

Федеральное государственное образовательное бюджетное учреждение
высшего образования
«Финансовый университет при Правительстве Российской Федерации»
Канашский филиал Финуниверситета

Методические рекомендации для студентов
по выполнению самостоятельной работы
по ПМ.02 Сопровождение и техническое обслуживание интеллектуальных
интегрированных систем
по специальности 09.02.08 Интеллектуальные интегрированные системы

2025 г.

Организация-разработчик: ФГОБУ ВО «Финансовый университет при Правительстве Российской Федерации» Канашский филиал Финуниверситета

Разработчик(и):

Николаева И.В.- преподаватель ВКК Канашского филиала Финуниверситета

Рабочая программа дисциплины рассмотрена и рекомендована к утверждению на заседании предметной (цикловой) комиссии интеллектуальных интегрированных систем

Протокол от «29» мар 20 25 г. № 1

Председатель предметно (цикловой) комиссии:  /Славкина А.И./

Пояснительная записка

Методические указания (рекомендации) для студентов по выполнению самостоятельной работы по дисциплине разработаны на основе требований Федерального государственного образовательного стандарта по специальности (специальностям) среднего профессионального образования по специальности 09.02.08 Интеллектуальные интегрированные системы.

Самостоятельная работа по ПМ.02 Сопровождение и техническое обслуживание интеллектуальных интегрированных систем проводится с целью:

- систематизации и закрепления полученных теоретических знаний и практических умений;
- формирования общих и профессиональных компетенций;
- углубления и расширения теоретических знаний;
- формирования умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развития познавательных способностей и активности: творческой инициативы, самостоятельности, ответственности и организованности;
- формирования самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развития исследовательских умений.

Самостоятельная работа по ПМ.02 Сопровождение и техническое обслуживание интеллектуальных интегрированных систем включает задания по выполнению практических заданий, составлению конспектов, решению ситуационных задач и др.

Самостоятельная работа по ПМ.02 Сопровождение и техническое обслуживание интеллектуальных интегрированных систем является внеаудиторной и обязательна для всех студентов. Внеаудиторная самостоятельная работа - планируемая учебная, учебно-исследовательская работа студентов, выполняемая вне занятий по заданию и при управлении преподавателем, но без его непосредственного участия.

Критериями оценки результатов внеаудиторной самостоятельной работы являются:

- уровень освоения учебного материала;
- умения использовать теоретические знания при выполнении практических задач;
- сформированность общеучебных умений;
- обоснованность и четкость изложения ответа;
- оформление материала в соответствии с требованиями.

САМОСТОЯТЕЛЬНАЯ РАБОТА СТУДЕНТА
по ПМ.02 СОПРОВОЖДЕНИЕ И СХЕМОТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ
ИНТЕЛЛЕКТУАЛЬНЫХ ИНТЕГРИРОВАННЫХ СИСТЕМ
СПЕЦИАЛЬНОСТИ 09.02.08 ИНТЕЛЛЕКТУАЛЬНЫЕ ИНТЕГРИРОВАННЫЕ
СИСТЕМЫ

Тема	Кол-во часов	Вид работы
2	3	4
МДК.02.01 Аппаратно-программные интерфейсы микроконтроллерных систем	6	Язык программирования и среда разработки микроконтроллера. Управление микроконтроллером со смартфона через Bluetooth. Взаимодействие с I2C модулем
МДК.02.02 Техническое сопровождение интегрированных систем	10	Память RAM и ROM Система ввода/вывода Компоненты ПК BIOS. Его разновидности Виды технического обслуживания Жесткие диски. Определение. Основные виды ошибок Виды блоков питания Виды сервисного оборудования Локальная сеть. Особенности, принцип построения и работы Основные элементы материнской платы. Особенности, принцип работы Wi-Fi. Особенности, принцип работы Флэш-накопители. Особенности, принцип работы Энергосберегающие технологии Модель OSI Модель TCP/IP Сеть. Виды сетей Принтеры. Виды принтеров. Способы печати

Самостоятельная работа 1

МДК.02.01 Аппаратно-программные интерфейсы микроконтроллерных систем

Задание 1. Язык программирования и среда разработки микроконтроллера.

