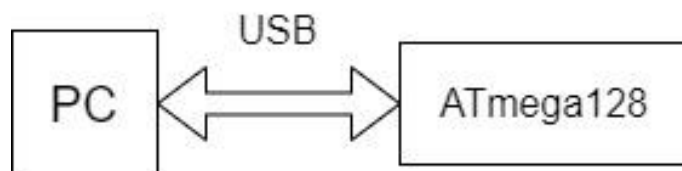


Zadání:

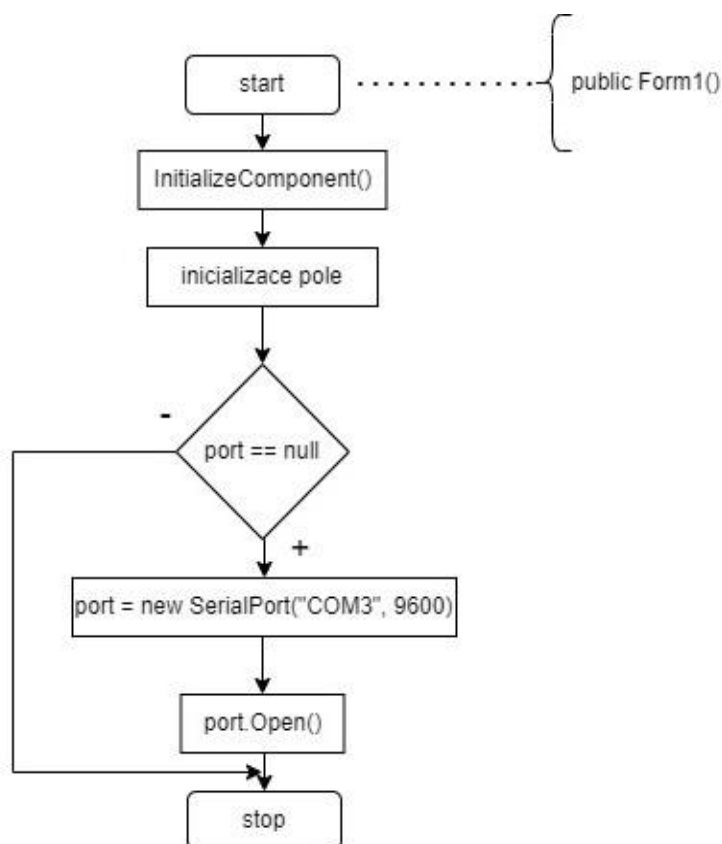
Zpracujte program v programovacím jazyce C a C# ovládající sériovou komunikaci mezi jedním přípravkem MB-ATmega128 a PC tak, aby obsahoval nejméně tyto funkce:

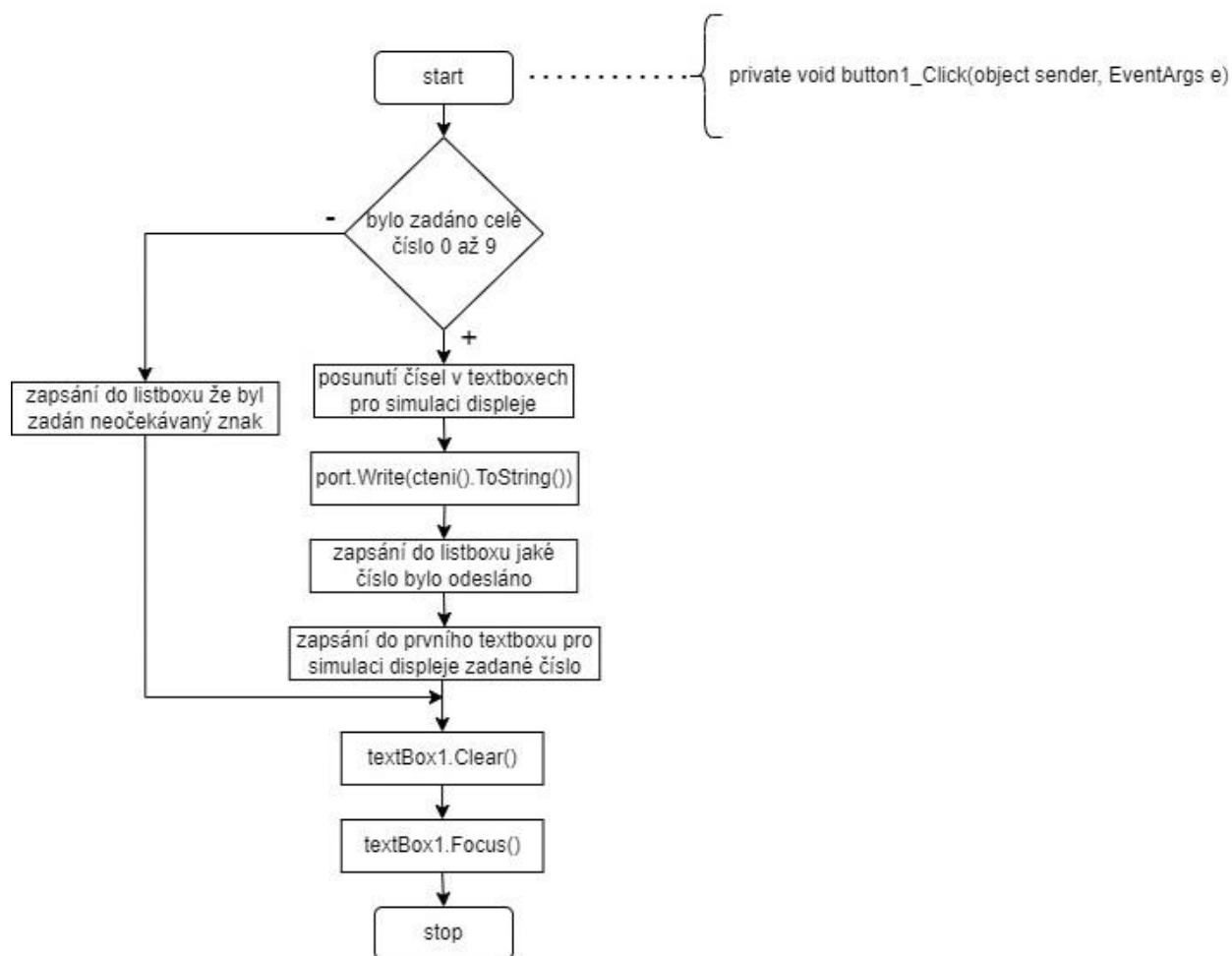
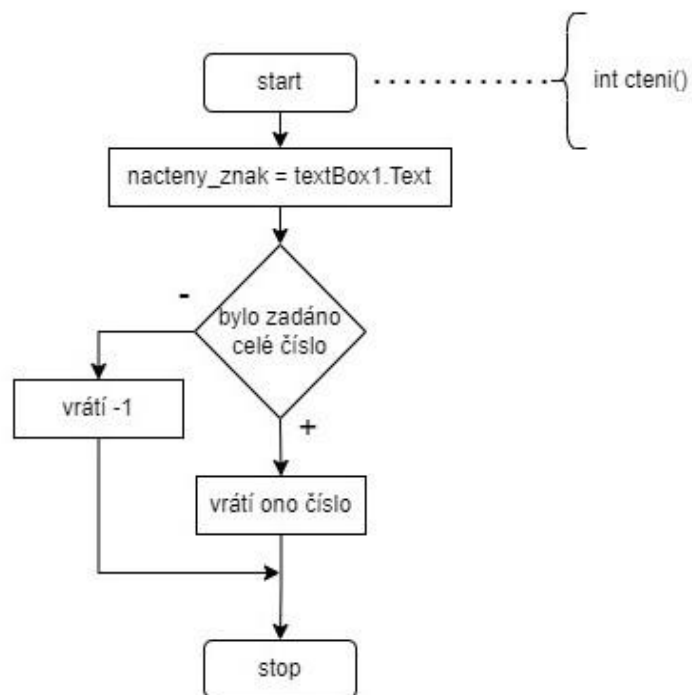
- 1) stisknuté tlačítko klávesnice počítače se sériovou linkou přenese do přípravku MB-ATmega128, kde se bude postupně zobrazovat na modulu multiplexovaných 8 LED displejů.
- 2) zvolte vhodný komunikační protokol
- 3) komunikační program v přípravku MB-ATmega128 s modulem multiplexovaných 8LED displejů měl mít (volitelně) funkci „autonegotiation“
- 4) využití všech vhodných HW možností přípravku MB-ATmega128.
- 5) sledování chybových stavů

