Dílenská praxe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A4** | 3. Model křižovatky | | | |
| Macháček Daniel | |  | 1/11 | Známka: |
| 15.11.2016 | | Datum odevzdání: | 7.12.2016 | 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 přepínejte režim denní/noční (volitelně plná/zjednodušená křižovatka)
3. Na monitoru počítače zobrazujte aktuální stav křižovatky, případně režimu činnosti křižovatky

**Zapojení křižovatky:**

** **

** **

Závěr:

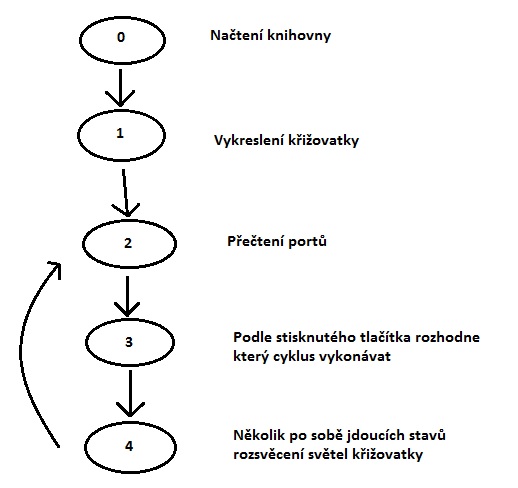
Přestože kompilátor nevyhazoval žádnou chybu, samotnou vizualizaci se mi nepovedlo rozchodit. Rozsvěcení křižovatky je dle mého názoru funkční, ale kvůli nedostatku prostředků na zapojení jsem si nemohl program vyzkoušet. Ve vizualizaci jsem vymodeloval křižovatku se všemi směry i semafory. Vynechal jsem označení, zda se jedná o denní nebo noční režim. Dalo by se to ovšem vyřešit minimálně slovem někde v levém horním rohu, nebo ztmavením celého pozadí (v případě, že by se jednalo o noční režim).

U nočního režimu jsem neuvedl v referátu stavy, protože se jedná pouze o stav „všechna žlutá/oranžová světla svítí“ a stav „vše vypnuto“.

Přílohy:

* Výpis programu – 6 stran
* Stavový diagram – 1 strana
* Jednotlivé stavy křižovatky (denní režim) – 2 strany

**Stavový Diagram:**



**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 WindowsFormsApplication1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

public class K8055D

{

[DllImport("K8055D.dll")] //Načtení knihovny a její funkce

public static extern int OpenDevice(int CardAddress);

[DllImport("K8055D.dll")] //Načtení knihovny a její funkce

public static extern int ReadDigitalChannel(int Channel);

[DllImport("K8055D.dll")] //Načtení knihovny a její funkce

public static extern int WriteAllDigital(int Data);

[DllImport("K8055D.dll")] //Načtení knihovny a její funkce

public static extern int CloseDevice(int CardAddress);

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

Graphics g = panel1.CreateGraphics(); //Zapnutí grafiky v panel1

Pen p = new Pen(Color.Black, 2); //Definice pera černé barvy s tloušťkou 2

/\*křižovatka\*/

g.DrawLine(p, 50, 200, 200, 200); //Nakreslení modelu křižovatky pomocí čar

g.DrawLine(p, 200, 50, 200, 200);

g.DrawLine(p, 400, 50, 400, 200);

g.DrawLine(p, 400, 200, 550, 200);

g.DrawLine(p, 50, 400, 200, 400);

g.DrawLine(p, 200, 400, 200, 550);

g.DrawLine(p, 400, 550, 400, 400);

g.DrawLine(p, 400, 400, 550, 400);

/\*spodní-prostřední semafor\*/

g.DrawEllipse(p, 300, 410, 20, 20); //Vykreslení jednotlivých semaforů

g.DrawEllipse(p, 300, 435, 20, 20);

g.DrawEllipse(p, 300, 460, 20, 20);

/\*spodní-krajní semafor\*/

g.DrawEllipse(p, 405, 410, 20, 20);

g.DrawEllipse(p, 405, 435, 20, 20);

g.DrawEllipse(p, 405, 460, 20, 20);