Описание задачи: Вашей задачей является разработка простого проекта на платформе Arduino, который будет демонстрировать взаимодействие микроконтроллера с внешней средой. Вы будете использовать язык программирования C/C++ в среде Arduino IDE для создания программы, управляющей светодиодами и кнопкой.

Что вам предстоит сделать:

Подготовка оборудования:

Подключите Arduino Uno к компьютеру через USB-кабель.

Подключите светодиод к одному из цифровых пинов Arduino (например, пину 13).

Подключите кнопку к другому цифровому пину (например, пину 12).

Написание программы:

Создайте новую программу в Arduino IDE.

Напишите код, который будет считывать состояние кнопки и управлять светодиодом следующим образом:

Если кнопка нажата, светодиод загорается.

Если кнопка отпущена, светодиод гаснет.

Используйте функции `pinMode()`, `digitalRead()` и `digitalWrite()` для работы с пинами.

Тестирование:

Загрузите программу в микроконтроллер.

Проверьте, что светодиод включается и выключается в зависимости от состояния кнопки.

Расширение функционала:

Добавьте задержку в 1 секунду после нажатия кнопки, чтобы избежать ложных срабатываний.

Реализуйте мигание светодиода с частотой 1 Гц, пока кнопка удерживается.

Документирование:

Подготовьте отчет, в котором опишите архитектуру программы, использованные функции и интерфейсы микроконтроллера.

Приведите примеры работы схемы и реакцию светодиода на нажатия кнопки.

Пример кода на Arduino IDE:

```
const int ledPin = 13; // Пин, к которому подключен светодиод
const int buttonPin = 12; // Пин, к которому подключена кнопка

void setup() {
  pinMode(ledPin, OUTPUT); // Устанавливаем пин светодиода как выход
  pinMode(buttonPin, INPUT_PULLUP); // Устанавливаем пин кнопки как вход с
  подтягивающим резистором
}

void loop() {
  int buttonState = digitalRead(buttonPin); // Считываем состояние кнопки

  if (buttonState == LOW) { // Если кнопка нажата
    digitalWrite(ledPin, HIGH); // Включаем светодиод
  } else { // Если кнопка отпущена
    digitalWrite(ledPin, LOW); // Выключаем светодиод
  }
}
```

Оборудование и материалы:

Микроконтроллер Arduino Uno.

Светодиод.

Кнопка.

Резистор 220 Ом.

Соединительные провода.

USB-кабель для подключения микроконтроллера к компьютеру.

Компьютер с установленной Arduino IDE.

Требуется продемонстрировать умение работать с языком программирования C/C++ в среде Arduino IDE, создавая программу, управляющую светодиодами и кнопкой.

Задание 2. Управление микроконтроллером со смартфона через Bluetooth

Описание задачи: Создайте систему, позволяющую управлять светодиодом, подключенным к микроконтроллеру, с помощью мобильного приложения через Bluetooth. Ваш проект должен включать микроконтроллер (например, Arduino), модуль Bluetooth (например, HC-05) и мобильное приложение (например, для Android или iOS).

Что вам предстоит сделать:

Подготовка оборудования:

Подключите микроконтроллер к компьютеру через USB-кабель.

Подключите светодиод к одному из цифровых пинов микроконтроллера (например, пину 13).

Подключите модуль Bluetooth к микроконтроллеру согласно схеме подключения.

Написание программы для микроконтроллера:

Создайте программу на языке C/C++ в Arduino IDE, которая будет получать команды через Bluetooth и управлять светодиодом.

Используйте функции SoftwareSerial для работы с модулем Bluetooth.

