БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

Ханты-Мансийского автономного округа – Югры

«Сургутский государственный университет»

Политехнический институт Кафедра автоматики и компьютерных систем

ОТЧЕТ

по лабораторной работе

дисциплина Теория Вычислительных Процессов

Тема: Модель Светофора

Выполнил: студент группы 609-21

Яковлев Алексей Викторович

Принял: доцент кафедры АиКС

Брагинский М. Я.

Сургут 2025 г.

**Задание:**

1. Описать работу ЦА (управляющего двухсекционным светофором) диаграммой переходов, таблицей переходов
2. Создать программную модель ЦА (на любом ЯП) с отображением состояний, входных и выходных сигналов (таймер и лампы)

**Граф работы (рисунок 1):**

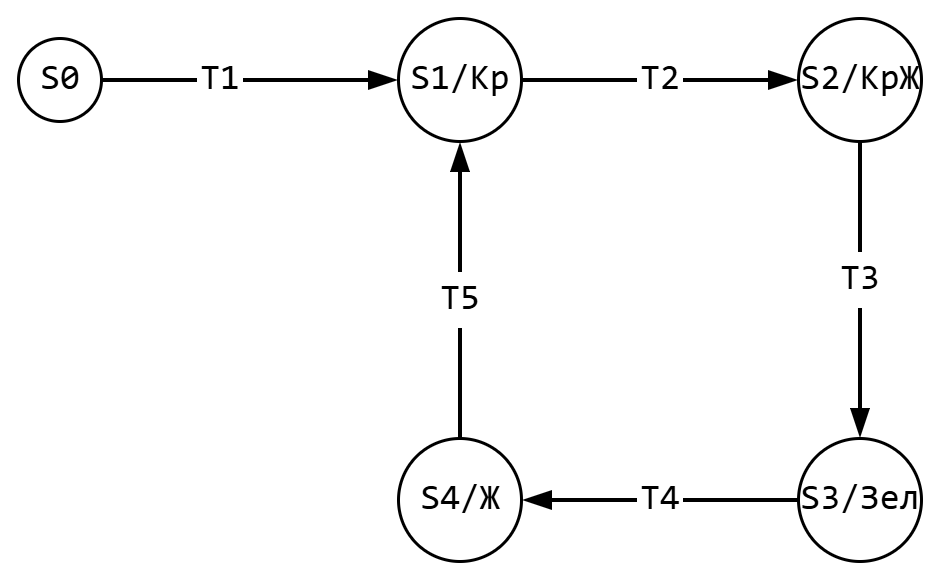


Рисунок 1 - Граф работы ЦА.

**Таблица переходов (таблица 1):**

Таблица 1 – Таблица переходов ЦА

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Время** | **Состояния** | | | | |
| S0 | S1 | S2 | S3 | S4 |
| T1 = 3 | S1 | - | - | - | - |
| T2 = 13 | - | S2 | - | - | - |
| T3 = 16 | - | - | S3 | - | - |
| T4 = 26 | - | - | - | S4 | - |
| T5 = 29 | - | - | - | - | S1 |

**Состояния:**

S0 – Светофор выключен, не горит ни один сигнал светофора;

S1 – Горит красный сигнал светофора;

S2 – Горят красный и желтый сигналы светофора;

S3 – Горит зеленый сигнал светофора;

S4 – Горит желтый сигнал светофора.

**Код программного обеспечения (листинг 1):**

Код был написан на языке программирования С# c MAUI в среде разработки Visual Studio 2022.

Листинг 1. Код программного обеспечения:

using System.Timers;

namespace lab1.\_1

{

public partial class MainPage : ContentPage

{

private enum LightState { Off, Red, RedYellow, Green, Yellow }

private LightState currentState = LightState.Off;

private int timeLeft;

private readonly Label timerLabel;

private readonly BoxView redLight, yellowLight, greenLight;

private readonly BoxView pedestrianRed, pedestrianGreen;

private readonly System.Timers.Timer timer;

public MainPage()

{

timerLabel = new Label { FontSize = 24, HorizontalOptions = LayoutOptions.Center };

redLight = new BoxView { Color = Colors.Gray, WidthRequest = 50, HeightRequest = 50, CornerRadius = 25 };

yellowLight = new BoxView { Color = Colors.Gray, WidthRequest = 50, HeightRequest = 50, CornerRadius = 25 };

greenLight = new BoxView { Color = Colors.Gray, WidthRequest = 50, HeightRequest = 50, CornerRadius = 25 };

pedestrianRed = new BoxView { Color = Colors.Gray, WidthRequest = 50, HeightRequest = 50, CornerRadius = 25 };

pedestrianGreen = new BoxView { Color = Colors.Gray, WidthRequest = 50, HeightRequest = 50, CornerRadius = 25 };

StackLayout lights = new StackLayout

{

Children = { redLight, yellowLight, greenLight, timerLabel },

Spacing = 10,

VerticalOptions = LayoutOptions.CenterAndExpand,

HorizontalOptions = LayoutOptions.Center

};

StackLayout pedestrianLights = new StackLayout

{

Children = { pedestrianRed, pedestrianGreen },

Spacing = 10,

VerticalOptions = LayoutOptions.CenterAndExpand,

HorizontalOptions = LayoutOptions.Center

};

Content = new HorizontalStackLayout

{

Children = { lights, pedestrianLights },

Spacing = 50,

VerticalOptions = LayoutOptions.CenterAndExpand,

HorizontalOptions = LayoutOptions.Center

};

timer = new System.Timers.Timer(1000);

timer.Elapsed += OnTimedEvent;

timer.AutoReset = true;

timer.Start();

SetLightState(LightState.Off, 2);

}

private void OnTimedEvent(object sender, ElapsedEventArgs e)

{

if (timeLeft > 0)

{

timeLeft--;

Device.BeginInvokeOnMainThread(() => timerLabel.Text = timeLeft.ToString());

}

else

{

switch (currentState)

{

case LightState.Off:

SetLightState(LightState.Red, 9);

break;

case LightState.Red:

SetLightState(LightState.RedYellow, 2);

break;

case LightState.RedYellow:

SetLightState(LightState.Green, 9);

break;

case LightState.Green:

SetLightState(LightState.Yellow, 2);

break;

case LightState.Yellow:

SetLightState(LightState.Red, 9);

break;

}

}

}

private void SetLightState(LightState state, int duration)

{

currentState = state;

timeLeft = duration;

redLight.Color = (state == LightState.Red || state == LightState.RedYellow) ? Colors.Red : Colors.Gray;

yellowLight.Color = (state == LightState.Yellow || state == LightState.RedYellow) ? Colors.Yellow : Colors.Gray;

greenLight.Color = state == LightState.Green ? Colors.Green : Colors.Gray;

pedestrianRed.Color = (state == LightState.Green || state == LightState.Yellow || state == LightState.RedYellow ) ? Colors.Red : Colors.Gray;

pedestrianGreen.Color = (state == LightState.Red) ? Colors.Green : Colors.Gray;

Device.BeginInvokeOnMainThread(() => timerLabel.Text = timeLeft.ToString());

}

}

}

Листинг 2. Код интерфейса:

<?xml version="1.0" encoding="utf-8" ?>

<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="lab1.\_1.MainPage">

<VerticalStackLayout VerticalOptions="Center" HorizontalOptions="Center" Spacing="10">

<BoxView x:Name="RedLight" Color="Gray" WidthRequest="50" HeightRequest="50" CornerRadius="25" />

<BoxView x:Name="YellowLight" Color="Gray" WidthRequest="50" HeightRequest="50" CornerRadius="25" />

<BoxView x:Name="GreenLight" Color="Gray" WidthRequest="50" HeightRequest="50" CornerRadius="25" />

<Label x:Name="TimerLabel" FontSize="24" HorizontalOptions="Center" />

</VerticalStackLayout>

</ContentPage>

Пример работы программы (рисунки 2-6):



Рисунок 2 – Светофор в состоянии S0

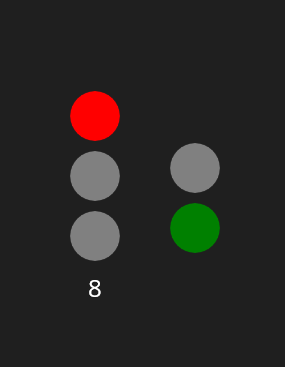


Рисунок 3 – Светофор в состоянии S1

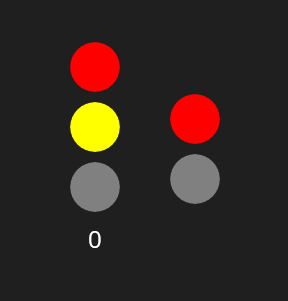


Рисунок 4 – Светофор в состоянии S2

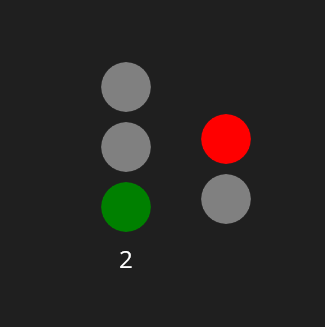


Рисунок 5 – Светофор в состоянии S3

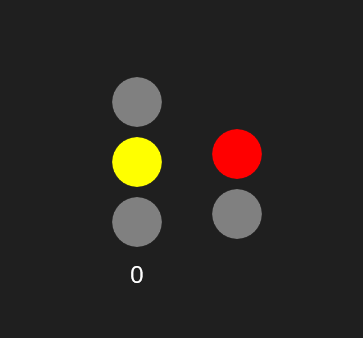


Рисунок 6 – Светофор в состоянии S4

Выводы: в ходе лабораторной работы была создана программная реализация цифрового автомата модели светофора.