

3.4 Отображение информации в текстовом и графическом виде.

Для отображения информации в текстовом виде мы будем использовать элемент управления TextBox, а отображение информации в графическом виде (подобие экрана осциллографа) – элемент Chart.

Работу этих элементов рассмотрим на примере учебной задачи – отобразить в окне программы «временные диаграммы» работы трехразрядного суммирующего двоичного счетчика. Переключение счетчика осуществлять тактирующими импульсами, формируемыми по прерыванию от элемента Timer. Запуск и остановку таймера будем осуществлять одной кнопкой «Пуск – Стоп». Элемент таймер не используется в наших программах обмена данными ПК и МК, но он избавляет нас от многократного нажимания на кнопку при имитации тактирующих импульсов.

Проект разработан в WindowsFormsApplication. С помощью мастера визуального программирования и панели Toolbox разместим на форме проекта необходимые элементы управления (рисунок 3.5).

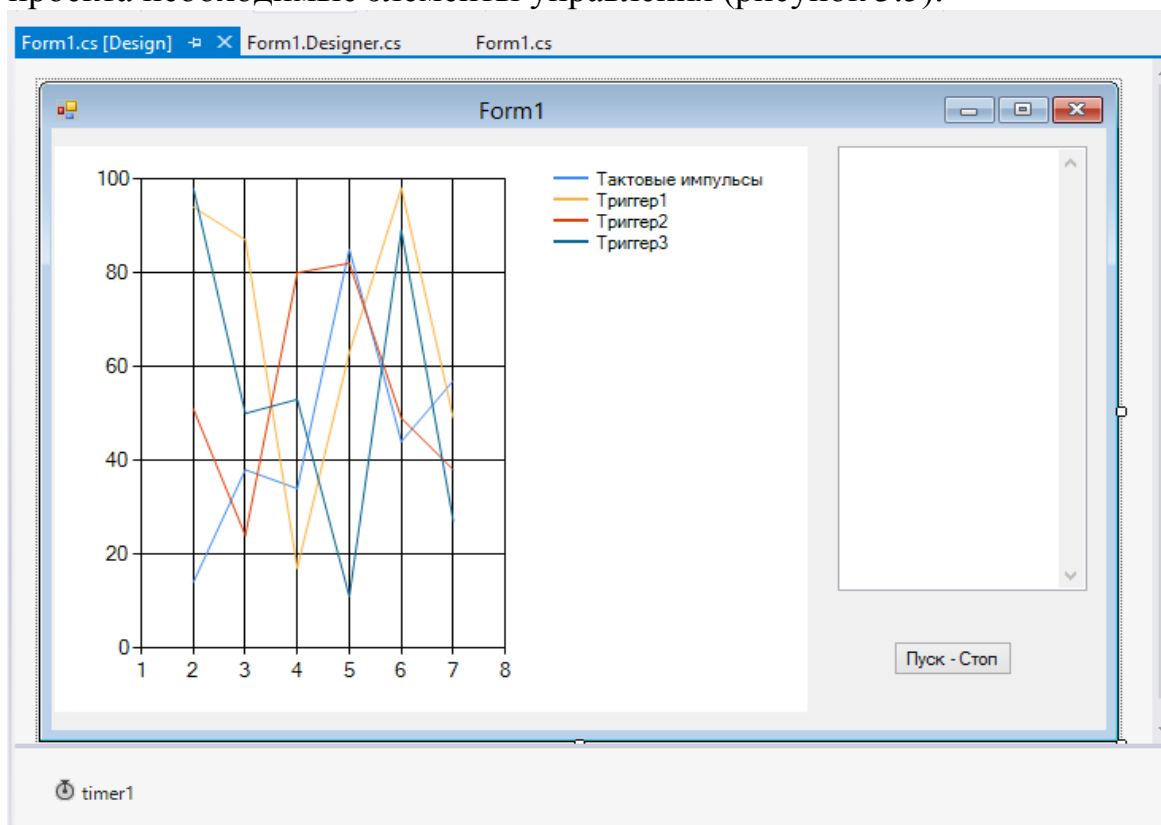


Рисунок 3.5 - Размещение элементов управления на форме проекта

В свойствах элемента chart1 (рисунок 3.6) редактируем позицию Series в соответствии с рисунком 3.7.

