



Díleňská praxe

A4	Model křížovatky		
		1/7	Známka:
	Datum odevzdání:		Odevzdáno:



Zadání:

Zpracujte program v programovacím jazyce C# ovládající model křižovatky tak, aby obsahoval nejméně tyto funkce:

- 1) funkce řízení světel křižovatky respektuje pravidla silničního provozu,
- 2) pomocí tlačítek modelu přepínáte denní/noční režim
- 3) pomocí tlačítek modelu přepínáte plná/zjednodušená křižovatka,
- 4) na monitoru počítače zobrazujete aktuální stav světel křižovatky, případně režimu činnosti křižovatky.

Schema zapojení (situační schema):

Out:

chodci hlavní		chodci vedlejší			hlavní ↑→		
8	7	6	5	4	3	2	1
Z	Č	Z	Č	X	Z	O	Č

adresa karty

karta 0 (SK5 - ZAP, SK6 - ZAP)

	hlavní ←			vedlejší			
8	7	6	5	4	3	2	1
X	Z	O	Č	Z →	Z	O	Č

adresa karty

karta 1 (SK5 - VYP, SK6 - ZAP)

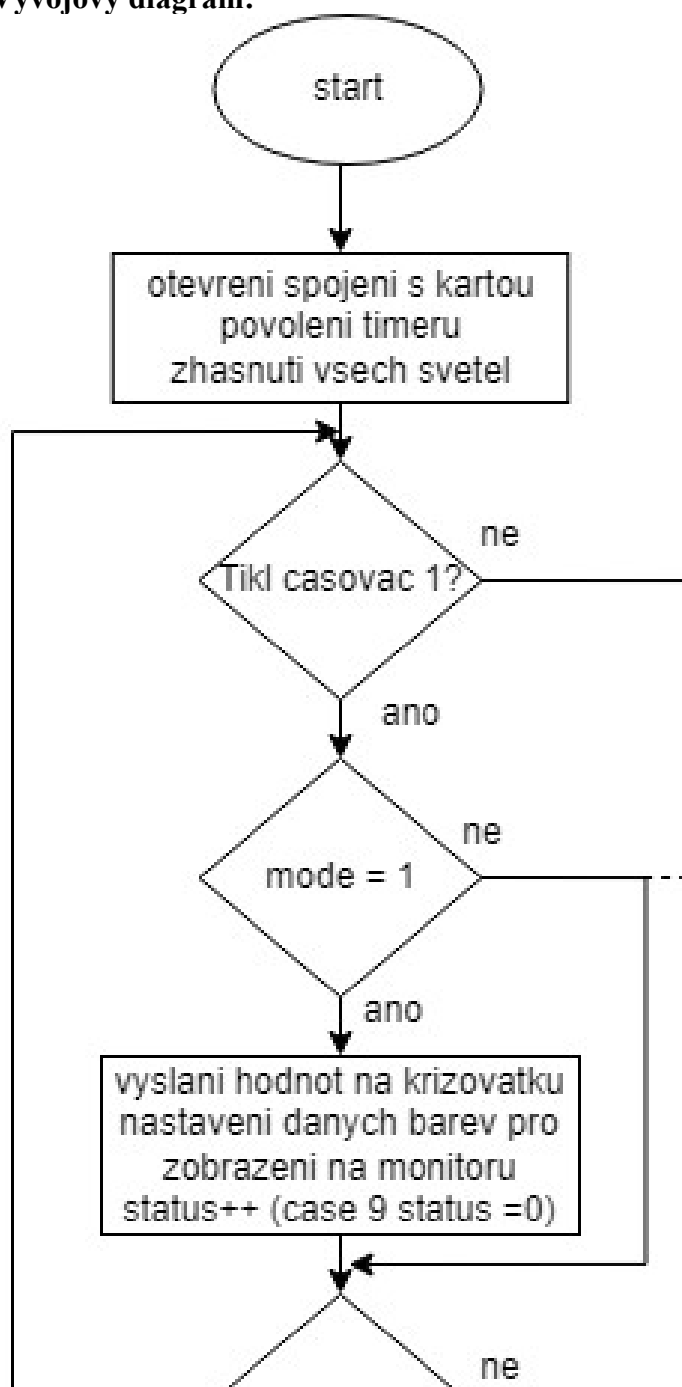
In:

Inp1	1 (A)
inp2	2 (B)

karta 0



Vývojový diagram:



Postup (principy řešení):

Vložení knihovny K8055.dll (bin\debug). Ve formu pomocí tlačítek simulace semaforu na monitoru.

Výpis programu:

V příloze.

Závěr:

Program by měl splňovat zadání, až na zjednodušenou křižovatku.



Přílohy:

Výpis programu:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Runtime.InteropServices;
namespace krizovatka
{
    public partial class Form1 : Form
    {
        int status = 0;
        int statusnoc = 0;
        int mode = 1;
        //funkce pro vyslani hodnot pro krizovatku
        void outport(int k0, int k1)
        {
            K8055D.SetCurrentDevice(0);
            K8055D.WriteAllDigital(k0);
            K8055D.SetCurrentDevice(1);
            K8055D.WriteAllDigital(k1);
        }
        public Form1()
        {
            InitializeComponent();
        }

        private class K8055D
        {
            [DllImport("K8055D.dll")]
            public static extern int OpenDevice(int CardAddress);
            [DllImport("K8055D.dll")]
            public static extern void WriteAllDigital(int Data);
            [DllImport("K8055D.dll")]
            public static extern bool ReadDigitalChannel(int Channel);
            [DllImport("K8055D.dll")]
            public static extern int SetCurrentDevice(int lngCardAddress);
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            K8055D.OpenDevice(0);
            K8055D.OpenDevice(1);
            timer1.Enabled = true; //zmena semaforu
            timer2.Enabled = true; //zkoumani tlacitek
            outport(0b1111_11_11, 0b1111_11_11); //vse zhasnout
        }

        private void Timer1_Tick(object sender, EventArgs e)
        {
            if (mode == 1) //normalni rezim
```



```
{
    switch (status)
    {
        case 0:
            outport(0b10101110, 0b10110110);
            hlrovneC.BackColor = Color.Red;
            hldolevaZ.BackColor = Color.Green;
            hldolevaC.BackColor = Color.Transparent;
            hldolevaO.BackColor = Color.Transparent;
            vedlejsipravaZ.BackColor = Color.Green;
            vedlejsiC.BackColor = Color.Red;
            hlchodecC.BackColor = Color.Red;
            vedlejsichodecC.BackColor = Color.Red;
            status++;
            break;
        case 1:
            outport(010011100, 0b11011110);
            hlrovneO.BackColor = Color.Orange;
            hldolevaO.BackColor = Color.Orange;
            hldolevaZ.BackColor = Color.Transparent;
            vedlejsipravaZ.BackColor = Color.Transparent;
            vedlejsichodecZ.BackColor = Color.Green;
            vedlejsichodecC.BackColor = Color.Transparent;
            hlchodecC.BackColor = Color.Red;
            status++;
            break;
        case 2:
            outport(0b10011011, 0b11101110);
            hlrovneZ.BackColor = Color.Green;
            hldolevaC.BackColor = Color.Red;
            hldolevaO.BackColor = Color.Transparent;
            hlrovneC.BackColor = Color.Transparent;
            hlrovneO.BackColor = Color.Transparent;
            status++;
            break;
        case 3:
            outport(0b10101101, 0b11001110);
            hlrovneO.BackColor = Color.Orange;
            hlrovneZ.BackColor = Color.Transparent;
            hldolevaO.BackColor = Color.Orange;
            vedlejsichodecZ.BackColor = Color.Transparent;
            vedlejsichodecC.BackColor = Color.Red;
            status++;
            break;
        case 4:
            outport(0b10101110, 0b10110110);
            hlrovneC.BackColor = Color.Red;
            hlrovneO.BackColor = Color.Transparent;
            hldolevaZ.BackColor = Color.Green;
            hldolevaO.BackColor = Color.Transparent;
            hldolevaC.BackColor = Color.Transparent;
            vedlejsipravaZ.BackColor = Color.Green;
            status++;
            break;
        case 5:
            outport(0b10101110, 0b11011110);
```



```
        hldolevaZ.BackColor = Color.Transparent;
        hldolevaO.BackColor = Color.Orange;
        vedlejsipravaZ.BackColor = Color.Transparent;
        status++;
        break;
    case 6:
        output(0b01101110, 0b11101100);
        hldolevaO.BackColor = Color.Transparent;
        hldolevaC.BackColor = Color.Red;
        vedlejsiO.BackColor = Color.Orange;
        hlchodecZ.BackColor = Color.Green;
        hlchodecC.BackColor = Color.Transparent;
        status++;
        break;
    case 7:
        output(0b01101110, 0b11101011);
        vedlejsiO.BackColor = Color.Transparent;
        vedlejsiC.BackColor = Color.Transparent;
        vedlejsiZ.BackColor = Color.Green;
        status++;
        break;
    case 8:
        output(0b10101110, 0b11101101);
        vedlejsiO.BackColor = Color.Orange;
        vedlejsiZ.BackColor = Color.Transparent;
        hlchodecZ.BackColor = Color.Transparent;
        hlchodecC.BackColor = Color.Red;
        status++;
        break;
    case 9:
        output(0b10101110, 0b11001110);
        vedlejsiC.BackColor = Color.Red;
        hldolevaO.BackColor = Color.Orange;
        vedlejsiO.BackColor = Color.Transparent;
        status = 0;
        break;
    }
}
if (mode == 2) //nocni rezim
{
    switch (statusnoc)
    {
        case 0:
            output(0b11111101, 0b11011101);
            statusnoc++;
            break;
        case 1:
            output(0b11111111, 0b11111111);
            statusnoc = 0;
            break;
    }
}
}
private void Timer2_Tick(object sender, EventArgs e) //tlacitka
{
    if (K8055D.ReadDigitalChannel(1))
```



```
{ mode = 1; }  
if (K8055D.ReadDigitalChannel(2))  
{ mode = 2; }  
}  
}
```