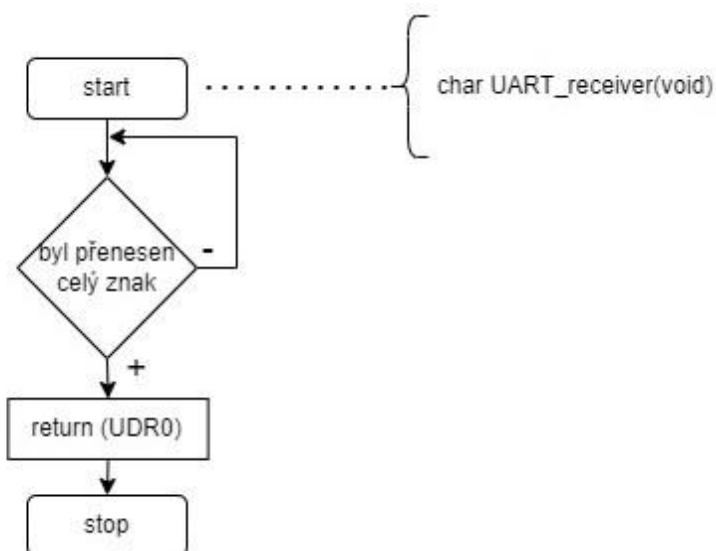
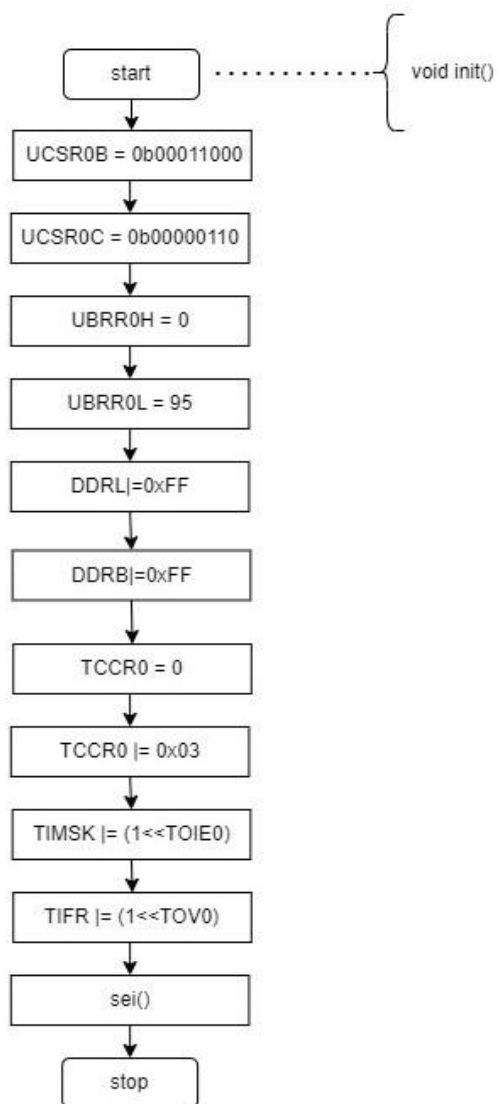
Schéma zapojení:

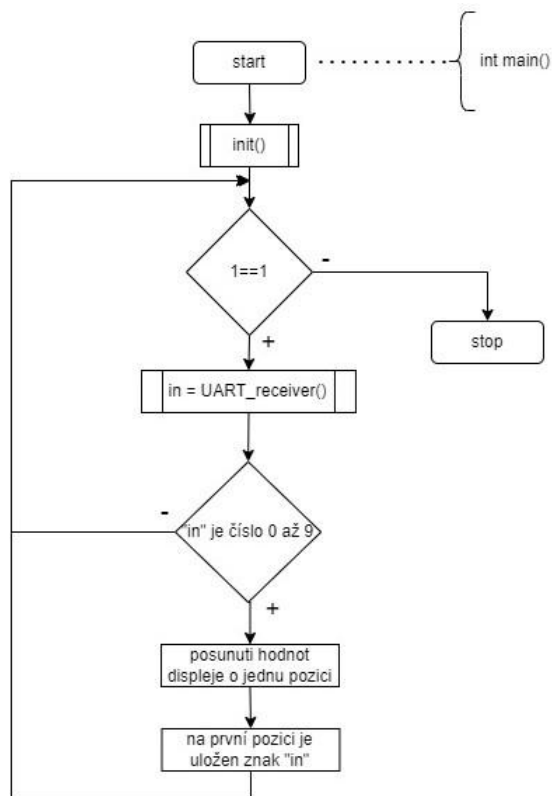
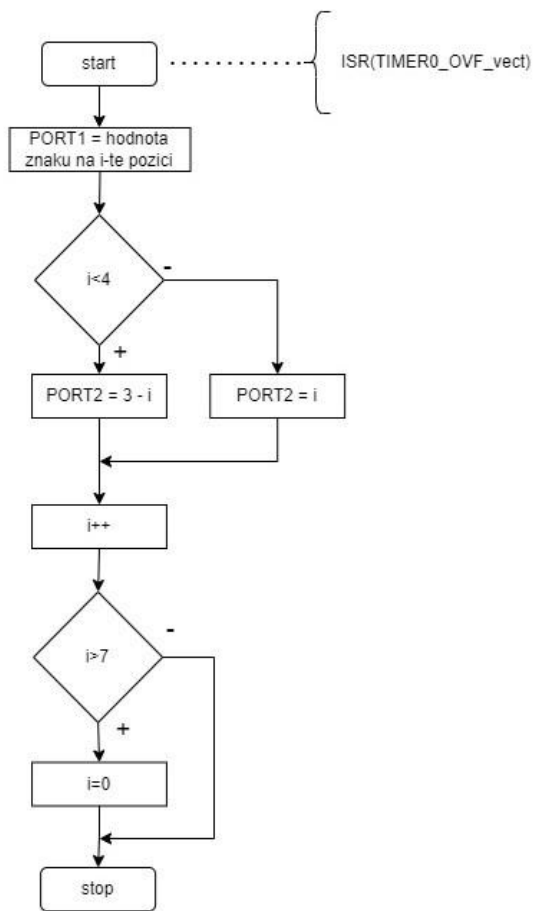


Vývojový diagram:











Postup (principy řešení):

Konfigurace USARTu byla provedena dle datasheetu:

1. V registru UCSR0B bylo nastaveno přepnutí RX a TX ze standardního provozu na USART
2. V registru UCSR0C bylo nastaveno že datové slovo bude mít šířku 8 bitů
3. Pro námi využívanou hodnotu frekvence CPU a baudové rychlosti byla z tabulky odečtena potřebná hodnota pro nastavení registrů UBRR0H a UBRR0L
4. Pro čtení je využit registr UCSR0A kde bit RXC0 indikuje, zda byl přijmut kompletní znak, který nahrajeme z bitu UDR0
5. Pro zobrazování znaků na displej využijeme přerušení, a to přetečení TIMERu0
6. Pro odesílání znaků byla použita třída SerialPort pomocí které lze vybrat port, nastavit baudovou rychlost, otevřít komunikaci a odesílat data

Výpis programu:

-Viz přílohy

Komentář k programu:

-Viz výpis programu

Závěr:

Byla ověřena vyzkoušena pouze funkčnost programu pro odesílání v jazyce C# a funkce pro čtení odeslané hodnoty v jazyce C, oba tyto programy byly funkční, jediné, co nebylo otestováno bylo zobrazování přečtených znaků na maticový displej.

Přílohy:

-Výpis programu (5 stránek)



C# kód:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO.Ports;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace seriova_komunikace
{
    public partial class Form1 : Form
    {
        public TextBox[] displej = new TextBox[8];
        SerialPort port;
        string adresa_portu = "COM3";
        int baud = 9600;
        public Form1()
        {
            InitializeComponent();
            //načtení textboxů do pole
            displej[0] = textBox2;
            displej[1] = textBox3;
            displej[2] = textBox4;
            displej[3] = textBox5;
            displej[4] = textBox6;
            displej[5] = textBox7;
            displej[6] = textBox8;
            displej[7] = textBox9;
            if (port == null)
            {
                //vybrání výstupního portu a baudové rychlosti
                port = new SerialPort(adresa_portu, baud);
                //otevření portu
                port.Open();
            }
        }
        int cteni()
        {
            int cislo;
            string nacteny_znak;
            //načtení řetězce z textboxu pro zadávání čísla do proměnné
            nacteny_znak = textBox1.Text;
            //pokud je načtený řetězec číslo pak tento řetězec převede na
            //celé číslo
            if (int.TryParse(nacteny_znak, out cislo))
            //vrátí toto číslo a ukončí funkci
            { return cislo; }
            //pokud zadaný řetězec není číslo pak vrátí -1 a ukončí funkci
            else return -1;
        }
    }
}
```