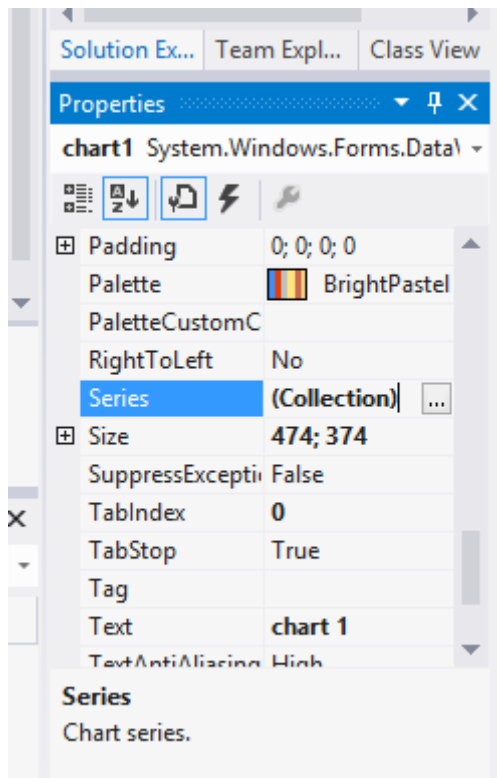


Рисунок 3.6 - Свойства элемента управления Chart1

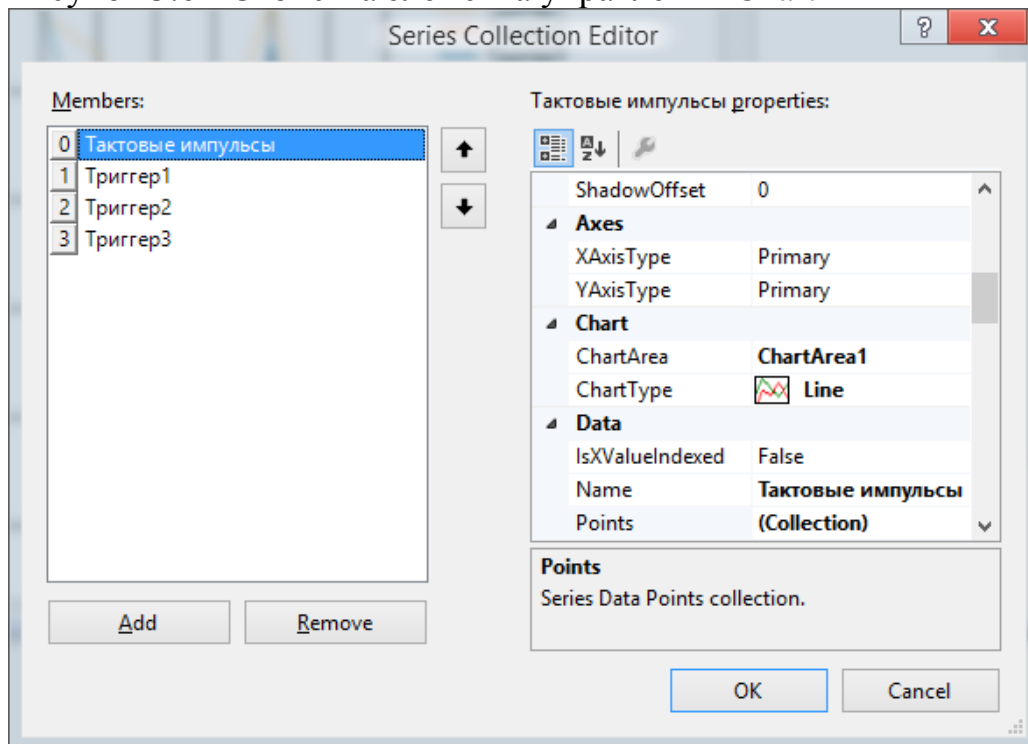


Рисунок 3.7 - Окно редактирования Series элемента Chart1

Для элемента timer1 в его свойствах на страничке Events устанавливаем прерывание timer1_Tick, а на общей страничке свойств задаем свойству Interval значение 1000 - время прерывания 1 секунда. В результате работы со

свойствами элемента timer1 в коде формы1 появится "пустая заготовка" на прерывания от таймера:

```
private void timer1_Tick(object sender, EventArgs e)
{
}
```

Для элемента Button1 в его свойствах редактируем значение свойства Text - устанавливаем «Пуск – Стоп». Двойным нажатием левой кнопки мышки на изображение кнопки на форме создадим "пустой обработчик", в который занесем код управления таймером:

```
private void button1_Click(object sender, EventArgs e)
{
    if (timer1.Enabled == false) timer1.Enabled =true;
    else timer1.Enabled =false;
}
```

Разместим в окне формы элемент управления textBox1, для которого зададим значение свойства Multiline равное True. Остальные значения свойств оставляем без изменений.

На этом этап визуального программирования закончен и Мы переходим к этапу написания кода программы.

Для улучшения восприятия кода программы напишем два метода AddPoints и AddText.

В методе AddPoints Мы опишем работу элемента chart1 при формировании очередных "точек" графиков временных диаграмм.

В методе AddText Мы будем формировать значения строковой переменной ss для отображения очередного состояния счетчика в текстовом виде в окне элемента textBox1. Состояние счетчика будем отображать в традиционном табличном виде с помощью нулей и единиц.

В обработчике прерывания таймера реализован алгоритм сложения четырехразрядных "двоичных чисел". Для "независимого" отображения графика каждого разряда "двоичного числа" использованы различные начальные нулевые значения:

```
static public byte[] tim = new byte[4] { 1, 6, 11, 16 };
```

Без учета начальных нулевых значений данные в каждом разряде могут изменяться от 1 - "нулевое" значение, до 4 - "единичное" значение.