/\*horní-prostřední semafor\*/

g.DrawEllipse(p, 280, 120, 20, 20);

g.DrawEllipse(p, 280, 145, 20, 20);

g.DrawEllipse(p, 280, 170, 20, 20);

/\*horní-krajní semafor\*/

g.DrawEllipse(p, 175, 120, 20, 20);

g.DrawEllipse(p, 175, 145, 20, 20);

g.DrawEllipse(p, 175, 170, 20, 20);

/\*levý semafor\*/

g.DrawEllipse(p, 175, 410, 20, 20);

g.DrawEllipse(p, 150, 410, 20, 20);

g.DrawEllipse(p, 125, 410, 20, 20);

g.DrawEllipse(p, 175, 435, 20, 20);

/\*pravý semafor\*/

g.DrawEllipse(p, 405, 170, 20, 20);

g.DrawEllipse(p, 430, 170, 20, 20);

g.DrawEllipse(p, 455, 170, 20, 20);

g.DrawEllipse(p, 405, 145, 20, 20);

}

private void Form1\_Load(object sender, EventArgs e)

{

Graphics g = panel1.CreateGraphics(); //Zapnutí grafiky v panel1

SolidBrush b1 = new SolidBrush(Color.Red); //Definice vybarvování červenou barvou

SolidBrush b2 = new SolidBrush(Color.Yellow); //Definice vybarvování žlutou barvou

SolidBrush b3 = new SolidBrush(Color.Green); //Definice vybarvování zelenou barvou

SolidBrush b4 = new SolidBrush(Color.White); //Definice vybarvování bílou barvou

int status1 = 0; //Nastaví proměnou status1 hodnotou 0

int status2 = 0; //Nastaví proměnou status2 hodnotou 0

int stav1 = 0; //Nastaví proměnou stav1 hodnotou 0

int stav2 = 0; //Nastaví proměnou stav2 hodnotou 0

int R1, R2; //Definice proměnných

int den, noc; //Definice proměnných

R1 = 0x00; //Poslání nul do registru R1

R2 = 0x00; //Poslání nul do registru R2

while (true) //nekonečná smyčka

{

K8055D.OpenDevice(0); //připojí (softwarově) první kartu

den = K8055D.ReadDigitalChannel(1); //Poslání hodnoty přečtené na vstupu první karty do proměnné den

noc = K8055D.ReadDigitalChannel(2); //Poslání hodnoty přečtené na vstupu první karty do proměnné noc

K8055D.CloseDevice(0); //odpojí (softwarově) první kartu

if (den == 0) //když bylo stisknuto tlačítko pro volbu denního režimu

{

noc = 1; //nastaví noc na 1, z důvodu předejití samovolné změny

switch (status1) //vybírá činnost podle hodnoty status1

{

case 0: //Výpis hodnot na porty

{

K8055D.OpenDevice(0); //připojí (softwarově) první kartu

K8055D.WriteAllDigital(R1); //Vypíše na výstup hodnotu z R1

K8055D.CloseDevice(0); //odpojí (softwarově) první kartu

K8055D.OpenDevice(3); //připojí (softwarově) první kartu

K8055D.WriteAllDigital(R2); //Vypíše na výstup hodnotu z R2

K8055D.CloseDevice(3); //odpojí (softwarově) první kartu

status1 = stav1 + 1; //Zvýší hodnotu stav1 o 1 a uloží ji do proměnné status1

if (status1 == 7) //Když se status1 rovná 7

{

status1 = 1; //Je nastaven status1 na 1

}

}

break; //Ukončení číinnosti

case 1: //První kombinace světel

{

R1 = 0x31; //Uloží do R1 kobinaci pro rozvícení příslušných světel křižovatky

R2 = 0x44; //Uloží do R2 kobinaci pro rozvícení příslušných světel křižovatky

stav1 = status1; //Přesune do svav1 hodnotu status1

status1 = 0; //nastaví status1 na 0

Task.Delay(3000); //zpoždění 3s

g.FillEllipse(b1, 300, 410, 20, 20); //Následně vykresluje příslušné barvy světel do panel1

g.FillEllipse(b3, 405, 460, 20, 20);

g.FillEllipse(b1, 280, 120, 20, 20);

g.FillEllipse(b3, 175, 120, 20, 20);

g.FillEllipse(b1, 175, 410, 20, 20);

g.FillEllipse(b1, 405, 170, 20, 20);