Реализуйте обработку команд, таких как включение и выключение светодиода.

Разработка мобильного приложения:

Создайте простое мобильное приложение, которое сможет отправлять команды через Bluetooth на микроконтроллер.

Используйте библиотеки для работы с Bluetooth (например, BluetoothAdapter и BluetoothSocket для Android).

Реализуйте интерфейс с кнопками для отправки команд включения и выключения светодиода.

Тестирование:

Загрузите программу в микроконтроллер.

Установите мобильное приложение на смартфон.

Проверьте, что светодиод включается и выключается в зависимости от команд, отправляемых через мобильное приложение.

Расширение функционала:

Добавьте возможность регулировки яркости светодиода через мобильное приложение.

Реализуйте обратную связь, чтобы мобильное приложение могло получать статус светодиода от микроконтроллера.

Документирование:

Подготовьте отчет, в котором опишите архитектуру программы, использованные функции и интерфейсы микроконтроллера.

Приведите примеры работы схемы и реакцию светодиода на команды из мобильного приложения.

Пример кода для Arduino IDE:

```
#include <SoftwareSerial.h>

const int ledPin = 13; // Пин, к которому подключен светодиод
const int rxPin = 10; // Пин RX модуля Bluetooth
const int txPin = 11; // Пин TX модуля Bluetooth

SoftwareSerial bluetooth(rxPin, txPin); // Создаем объект SoftwareSerial

void setup() {
  pinMode(ledPin, OUTPUT); // Устанавливаем пин светодиода как выход
  bluetooth.begin(9600); // Начинаем общение через Bluetooth
}

void loop() {
  if (bluetooth.available()) { // Если есть данные от Bluetooth
    char command = bluetooth.read(); // Считываем команду

    if (command == '1') { // Команда включения светодиода
      digitalWrite(ledPin, HIGH); // Включаем светодиод
    } else if (command == '0') { // Команда выключения светодиода
      digitalWrite(ledPin, LOW); // Выключаем светодиод
    }
  }
}
```

Оборудование и материалы:
Микроконтроллер Arduino Uno.
Модуль Bluetooth HC-05.

Светодиод.

Резистор 220 Ом.

Соединительные провода.

USB-кабель для подключения микроконтроллера к компьютеру.

Смартфон с поддержкой Bluetooth.

Требуется продемонстрировать умение работать с Bluetooth-модулем и мобильным приложением, создавая систему дистанционного управления микроконтроллером.

Задание 3. Взаимодействие с I2C модулем

Описание задачи: Вам необходимо подключить микроконтроллер (например, Arduino Uno) к I2C-датчику температуры DS3231 и написать программу, которая будет считывать текущую температуру каждые 5 секунд и выводить её на последовательный порт.

Что вам предстоит сделать:

Подготовка оборудования:

Подключите микроконтроллер к компьютеру через USB-кабель.

Подключите I2C-датчик DS3231 к микроконтроллеру, используя линии SDA и SCL. Убедитесь, что питание датчика соответствует напряжению микроконтроллера (обычно 3.3V или 5V).

Написание программы:

Создайте новую программу в Arduino IDE.

Импортируйте библиотеку Wire для работы с I2C.

Используйте

функции Wire.begin(), Wire.requestFrom(), Wire.write() и Wire.read() для взаимодействия с датчиком.

Реализуйте чтение данных с датчика DS3231 и преобразование их в человечески понятную форму (температуру в градусах Цельсия).

Выведите результат на последовательный порт с интервалом в 5 секунд.

Тестирование:

Загрузите программу в микроконтроллер.

Откройте монитор последовательного порта (например, в Arduino IDE) и убедитесь, что температура выводится корректно.

Проверьте стабильность работы схемы при изменении температуры окружающей среды.

Расширение функционала:

Добавьте возможность регулировки интервала измерения температуры через последовательный порт.

Реализуйте возможность сохранения данных измерений в энергонезависимую память (EEPROM).