Для предания тактовым "импульсам" формы прямоугольников использован цикл for с пятикратным повтором вывода данных в элемент chart1.

При инициализации формы в элементе chart1 формируются "нулевые" значения, а для textBox1 выводится заголовок таблицы.

Исходный код файла Form1.cs:

```
using System;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
```

```

public partial class Form1 : Form
{
    private void AddPoints(byte[] points)
    {
        chart1.Series[0].Points.AddY(points[0]);
        chart1.Series[1].Points.AddY(points[1]);
        chart1.Series[2].Points.AddY(points[2]);
        chart1.Series[3].Points.AddY(points[3]);
    }

    private void AddText(byte[] points)
    {
        string ss = "";
        if (points[3] == 16) ss = ss + " 0"; else ss = ss + " 1";
        if (points[2] == 11) ss = ss + " 0"; else ss = ss + " 1";
        if (points[1] == 6) ss = ss + " 0"; else ss = ss + " 1";
        if (points[0] == 1) ss = ss + " 0 \r\n"; else ss = ss + " 1
\r\n";
        textBox1.AppendText(ss);
    }
    static public byte[] tim = new byte[4] { 1, 6, 11, 16 };
    public Form1()
    {
        InitializeComponent();
        for (int j = 0; j < 5; j++) AddPoints(tim);
        textBox1.Text = " T3 T2 T1 Тим \r\n";
    }

    private void button1_Click(object sender, EventArgs e)
    {
        if (timer1.Enabled == false) timer1.Enabled = true;
        else timer1.Enabled = false;
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        if (tim[3] == 19 && tim[2] == 14 && tim[1] == 9 && tim[0] == 4)
tim[3] = 16;
        else if (tim[3] == 16 && tim[2] == 14 && tim[1] == 9 && tim[0] == 4)
tim[3] = 19;
        if (tim[2] == 14 && tim[1] == 9 && tim[0] == 4) tim[2] = 11;
        else if (tim[2] == 11 && tim[1] == 9 && tim[0] == 4) tim[2] = 14;
        if (tim[1] == 9 && tim[0] == 4) tim[1] = 6;
        else if (tim[1] == 6 && tim[0] == 4) tim[1] = 9;
        if (tim[0] == 4) tim[0] = 1; else tim[0] = 4;
        AddText(tim);
        for (int j = 0; j < 5; j++) AddPoints(tim);
    }
}
}
}

```

Результат работа программы представлен на рисунке 3.8.

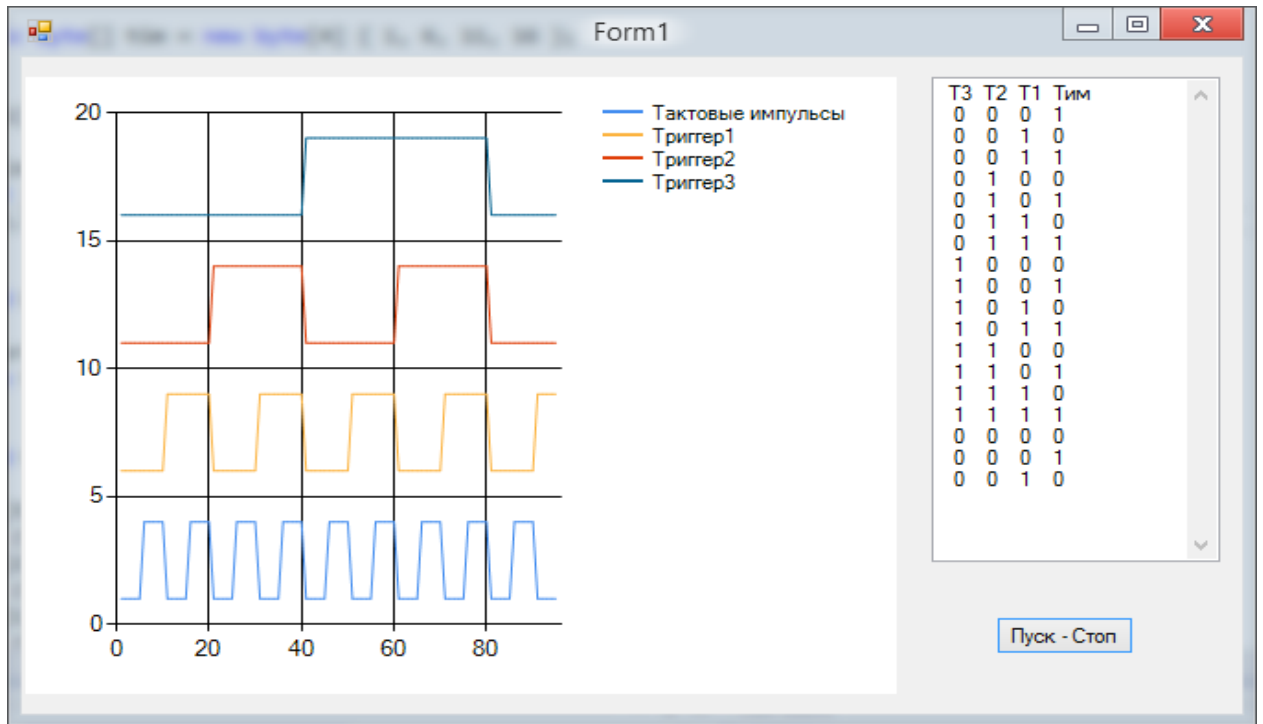


Рисунок 3.8 - Работа программы