

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«Петербургский государственный университет путей сообщения
Императора Александра I»
(ФГБОУ ВО ПГУПС)**

Факультет «Автоматизация и интеллектуальные технологии»
Кафедра «Автоматика и телемеханика на железных дорогах»

Специальность 23.05.05 Системы обеспечения движением поездов
Специализация Автоматика и телемеханика на железнодорожном транспорте

Отчет по практическому заданию № 1

по дисциплине «Микропроцессорные информационно-управляющие
системы»

«Изучение основ интегрированной среды разработки языка
программирования C++ Builder 5»

Обучающийся

дата

Преподаватель

Отметка о зачете, подпись, дата

должность, Ф.И.О.

Санкт-Петербург

Цель практического задания: ознакомление со структурой лабораторной установки, изучение основных программных инструментов Интегрированной среды разработки, освоение методики разработки программных приложений в системе программирования C++ Builder 5 на примере создания проекта «Pause».

1. Краткое описание лабораторной установки

В состав лабораторной установки входят:

- устройство включения питания лабораторной установки «Пилот»;
- устройство бесперебойного питания компьютера (УПС);
- промышленный компьютер;
- монитор;
- клавиатура;
- манипулятор «мышь»;
- блок ввода /вывода дискретных и аналоговых сигналов;
- осциллограф.

В корпусе компьютера установлены пять плат расширения, подключённых к шине ISA компьютера:

- плата PCL-836, которая содержит БИС программно и аппаратно управляемого интервального таймера;
- плата 5600, содержащая программно управляемый БИС для ввода и вывода дискретных сигналов (параллельный, периферийный адаптер ППА);
- плата 5720, содержащая БИС программно управляемого аналогово-цифрового преобразователя для работы с однополярными аналоговыми сигналами;
- плата PCL-818, содержащая программно управляемые БИС аналогово-цифрового и цифро-аналогового преобразователей для работы с биполярными сигналами;

- плата Realtek RTL 8019 (AS) ISA, содержащая сетевой адаптер для включения лабораторной установки в локальную вычислительную сеть.

2. Этапы разработки проекта «Pause»

1. Перенести на Форму компонент Label1 (поле вывода).

Компонент Label предназначен для вывода текста на поверхность формы. Свойства компонента определяют вид и расположение текста.

Таблица 1

| | |
|----------------|---|
| Name | Имя компонента. Используется в программе для доступа к компоненту и его свойствам |
| Caption | Отображаемый текст |
| Font | Шрифт, используемый для отображения текста. Уточняющие свойства определяют шрифт (Name), размер (size), стиль (style) и цвет символов (Color) |
| Color | Цвет фона области вывода текста |
| Visible | Позволяет скрыть текст (false) или сделать его видимым (true) |

2. Перенести на Форму компонент Label2.

3. Перенести на Форму компонент Button1 (кнопка). Компонент Button представляет собой командную кнопку. Свойства:

Таблица 2

| | |
|----------------|---|
| Name | Имя компонента. Используется в программе для доступа к компоненту и его свойствам |
| Caption | Текст на кнопке |
| Enabled | Признак доступности кнопки. Если значение свойства равно true , то кнопка доступна. Если значение свойства равно false , то кнопка недоступна— например, в результате щелчка на кнопке, событие Click не возникает |

4. Перенести на Форму компонент Button2.

5. Перенести на Форму компонент Timer (таймер). Компонент Timer обеспечивает генерацию последовательности событий onTimer. Свойства:

Таблица 3

| | |
|-----------------|--|
| Name | Имя компонента. Используется для доступа к компоненту |
| Interval | Период генерации события OnTimer . Задается в миллисекундах |
| Enabled | Разрешение работы. Разрешает (значение true) или запрещает (значение false) генерацию события OnTimer |

6. Занести текст «Входная переменная» в заголовок компонента Label1, для чего в окне Инспектора объектов изменить текст свойства Caption (по умолчанию – «Label1»);

7. Занести текст «Пуск» в заголовок компонента Button1, для чего в окне Инспектора объектов изменить текст свойства Caption (по умолчанию – «Button1»);

8. Аналогично занести текст «Закрыть» в заголовок компонента Button2;

9. Установить необходимые свойства для компонента Timer1, включая время срабатывания по заданному варианту:

- изменить свойство Enabled (разрешение), которое по умолчанию имеет вид true, на false, тем самым запрещая работу Timer1;
- изменить свойство Interval (интервал), которое по умолчанию имеет 1000 мс, на значение 2000 мс.