}

break;

case 2: //Druhá kombinace světel

{

R1 = 0x52;

R2 = 0x42;

stav1 = status1;

status1 = 0;

Task.Delay(3000);

g.FillEllipse(b2, 300, 435, 20, 20);

g.FillEllipse(b2, 405, 435, 20, 20);

g.FillEllipse(b2, 280, 145, 20, 20);

g.FillEllipse(b2, 175, 145, 20, 20);

g.FillEllipse(b1, 175, 410, 20, 20);

g.FillEllipse(b1, 405, 170, 20, 20);

}

break;

case 3: //Třetí kombinace světel

{

R1 = 0x54;

R2 = 0x11;

stav1 = status1;

status1 = 0;

Task.Delay(3000);

g.FillEllipse(b1, 300, 410, 20, 20);

g.FillEllipse(b3, 300, 460, 20, 20);

g.FillEllipse(b1, 280, 120, 20, 20);

g.FillEllipse(b3, 175, 170, 20, 20);

g.FillEllipse(b3, 175, 435, 20, 20);

g.FillEllipse(b3, 405, 145, 20, 20);

}

break;

case 4: //Čtvrtá kombinace světel

{

R1 = 0x54;

R2 = 0x22;

stav1 = status1;

status1 = 0;

Task.Delay(3000);

g.FillEllipse(b2, 300, 435, 20, 20);

g.FillEllipse(b1, 405, 410, 20, 20);

g.FillEllipse(b2, 280, 145, 20, 20);

g.FillEllipse(b1, 175, 170, 20, 20);

g.FillEllipse(b2, 150, 410, 20, 20);

g.FillEllipse(b2, 430, 170, 20, 20);

}

break;

case 5: //Pátá kombinace světel

{

R1 = 0x4C;

R2 = 0x0C;

stav1 = status1;

status1 = 0;

Task.Delay(3000);

g.FillEllipse(b1, 300, 410, 20, 20);

g.FillEllipse(b1, 405, 410, 20, 20);

g.FillEllipse(b1, 280, 170, 20, 20);

g.FillEllipse(b1, 175, 170, 20, 20);

g.FillEllipse(b3, 125, 410, 20, 20);

g.FillEllipse(b3, 455, 170, 20, 20);

}

break;

case 6: //Šestá kombinace světel

{

R1 = 0x52;

R2 = 0x24;

stav1 = status1;

status1 = 0;

Task.Delay(3000);

g.FillEllipse(b1, 300, 410, 20, 20);

g.FillEllipse(b2, 405, 435, 20, 20);

g.FillEllipse(b1, 280, 170, 20, 20);

g.FillEllipse(b2, 175, 145, 20, 20);

g.FillEllipse(b2, 150, 410, 20, 20);

g.FillEllipse(b2, 430, 170, 20, 20);

}

break;

}

}

if (noc == 0) //když bylo stisknuto tlačítko pro volbu nočního režimu

{

den = 1; //nastaví den na 1, z důvodu předejití samovolné změny

switch (status2) //vybírá činnost podle hodnoty status2

{

case 0: //Výpis hodnot na porty

{

K8055D.OpenDevice(0);

K8055D.WriteAllDigital(R1);

K8055D.CloseDevice(0);

K8055D.OpenDevice(3);

K8055D.WriteAllDigital(R2);

K8055D.CloseDevice(3);

status2 = stav2 + 1;

if (status2 == 3)

{

status2 = 1;

}

}

break;

case 1: //První kombinace světel

{

R1 = 0x00;

R2 = 0x00;

stav2 = status2;

status2 = 0;

