



**Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

**ФАКУЛЬТЕТ** Информатика и системы управления (ИУ)

**КАФЕДРА** «Информационная безопасность» (ИУ8)

## **МОДУЛЬ ВНЕШНИХ УСТРОЙСТВ ПРОГРАММНОГО СИМУЛЯТОРА PDP-11**

**Пояснительная записка**

**А.В.00001-01 81 02**

**листов 10**

## **Аннотация**

В данном программном документе приведена пояснительная записка к модулю внешних устройств для программного симулятора PDP-11. Модуль предназначен для подключения внешних устройств к симулятору.

## Содержание

Аннотация.....	2
Основная часть.....	4
1 Введение.....	4
1.1 Наименование программы.....	4
1.2 Условное обозначение темы разработки.....	4
2 Назначение и область применения.....	4
2.1 Назначение программы.....	4
2.2 Область применения программы.....	4
3 Технические характеристики.....	4
3.1 Постановка задачи на разработку программы.....	4
3.2 Описание функционирования программы.....	5
3.2.1 Описание интерфейса внешнего устройства.....	5
3.2.2 Описание функционирования программы.....	6
3.2.3 Возможные взаимодействия программы с другими программами.....	6
3.2.4 Описание и обоснование метода организации входных данных.....	7
3.2.5 Описание и обоснование метода организации выходных данных.....	7
3.2.6 Описание и обоснование выбора состава технических средств.....	7
3.2.7 Описание и обоснование выбора состава программных средств.....	7
4 Ожидаемые технико-экономические показатели.....	7
5 Источники, используемые при разработке.....	8
Лист регистрации изменений.....	9

## **Основная часть**

### **1 Введение**

#### **1.1 Наименование программы**

Наименование – «Модуль внешних устройств программного симулятора PDP-11».

#### **1.2 Условное обозначение темы разработки**

Наименование темы разработки – «Программный симулятор PDP-11».

Условное обозначение темы разработки (шифр темы) – «?».

### **2 Назначение и область применения**

#### **2.1 Назначение программы**

Основное назначение модуля внешних устройств (ВУ) – подключение и внешних устройств к модулю исполнителя программного симулятора PDP-11.

Под ВУ понимается динамическая библиотека (DLL), которая служит драйвером между исполнителем и реальным внешним устройством. Далее под ВУ будет понимать именно библиотеку.

Модуль предоставляет интерфейс, через который исполнитель способен обращаться к ВУ.

Также вместе с модулем предоставляется SDK для разработки ВУ.

#### **2.2 Область применения программы**

Программа предназначена к применению в учебных целях для разработки программ для ЭВМ PDP-11 с использованием ВУ и для разработки самих ВУ.

### **3 Технические характеристики**

#### **3.1 Постановка задачи на разработку программы**

ЭВМ PDP-11 предоставляет возможность своим пользователям подключать к общей шине множество устройств ввода-вывода. Модуль ВУ позволяет реализовать данную возможность у программного симулятора.

## 3.2 Описание функционирования программы

### 3.2.1 Описание интерфейса внешнего устройства

Вместе с программой поставляется SDK, позволяющий разрабатывать ВУ. SDK разработан с помощью языка программирования C# на платформе .NET6. SDK предоставляет интерфейс внешнего устройства (см. листинг 1).

Листинг 1. Интерфейс ВУ

```
public interface IDevice : IDisposable
{
    /// <summary>
    /// Имя устройства, используется для отображения в окне исполнителя
    /// </summary>
    string Name { get; }

    /// <summary>
    /// Адрес регистра буфера
    /// </summary>
    ushort BufferRegisterAddress { get; }

    /// <summary>
    /// Адрес регистра управления
    /// </summary>
    ushort ControlRegisterAddress { get; }

    /// <summary>
    /// Адрес вектора прерывания
    /// </summary>
    ushort InterruptVectorAddress { get; }

    /// <summary>
    /// Флаг запроса прерывания
    /// </summary>
    bool HasInterrupt { get; }

    /// <summary>
    /// Значение регистра буфера
    /// </summary>
    ushort BufferRegisterValue { get; set; }

    /// <summary>
    /// Значение регистра управления
    /// </summary>
    ushort ControlRegisterValue { get; set; }

    /// <summary>
    /// Функция инициализации
    /// </summary>
    int Init();

    /// <summary>
    /// Функция обработки прерывания, вызывается исполнителем, чтобы сообщить ВУ,
    что он принял прерывание в обработку