10. Создать обработчики событий для компонентов Button1, Button2, Timer1. Реакцией на событие должно быть какое-либо действие. В C++ Builder реакция на событие реализуется как функция обработки события. Для того чтобы добавить обработчик событий, нужно выбрать на Форме с помощью мыши компонент, которому необходим обработчик событий, затем открыть страницу событий инспектора объектов и дважды щелкнуть левой клавишей мыши на колонке значений рядом с событием, чтобы заставить C++ Builder сгенерировать прототип обработчика событий и показать его в

редакторе кода. При этом автоматически генерируется текст пустой функции, и редактор открывается в том месте, где следует вводить код. Обработчик событий может иметь параметры, которые указываются после имени функции в круглых скобках.

11. Объявление переменной (*y*) как беззнакового короткого целого числа.

12. Инициализация переменной: в функцию конструктора формы заносится начальное значение переменной *y*.

13. Разрешение работы таймера: в функцию обработчика события (нажатие клавиши «Пуск») заносится разрешение на работу Таймера.

14. В функцию обработчика события (нажатие клавиши «Заккрыть») заносится функцию закрытия приложения.

15. Обработка переменной *y* по заданному алгоритму работы: в функцию обработчика событий таймера заносятся операторы, отвечающие за изменение параметров поля вывода (цвет, шрифт, положение и т.д.).

16. Компиляция проекта.