Документирование:

Подготовьте отчет, в котором опишите архитектуру программы, использованные функции и интерфейсы микроконтроллера.

Приведите примеры работы схемы и реакцию системы на изменение температуры.

Пример кода на Arduino IDE:

```
#include <Wire.h> // Библиотека для работы с I2C

// Адрес I2C для DS3231
#define DS3231_I2C_ADDRESS 0x68
```

```

void setup() {
  Serial.begin(9600); // Инициализация последовательного порта
  Wire.begin(); // Инициализация I2C
}

void loop() {
  byte temp_msb, temp_lsb; // Переменные для хранения старших и младших байтов
  температуры

  Wire.beginTransmission(DS3231_I2C_ADDRESS); // Начало передачи данных
  Wire.write(0x11); // Адрес регистра температуры
  Wire.endTransmission(false); // Завершаем передачу, сохраняя возможность чтения

  Wire.requestFrom(DS3231_I2C_ADDRESS, 2); // Запрашиваем 2 байта данных
  if (Wire.available() >= 2) {
    temp_msb = Wire.read(); // Чтение старшего байта
    temp_lsb = Wire.read(); // Чтение младшего байта
  }

  // Преобразование температуры в градусы Цельсия
  float temperature = ((temp_msb & 0x7F) << 1) | (temp_lsb >> 7);
  if (temp_msb & 0x80) temperature *= -1; // Коррекция знака температуры

  Serial.print("Температура: "); // Вывод температуры на последовательный порт
  Serial.print(temperature);
  Serial.println(" °C");

  delay(5000); // Интервал измерения 5 секунд
}

```

Оборудование и материалы:

Микроконтроллер Arduino Uno.

I2C-датчик температуры DS3231.

Соединительные провода.

USB-кабель для подключения микроконтроллера к компьютеру.

Компьютер с установленной Arduino IDE.

Требуется продемонстрировать умение работать с I2C-протоколом, подключаться к внешнему устройству и обрабатывать данные.

Самостоятельная работа 2

МДК.02.02 Техническое сопровождение интегрированных систем

Задание 1. Создать презентацию по следующим темам

Память RAM и ROM

Система ввода/вывода

Компоненты ПК

BIOS. Его разновидности

Виды технического обслуживания

Жесткие диски. Определение.

Основные виды ошибок

Виды блоков питания

Виды сервисного оборудования

Локальная сеть. Особенности, принцип построения и работы

Основные элементы материнской платы. Особенности, принцип работы

Wi-Fi. Особенности, принцип работы

Флэш-накопители. Особенности, принцип работы
Энергосберегающие технологии
Модель OSI
Модель TCP/IP
Сеть. Виды сетей
Принтеры. Виды принтеров. Способы печати

Список использованной литературы

1. Назаров, А. В. Эксплуатация объектов сетевой инфраструктуры: учебник / А.В. Назаров, А.Н. Енгальчев, В.П. Мельников. — Москва: КУРС: ИНФРА-М, 2022. — 360 с. — (Среднее профессиональное образование). - ISBN 978-5-906923-06-6. - Текст: электронный. - URL: <https://znanium.com/catalog/product/186018>
2. Сажнев, А. М. Микропроцессорные системы: цифровые устройства и микропроцессоры: учебное пособие для среднего профессионального образования / А. М. Сажнев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 139 с. — (Профессиональное образование). — ISBN 978-5-534-12092-9. — Текст: электронный // Образовательная платформа Юрайт.]. — URL: <https://urait.ru/bcode/518734>
3. Назаров, А. В. Эксплуатация объектов сетевой инфраструктуры: учебник / А.В. Назаров, А.Н. Енгальчев, В.П. Мельников. — Москва : КУРС : ИНФРА-М, 2022. — 360 с. — (Среднее профессиональное образование). - ISBN 978-5-906923-06-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1860128>.