```
private void button1_Click(object sender, EventArgs e)
{
    //pokud se jedná o celé číslo 0 až 9
    if ((cteni() <= 9) && (cteni() >= 0))
    {
        //posun čísel v textbozech pro zobratování stavu displeje
        for (int i = 7; i > 0; i--)
        {
            displej[i].Text = displej[i-1].Text;
        }
        //odeslání hodnoty na výstup
        port.Write(cteni().ToString());
        //zapsání do listboxu co bylo právě provedeno
        listBox1.Items.Add("Bylo odesláno číslo " + cteni().ToString());
        //zapsní čísla do prvního textboxu
        textBox2.Text = Convert.ToString(cteni());
    }
    //jinak se jedná o neočekávaný znak
    else
    {
        //zapsání do listboxu co bylo právě provedeno
        listBox1.Items.Add
("Byl zadán neočekávaný znak, zadejte prosím celé číslo 0 až 9");
    }
    //vymazání textboxu pro zadávání čísla
    textBox1.Clear();
    textBox1.Focus();
}
}
```




C kód:

```
#define F_CPU 14745600UL
#define BAUD 9600
#include<avr/interrupt.h>//obsluha přerušení
#include<avr/io.h>//registry řadiče
#include<avr/pgmspace.h>//data v programové paměti
#include<util/delay.h>//odměřování času
#include<stdio.h>
#define preddelicka 0x03
#define vystup 0xFF
#define ascii_9 57//ascii hodnota pro znak 9
#define ascii_0 48//ascii hodnota pro znak 0
#define PORT1 PORTF//PORT pro odesílání dat na displej
#define PORT2 PORTA//PORT pro výběr segmentu displeje

void init()
{
    //povolení USART přijímače, povolení USART vysílače
    UCSRB = 0b00011000;
    //nastavení přenášeného znaku na 8 bitů
    UCSRC = 0b00000110;
    //nastavení pro baudrate 9600, frekvenci CPU 14,7456 MHz
    UBRRH = 0;
    UBRRL = 95;
    //nastavení portu jako výstupní
    DDRA|=vystup;
    DDRF|=vystup;
    //vynulování časovače 0
    TCCR0 = 0;
    //nastavení předděličky časovače 0
    TCCR0 |= preddelicka;
    //nastavení režimu přerušení
    TIMSK |= (1<<TOIE0);
    TIFR |= (1<<TOV0);
    //povolení globálního přerušení
    sei();
}

//zobrazená čísla
volatile int zobrazeni[8]= {0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};
volatile int i=0;
//definice znaků a čísel
volatile uint8_t definice[10]
={0xFC,0x60,0xDA,0xF2,0x66,0xB6,0xBE,0xE0,0xFE};
```



```
char UART_receiver(void)
{
    //čekání dokud nebyl přenesen celý znak
    while ((UCSR0A & (1 << RXC0))==0);
    //vrátí přenesený znak
    return (UDR0);
}

ISR(TIMER0_OVF_vect) //přerušeni reagující na přetečení timeru 0
{
    //na PORT pošle hodnotu pro vykreslení daného znaku
    PORT1 = definice[zobrazeni[i]];
    if (i<4)
    {
        //u vybraného displeje jsou první 4 segmenty přehozeny
        PORT2 = 3 - i;
    }
    else
    {
        //poslední čtyři jsou již vybírány normálně
        PORT2 = i;
    }
    i++;
    if (i>7)
    {
        //pokud dojedeme na konec displeje vracíme se na začátek
        i = 0;
    }
}

int main()
{
    init();
    int j, in;
    while(1)
    {
        //uložení přečtené acii hodnoty znaku
        in = UART_receiver();
        //pokud se jedná o číslo mezi 0 až 9
```



```
if (ascii_9>=in>=ascii_0)
{
    for (j=7;j>0;j--)
    {
        //posuneme hodnoty na jednotlivých pozicích o jednu doprava
        zobrazeni[j]=zobrazeni[j-1];
    }
    //na první pozici uložíme zadané číslo
    zobrazeni[0] = in - ascii_0;
}
}
```