3. Блок-схемы алгоритмов проекта в целом и функции Timer1.Timer

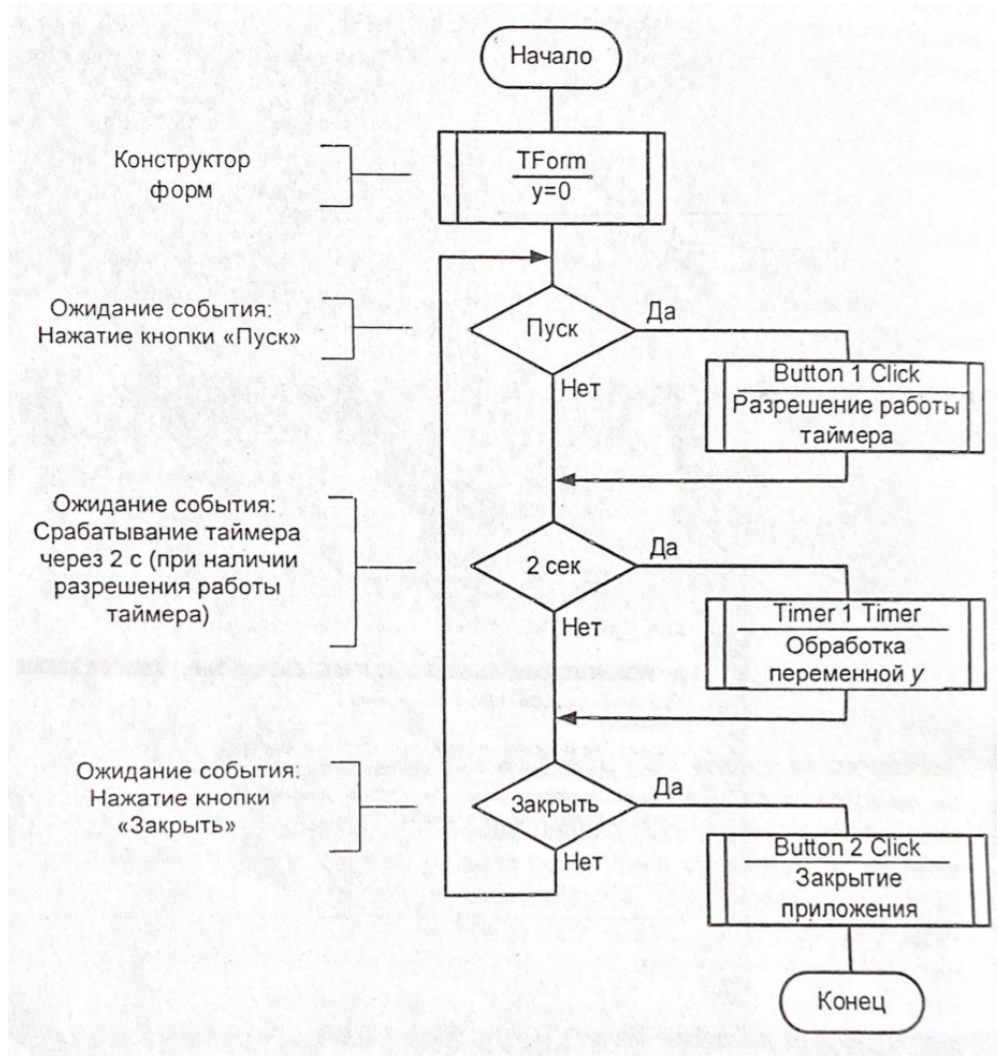


Рис. 1 – блок-схема алгоритма проекта «Пауза»

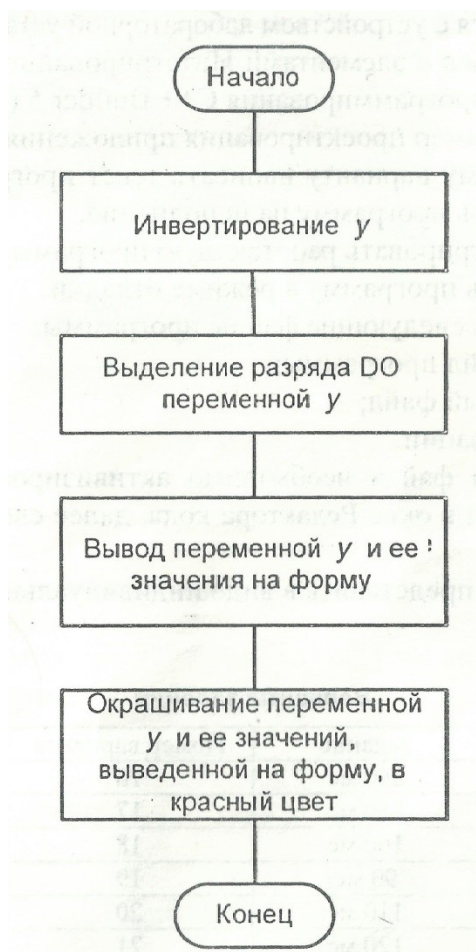


Рис. 2 – блок-схема алгоритма функции Timer1.Timer

4. Программные листинги заголовочного, головного файлов и файла реализации проекта

Код головного файла:

```

PauseProject 1.cpp      30.11.22      19:45:54

//-----
#include <vcl.h> // подключение библиотеки визуальных компонентов
#pragma hdrstop // директива препроцессора указывает компилятору
                // конец списка общих файлов
USERES("PauseProject1.res");
USEFORM("PauseUnit.cpp",Form1);
//-----
// главная функция
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
try // обработка аварийных ситуаций при работе программы

```

```

{
    Application->Initialize(); // инициализация компонентов приложения
    Application->CreateForm(__classid(TForm1), &Form1); // создание
                                // компонентов, расположенных на Форме
    Application->Run(); // включение программы на выполнение
}
catch (Exception &exception) // выполнение операторов при аварийной
                                // ситуации
{
    Application->ShowException(&exception);
}
return 0; // завершение проекта
}
//-----

```

Код заголовочного файла:

PauseUnit1.h 30.11.22 19:45:54

```

#include <ExtCtrls.hpp>
//-----
class TForm1: public TForm // объявление класса формы TForm1
{
    published: // IDE-managed Components
    TLabel*Label1;
    TLabel*Label2;
    TButton*Button1;
    TButton*Button2;
    TTimer*Timer1;
    void __fastcall Button1Click(TObject *Sender);
    void __fastcall Button2Click(TObject *Sender);
    void __fastcall Timer1Timer(TObject *Sender);
// закрытый раздел класса, где могут помещаться объявления типов
// переменных, функций, включаемых в класс формы, но недоступных
// для других модулей
private: // User declarations
// открытый раздел класса, где могут помещаться объявления типов
// переменных, функций, включаемых в класс формы и доступных
// для других модулей
public: // User declarations
    __fastcall TForm1(TComponent* Owner); // объявление конструктора формы
};
//-----
// ранее объявленный объект Форма включён по умолчанию в пакетный

```



```
// исходный файл
extern PACKAGE TForm1 *Form1;
//-----
#endif // директива условной компиляции препроцессора
```

Код файла реализации:

PauseUnit1.cpp 30.11.22 19:45:54

```
1: //-----
2:
3: #include <vcl.h> // директива препроцессора подключает файлы
4: // визуальных компонентов
5: #pragma hdrstop // директива препроцессора указывает компилятору
6: // конец списка общих файлов
7: #include "PauseUnit1.h" // директива препроцессора подключает в файл
8: // реализации заголовочный файл
9: //-----
10: #pragma package(smart_init) // директива препроцессора указывает
11: // компилятору последовательность
12: // инициализации пакетов такой, какая
13: // устанавливается взаимными ссылками
14: // использующих их модулей
15: #pragma resource "*.dfm" // директива препроцессора указывает
16: // компилятору, что для формы надо
17: // использовать файл .dfm с тем же именем,
18: // что и имя данного файла
19: TForm1 *Form1; // объявление указателя *Form1 для объекта формы
20: // TForm1
21: unsigned short int y; // объявление переменной "y" как беззнакового,
22: // короткого, целого числа
23: //-----
24: // Объявление конструктора формы и функции конструктора формы
25: __fastcall TForm1::TForm1(TComponent* Owner)
26: : TForm(Owner)
27: {
28: y=0; // начальное значение переменной y
29: }
30: //-----
31:
32:
33: // Объявление наличия кнопки Button1 (Пуск)
34: void __fastcall TForm1::Button1Click(TObject *Sender)
35: {
```

```

36: Timer1->Enabled=true;    // Разрешение работы таймера
37:
38: }
39: //-----
40:
41: // Объявление наличия кнопки Button2 (Закреть)
42: void __fastcall TForm1::Button2Click(TObject *Sender)
43: {
44: Close();    // Закрывает форму и завершение приложения
45:
46: }
47: //-----
48:
49: // Объявление наличия таймера Timer1
50: void __fastcall TForm1::Timer1Timer(TObject *Sender)
51: {
52: y=~y;    // Инвертирование y
53: y=y&0x01;    // Выделение разряда D0 переменной y
54: // Вывод значения переменной "y" на форму
55: Label2->Caption="y="+AnsiString(y);
56: // Окрашивание переменной "y" в красный цвет
57: Label2->Font->Color=clRed;
58: }
59: //-----

```

Вывод

В ходе выполнения практического занятия была освоена методика разработки программных приложений в интегрированной среде разработки C++ Builder 5. Были изучены функции и свойства некоторых основных компонентов C++ Builder 5, таких как Label, Button, Timer. На основе полученных навыков создан проект Pause, реализующий вывод значения переменной на Форму через определённый промежуток времени, задаваемый компоненту Timer (Таймер). Из создаваемых программой автоматически 6 файлов рассмотрены подробно 3: заголовочный, головной и файл реализации. В работе приведены коды вышеуказанных файлов, полученные после сохранения проекта. Также приложены блок-схемы алгоритмов, реализованных в данной работе.