Task.Delay(2000);

g.FillEllipse(b4, 300, 435, 20, 20);

g.FillEllipse(b4, 405, 435, 20, 20);

g.FillEllipse(b4, 280, 145, 20, 20);

g.FillEllipse(b4, 175, 145, 20, 20);

g.FillEllipse(b4, 150, 410, 20, 20);

g.FillEllipse(b4, 430, 170, 20, 20);

}

break;

case 2: //Druhá kombinace světel

{

R1 = 0x02;

R2 = 0x22;

stav2 = status2;

status2 = 0;

Task.Delay(1000);

g.FillEllipse(b2, 300, 435, 20, 20);

g.FillEllipse(b2, 405, 435, 20, 20);

g.FillEllipse(b2, 280, 145, 20, 20);

g.FillEllipse(b2, 175, 145, 20, 20);

g.FillEllipse(b2, 150, 410, 20, 20);

g.FillEllipse(b2, 430, 170, 20, 20);

}

break;

}

}

}

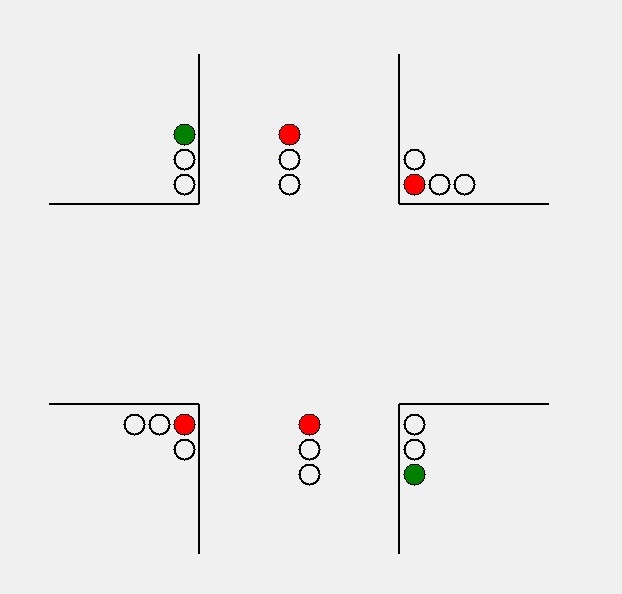
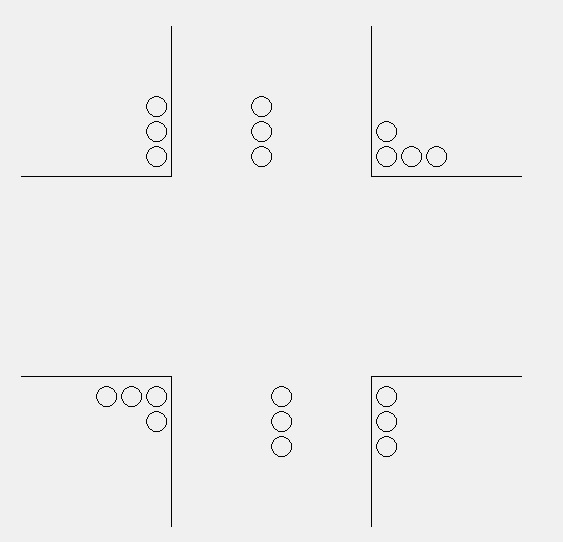
}