```

```
/// </summary>
void AcceptInterrupt();
}
```

### 3.2.2 Описание функционирования программы

Программа имеет следующую ООП структуру:

- `DeviceManager` – класс, хранящий все открытые ранее ВУ – предоставляет возможность добавлять новые и удалять ранее открытые ВУ;
- `DeviceProvider` – класс, позволяющий открывать ВУ;
- `DeviceValidator` – класс, позволяющий проверять ВУ на корректность.

`DeviceProvider` реализует метод *Load(string): IDevice*, который принимает путь к ВУ и возвращает объект, реализующий интерфейс ВУ.

`DeviceManager` принимает на вход путь к ВУ, который передает в `DeviceProvider`. Полученное от провайдера ВУ он сохраняет

`DeviceProvider` реализует следующие методы:

- *Add(string): void* – принимает на вход путь к ВУ, которые передается в `DeviceProvider`. Полученное от последнего ВУ сохраняется в список открытых ВУ;
- *Delete(string): void* – закрывает открытое ВУ, и очищает использованные им ресурсы, и удаляет его из списка открытых устройств;
- *Clear()* – закрывает все ВУ;
- *GetDevices(): List<IDevice>* – возвращает список открытых устройств.

`DeviceValidator` реализует метод *Validate(string): bool*, который принимает путь к ВУ и проводит проверку ВУ на корректность.

### 3.2.3 Возможные взаимодействия программы с другими программами

Модуль ВУ взаимодействует непосредственно с ВУ, которые, являясь библиотеками, являются программами.

Конкретные ВУ предоставляются пользователем.

### **3.2.4 Описание и обоснование метода организации входных данных**

Входными данными программы являются пути к ВУ и сами ВУ.

Интерфейс ВУ, приведенный в листинге 1, обоснован архитектурой ЭВМ PDP-11: с точки зрения ЭВМ PDP-11 любое внешнее устройство представляет собой пару регистров – буферный и управляющий – соответственно интерфейс предоставляет адреса данных регистров и их значения; также ЭВМ PDP-11 поддерживает прерывания, потому интерфейс предоставляет адрес вектора прерываний, флаг запроса прерывания и метод ответа на прерывания [1, стр. 150-152].

Метод Init() обоснован существованием команды RESET, который позволяет сбрасывать ВУ до состояния на момент подачи питания [1, стр. 286]..

Свойство Name предоставляется для удобства идентификации устройств.

Интерфейс ВУ наследует от IDisposable [2], чтобы предоставить возможность очистки неуправляемых ресурсов, находящихся за пределами среды CLR [3].

### **3.2.5 Описание и обоснование метода организации выходных данных**

Программа в ходе своей работы не создает никаких выходных данных.

### **3.2.6 Описание и обоснование выбора состава технических средств**

Программа, как модуль для программного симулятора, обладает теми же требованиями аппаратной части, как у программного симулятора PDP-11.

### **3.2.7 Описание и обоснование выбора состава программных средств**

Программа, как модуль для программного симулятора, обладает теми же требованиями программной части, как у программного симулятора PDP-11.

## **4 Ожидаемые технико-экономические показатели**

Программа модуля ВУ позволяет открывать и использовать ВУ в программном симуляторе PDP-11 и разрабатывать ВУ с помощью SDK.

## **5 Источники, используемые при разработке**

- 1 Лин В. PDP-11 и VAX-11 Архитектура ЭВМ и программирование на языке ассемблера. М.: «Радио и связь», 1989. – 321 с.
- 2 IDisposable [Электронный ресурс] // Техническая документация Microsoft – URL: <https://learn.microsoft.com/ru-ru/dotnet/api/system.idisposable> (дата обращения: 21.12.2023)
- 3 Неуправляемые ресурсы [Электронный ресурс] // Техническая документация Microsoft – URL: <https://learn.microsoft.com/ru-ru/dotnet/standard/garbage-collection/unmanaged> (дата обращения: 21.12.2023)



<b>Лист регистрации изменений</b>
-----------------------------------

[illegible]