}

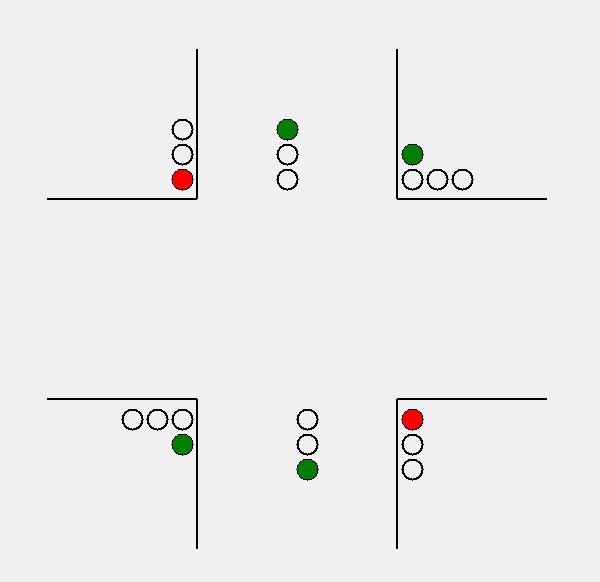
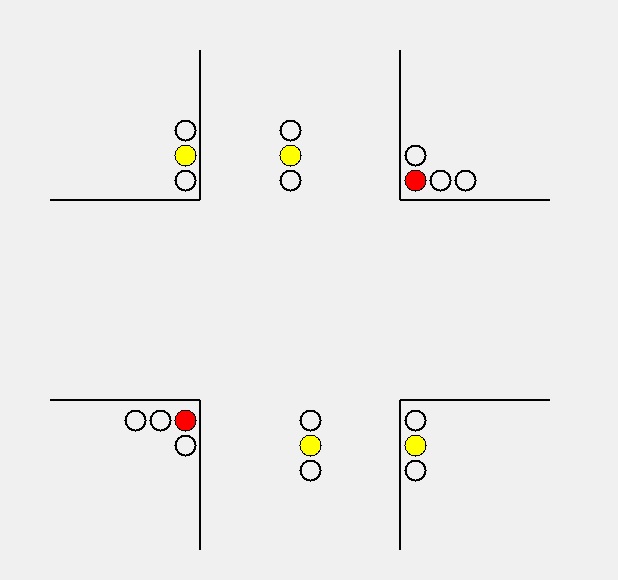
}

**Jednotlivé stavy křižovatky:**

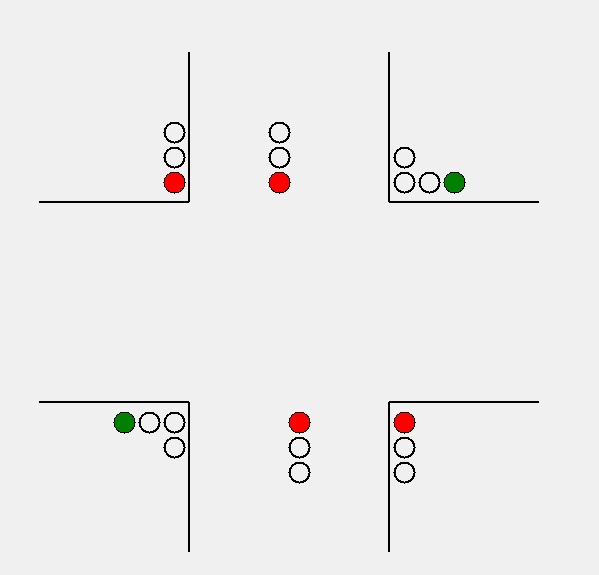
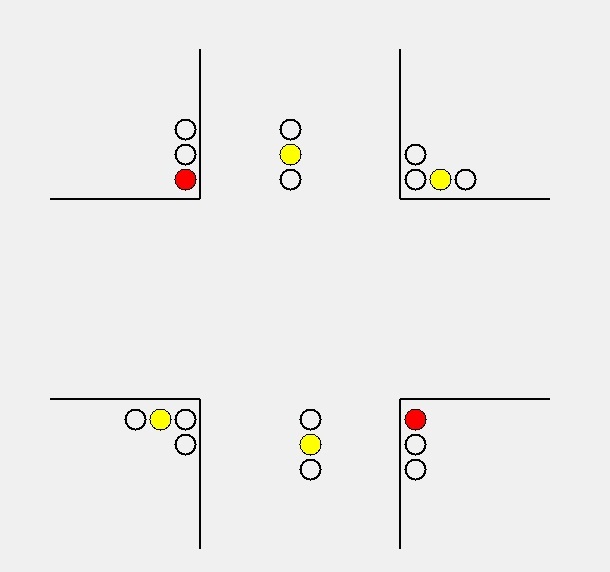
Výchozí stav: První stav:



Druhý stav: Třetí stav:



Čtvrtý stav: Pátý stav:



Šestý člen:

