

**Методические рекомендации по выполнению курсового
проекта по дисциплине «Операционные системы»**

Оглавление

Введение	3
Порядок и сроки выполнения и сдачи курсового проекта	3
Требования к оформлению курсового проекта	4
Общие требования к работе встраиваемой системы	4
Варианты курсовых проектов	5
Вариант 1.	5
Вариант 2.	9
Вариант 3.	13
Вариант 4.	17
Вариант 5.	21
Вариант 6.	25
Вариант 7.	29
Вариант 8.	32
Вариант 9.	36
Правила написания исходных кодов курсового проекта на Си.....	40
Критерии оценки курсового проекта.....	42
Список рекомендуемой литературы.....	45

Введение

Студент бакалавриата – будущий инженер, должен освоить компетенции по самостоятельному поиску информации и материалов, позволяющих решить поставленную перед ним практическую задачу. Поэтому данные рекомендации не содержат прямых указаний по написанию исходных кодов и настройки отдельных компонент разрабатываем встраиваемой системы, но даются ссылки на литературные источники, спецификации модулей ВС и примеры работы с драйверами и периферийными устройствами raspberry pi 4.

Каждый вариант курсового проекта – это задание на разработку программного обеспечения встраиваемых систем.

Порядок и сроки выполнения и сдачи курсового проекта

На 3-м лабораторном занятии преподаватель раздает варианты заданий курсовых проектов (приведены в ниже).

В процессе выполнения курсового проекта преподаватель помогает студентам, отвечая на их вопросы. Прежде, чем обратиться за помощью преподавателя, рекомендуется предварительно сформировать собственное мнение по интересующему вопросу, и, при необходимости, корректировать его, выслушав советы преподавателя.

Курсовой проект выполняется студентами самостоятельно во время проведения лабораторных работ после сдачи всех заданий лабораторного практикума и во время самостоятельной работы студента. При сдаче курсового проекта студентом демонстрируется работа прототипа встраиваемой системы, разработанной в рамках выполнения курсового проекта, описываются исходные коды программ и скриптов, даются ответы на вопросы преподавателя о принципах функционирования механизмов используемой операционной системы и периферийных устройств встраиваемой системы.

Сроки выполнения курсового проекта 3-8 лабораторное занятие.

Срок сдачи и защиты курсового проекта 7-8 лабораторное занятие.

При согласовании с преподавателем, курсовые проекты могут выполняться командами студентов из 2-3 человек. Рекомендуется распределить задачи между членами команды и вести git репозиторий совместно. Однако, при защите курсовых проектов каждый из команды должен иметь представление об функционировании встраиваемой системы и исходных кодов проекта, которые были написаны другими членами команды. Преподаватель в праве задавать вопросы и дополнительные задания при сдаче курсового проекта. Критерии оценки приведены в соответствующем разделе данных методических рекомендаций ниже.

Требования к оформлению курсового проекта

1. Каждое выполненное задание или часть курсового проекта должна быть сохранена с использованием системы контроля версий git. Допускается сохранение промежуточных вариантов.
2. Коммиты в git должны иметь содержательное описание произведенных изменений.
3. Для каждого выполненного задания или части курсового проекта должны быть созданы файлы «README.MD», располагающийся в одном каталоге с исходными кодами и сохраненный с помощью git. Данные файлы должны описывать поведение работы скрипта и показывать пример запуска.
4. Итоговая версия курсового проекта должна сопровождаться «README.MD» файлом, описывающим принцип функционирования встраиваемой системы, способы подключения к ней через терминал putty, список и формат команд управления, и формат отображения результатов работы в браузере.

Общие требования к работе встраиваемой системы

1. Встраиваемая система автоматически запускает все приложения, обеспечивающие выполнение всех задач варианта задания, после загрузки операционной системы RPi.
2. Встраиваемая система работает все время при наличии питания и останавливается только при его отключении или при получении команды на прекращение работы.
3. Встраиваемая система обеспечивает прием и исполнение следующих команд управления по стандартному потоку ввода/вывода (формат команд может определяться студентом самостоятельно):
 - остановка работы всех приложений;
 - установка скорости опроса датчиков и т.д.

Варианты курсовых проектов

Вариант 1.

1. Описание задания:

Необходимо разработать прототип встраиваемой системы, как совокупность программного обеспечения и стенда на базе Raspberry Pi4, которая позволяла бы определять расстояние до грозы по принципу измерения транзитного времени – разница между вспышкой (имитируется фонарем) и звуковым сигналом (имитируется хлопком). Встраиваемая система должна обеспечивать измерение расстояний для серии из нескольких вспышек и последующих за ними звуковыми сигналами. Результат измерения расстояния (или серии измеренных расстояний) должны отправляться на web-сервер и отображаться в окне браузера, а также дублироваться по ssh-соединению на персональный компьютер.

2. Используемые отладочные комплекты:

- микрокомпьютер Raspberry Pi 4;
- макетная плата Freescale PBMCUSLK;
- модуль АЦП ADS1115 16-Bit;
- датчик звука (шума) с микрофоном KY-038;
- фоторезистор GL5516;
- резистор 100 кОм;
- персональный компьютер.

3. Требования к работе встраиваемой системы:

3.1 Встраиваемая система должна обеспечивать вычисление расстояния до грозы за счет выполнения трех основных программ: 1 – программа, принимающая сигналы с фоторезистора, 2 – программа, принимающая сигналы с датчика звука, 3 – программа, получающая данные от программ 1 и 2 по одному из способов межпроцессного взаимодействия (именованные и неименованные каналы).

3.2 Программа-1 должна обеспечивать взаимодействие RPi 4 с модулем АЦП ADS1115 по интерфейсу I2C и выполняет следующие функции:

- настройка режима работы АЦП: тип подключения - несимметричный/дифференциальный, скорость обработки данных;
- запись и чтение значений регистров АЦП;
- изменение скорости опроса АЦП по аргументу при старте или через команду по именованным/не именованным каналам;
- преобразование данных АЦП в значение освещенности;
- установка уровня освещенности для распознавания вспышки по аргументу при старте или через команду по именованным/не именованным каналам;

- отправка сигнала о вспышке с временной меткой системы по именованным/не именованным каналам.

3.3 Программа-2 должна обеспечивать взаимодействие RPi с датчиком звука по GPIO и выполняет следующие функции:

- чтение значений GPIO о срабатывании датчика звука;
- изменение скорости опроса датчика звука по аргументу при старте или через команду по именованным/не именованным каналам;
- отправка сигнала о хлопке с временной меткой системы по именованным/не именованным каналам.

3.4 Программа-3 должна обеспечивать взаимодействие с программами-1,-2, web-сервером и стандартным вводом, работает в многопоточном режиме, а также выполняет следующие функции:

- обмен сообщениям по именованным/не именованным каналам с приложениями-1,-2;
- обмен сообщениями со стандартным потоком ввода/вывода, в том числе прием команд от пользователя (см. общие требования к работе ВС);
- выдача результатов на web-сервер;

3.5. Встраиваемая система должна обеспечивать выдачу результатов измерения дальности в следующем формате: «время измерения» (hh:mm:ss), «расстояние до грозы № x» в метрах на web-сервер.

4. Общая схема и принципы работы встраиваемой системы:

Схема соединений всего используемого оборудования представлена на рисунке 1. Для соединения проводников в нескольких точках используйте безопасную макетную плату, установленную в Freescale PBMUSLK и набор перемычек. В качестве датчика измерения вспышки используется фоторезистор GL5516, который подключен ко входу аналого-цифрового преобразователя ADS1115. Данный АЦП обменивается сообщениями с микрокомпьютером Raspberry Pi 4 посредством интерфейса I2C. Для измерения хлопков используется датчик звука KY-038, цифровой выход которого подключается на вход GPIO RPi4.

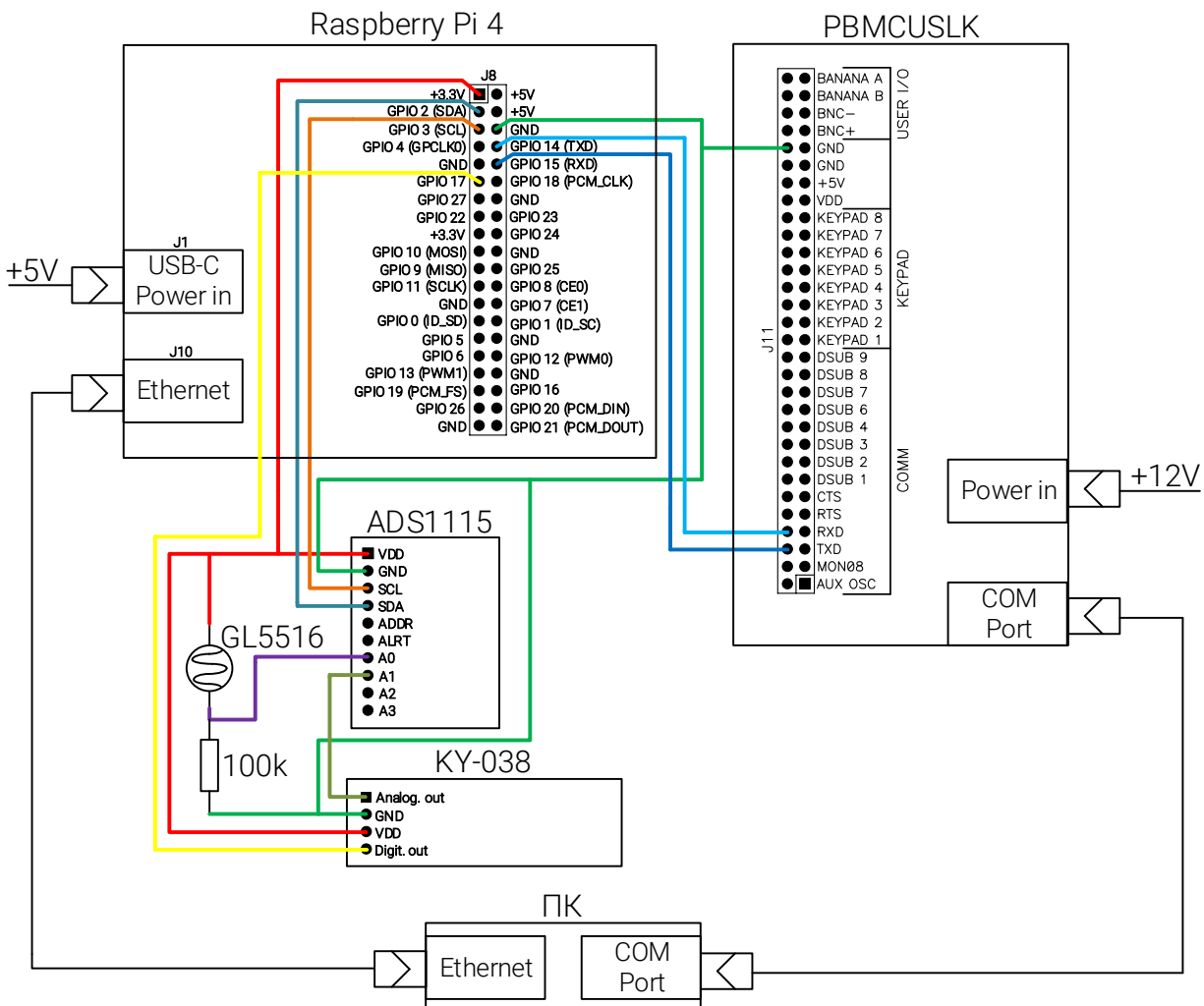


Рисунок 1. Схема соединений по варианту 1

5. Рекомендуемый порядок выполнения курсового проекта:

Выполнение курсового проекта рекомендуется делать поэтапно, разобравшись первоначально с работой периферии и интерфейсов, далее реализовать отдельные программы работы с периферией, далее модифицировать их согласно требованиям к функционированию встраиваемой системы.

Выполнение курсового проекта может осуществляться в рамках лабораторных занятий по дисциплине «Операционные системы», при этом, с разрешения преподавателя, можно заменить индивидуальные задания лабораторной работы на следующие пункты курсового проекта, представленные в таблице 1:

Таблица 1. Задания лабораторных работ из курсового проекта по варианту 1

Лабораторная работа	Сдача этапа курсового проекта
«Процессы и межпроцессное взаимодействие»	<p>Задание №3.1. Разработайте программу-1, которая будет записывать в файл время срабатывания хлопка. Путь к файлу передается в качестве входного аргумента. Разработайте программу-2, которая будет считывать значения по интерфейсу i2c с АЦП, к которому подключен фоторезистор. Определите экспериментально значение АЦП по уровню срабатывания фоторезистора на вспышку фонаря и выводите значение в стандартный поток ввода-вывода.</p> <p>Задание №3.2. Модернизируйте программы, чтобы данные о времени срабатывания датчиков направлялись по именованным каналам. Напишите третью программу, которая бы последовательно принимала сообщения по именованным каналам.</p> <p>Задание №3.3. Усовершенствуйте первые две программы, добавив в нее обработчик сигнала SIGINT, по которому в третью программу будет отправлять сообщение об окончании работы первых двух программ.</p>
«Разработка многопоточных программ»	<p>Задание №4.1. Разработайте программу, которая в двух независимых потоках будет принимать сообщения по именованным каналам от программ, взаимодействующим с датчиками звука и освещенности.</p> <p>Задание №4.2. Модернизируйте программу, добавив еще один поток вычисления расстояния как разница времени между вспышкой и хлопком умноженная на константное значение скорости звука.</p> <p>Задание №4.3. Добавьте функцию отображения результатов измерения расстояния в web-интерфейсе.</p>

6. Вспомогательные материалы:

Спецификация на используемые модули:

- [микрокомпьютер Raspberry Pi 4;](#)
- макетная плата Freescale PBMCSLК: [схема](#), [описание](#), [быстрый старт](#);
- [модуль АЦП ADS1115 16-Bit;](#)
- [датчик звука \(шума\) с микрофоном KY-038;](#)
- [фоторезистор GL5516;](#)

Описание, настройка и использование периферийных устройств Raspberry Pi 4 доступно по следующим ссылкам:

- [основы работы с GPIO в Raspberry Pi OS;](#)
- [How to Control GPIO Hardware from C or C++;](#)
- [пример исходных кодов для ADS1115;](#)
- [использование АЦП ADS1115;](#)
- [исходные коды для ADS1115.](#)

Вариант 2.

1. Описание задания:

Необходимо разработать прототип встраиваемой системы, как совокупность программного обеспечения и стенда на базе Raspberry Pi4, которая позволяла бы определять с помощью двух ультразвуковых датчиков расстояние до препятствия и выводить результат на LCD-экран. Датчики могут быть с различными интерфейсами (uart, GPIO). Результат измерения расстояния (или серии измеренных расстояний) должны отправляться на web-сервер и отображаться в окне браузера, а также дублироваться по ssh-соединению на персональный компьютер.

2. Используемые отладочные комплекты:

- микрокомпьютер Raspberry Pi 4;
- макетная плата Freescale PBMCUSLK;
- ультразвуковой датчик HC-SR04;
- ультразвуковой датчик Pmod MAXSONAR;
- персональный компьютер.

3. Требования к работе встраиваемой системы:

3.1 Встраиваемая система должна обеспечивать измерения расстояния и вывод его на экран за счет выполнения трех основных программ: 1 – программа, принимающая сигналы с датчика №1, 2 – программа, принимающая сигналы с датчика №3, 3 – программа, получающая данные от программ 1 и 2 по одному из способов межпроцессного взаимодействия (именованные и неименованные каналы) и выводящее измеряемые расстояния на LCD-экран.

3.2 Программы-1 и -2 должны обеспечивать взаимодействие RPi 4 с модулями ультразвуковых датчиков по интерфейсу uart или GPIO, работают в многопоточном режиме и выполняют следующие функции:

- настройка режима работы датчика и его тип подключения (uart, gpio) по аргументу при старте или через команду по именovanным/не именovanным каналам;
- чтение значений расстояния по uart или измерение расстояния через GPIO в метрах;
- выдача данных о расстоянии с временной меткой системы по именovanным/не именovanным каналам;
- прием команд остановки и старта работы по именovanным/не именovanным каналам;

3.3 Программа-3 должна обеспечивать взаимодействие с программами-1,-2, web-сервером и стандартным вводом, выводит информацию на LCD-экран по SPI, работает в многопоточном режиме, а также выполняет следующие функции:

- обмен сообщениям по именованным/не именованным каналам с приложениями-1,-2;
- конфигурирует работу SPI интерфейса по аргументам при старте или через команду по именованным/не именованным каналам;
- вывод информации на LCD-экран по SPI;
- обмен сообщениями со стандартным потоком ввода/вывода, в том числе прием команд от пользователя (см. общие требования к работе ВС);
- выдача результатов на web-сервер;

3.5. Встраиваемая система обеспечивает выдачу результатов измерения цвета на web-server в следующем формате: «время измерения» (hh:mm:ss), «цвет измеренный - ...», «цвет выдаваемый – ...».

4. Общая схема и принципы работы встраиваемой системы:

Схема соединений всего используемого оборудования представлена на рисунке 2. Для соединения проводников в нескольких точках используйте безопасную макетную плату, установленную в Freescale PBMCUSLK и набор перемычек.

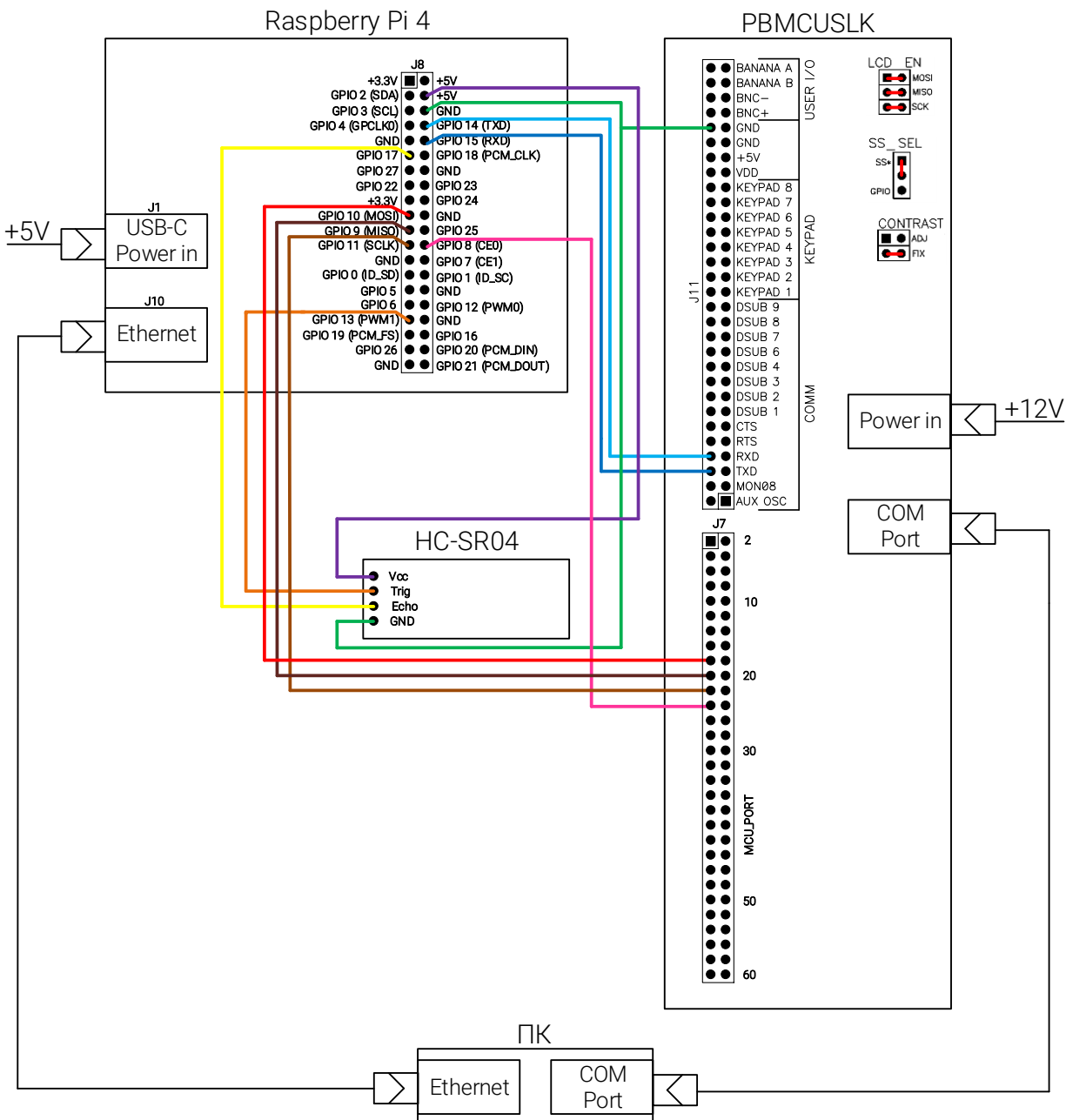


Рисунок 2. Схема соединений по варианту 2

5. Рекомендуемый порядок выполнения курсового проекта:

Выполнение курсового проекта рекомендуется делать поэтапно, разобравшись первоначально с работой периферии и интерфейсов, далее реализовать отдельные программы работы с периферией, далее модифицировать их согласно требованиям к функционированию встраиваемой системы.

Выполнение курсового проекта может осуществляться в рамках лабораторных занятий по дисциплине «Операционные системы», при этом, с

разрешения преподавателя, можно заменить индивидуальные задания лабораторной работы на следующие пункты курсового проекта, представленные в таблице 2:

Таблица 2. Задания лабораторных работ из курсового проекта по варианту 2

Лабораторная работа	Сдача этапа курсового проекта
«Процессы и межпроцессное взаимодействие»	<p>Задание №3.1. Разработайте программу, которая будет записывать в файл время и измеренное расстояние. Пусть к файлу, тип интерфейса дальномера и период записи результата передается в качестве входного аргумента. Разработайте еще одну программу, которая будет выводить значения и графику по интерфейсу SPI на LCD-экран.</p> <p>Задание №3.2. Модернизируйте программы, чтобы данные о времени срабатывания датчиков направлялись по именованным каналам. Напишите третью программу, которая бы последовательно принимала сообщения по именованным каналам.</p> <p>Задание №3.3. Усовершенствуйте первые две программы, добавив в нее обработчик сигнала SIGINT, по которому в третью программу будет отправлять сообщение об окончании работы первых двух программ.</p>
«Разработка многопоточных программ»	<p>Задание №4.1. Разработайте программу, которая в двух независимых потоках будет принимать сообщения по именованным каналам от программ, взаимодействующим с дальномерами, и будет выводить данные на LCD-экран в режиме раздельного использования измерений.</p> <p>Задание №4.2. Модернизируйте программу, добавив еще один поток вычисления разницы расстояний по показаниям дальномеров и реализации режима совместного измерения.</p> <p>Задание №4.3. Добавьте функцию отображения результатов измерения расстояния в web-интерфейсе и в стандартный поток ввода/вывода.</p>

6. Вспомогательные материалы:

Спецификация на используемые модули:

- [микрокомпьютер Raspberry Pi 4](#);
- макетная плата Freescale PBMUSLK: [схема](#), [описание](#), [быстрый старт](#);
- [ультразвуковой дальномер HC-SR04](#);
- [ультразвуковой дальномер Pmod MAXSONAR](#);

Описание, настройка и использование периферийных устройств Raspberry Pi 4 доступно по следующим ссылкам:

- [Основы работы с GPIO в Raspberry Pi OS](#);
- [How to Control GPIO Hardware from C or C++](#);
- [исходные коды по настройке и использованию интерфейса SPI](#);
- [исходные коды для HC-SR04](#);
- [исходные коды для LCD-дисплея](#).

Вариант 3.

1. Описание задания:

Необходимо разработать прототип встраиваемой системы, как совокупность программного обеспечения и стенда на базе Raspberry Pi4, которая позволяла бы управлять переключением светофора по расписанию (три светодиода), а также считывать цвет с помощью соответствующего сенсора (Color Sensor Module PN 410-348). Результаты распознанных цветов должны отправляться на web-сервер и отображаться в окне браузера, а также дублироваться по ssh-соединению на персональный компьютер.

2. Используемые отладочные комплекты:

- микрокомпьютер Raspberry Pi 4;
- макетная плата Freescale PBMCSLK;
- светодиоды (красный, желтый, зеленый) и резисторы;
- Color Sensor Module PN 410-348;
- персональный компьютер.

3. Требования к работе встраиваемой системы:

3.1 Встраиваемая система должна обеспечивать управление тремя светодиодами в режиме светофора и определение свечения цвета, вывод данной информации за счет выполнения трех основных программ: 1 – программа, управляющая светодиодами и опросом кнопок отладочного стенда, 2 – программа, принимающая сигналы с датчика цвета по I2C интерфейсу, 3 – программа, получающая данные от программ 1 и 2 по одному из способов межпроцессного взаимодействия (именованные и неименованные каналы) и выводящее результат работы светофора и определенного цвета по последовательному каналу и на web-сервер.

3.2 Программа-1 должна обеспечивать взаимодействие RPi 4 со светодиодами и кнопками с помощью GPIO, работает в многопоточном режиме и выполняют следующие функции:

- настройка режима работы светофора (времена переключения) по аргументу при старте или через команду по именованным/не именованным каналам;
- чтение значений кнопки для ускорения переключения;
- выдача данных о светящемся светодиоде с временной меткой системы;
- прием команд остановки и старта работы по именованным/не именованным каналам;

3.3 Программа-2 должна обеспечивать взаимодействие RPi 4 с датчиком цвета по интерфейсу I2C, работает в многопоточном режиме и выполняют следующие функции:

- запись и чтение значений регистров датчика цвета;
- преобразование данных датчика в значение цвета из трех возможных;
- отправка сигнала о смене цвета или отсутствии данных с временной меткой системы по именованным/не именованным каналам.

3.4 Программа-3 должна обеспечивать взаимодействие с программами-1,-2, web-сервером и стандартным вводом, работает в многопоточном режиме, а также выполняет следующие функции:

- обмен сообщениям по именованным/не именованным каналам с приложениями-1,-2;
- обмен сообщениями со стандартным потоком ввода/вывода, в том числе прием команд от пользователя (см. общие требования к работе ВС);
- выдача результатов на web-сервер;

3.5 Встраиваемая система обеспечивает выдачу результатов измерения цветов и режима работы светофора на web-server и в следующем формате: «время измерения» (hh:mm:ss), «режим работы светофора - ... », «измеренный цвет - ...».

4. Общая схема и принципы работы встраиваемой системы:

Схема соединений всего используемого оборудования представлена на рисунке 3. Для соединения проводников в нескольких точках используйте безопасную макетную плату, установленную в Freescale PBMUSLK и набор перемычек.

разрешения преподавателя, можно заменить индивидуальные задания лабораторной работы на следующие пункты курсового проекта, представленные в таблице 3:

Таблица 3. Задания лабораторных работ из курсового проекта по варианту 3

Лабораторная работа	Сдача этапа курсового проекта
«Процессы и межпроцессное взаимодействие»	<p>Задание №3.1. Разработайте программу, которая будет записывать в файл время и режим работы светофора, который может переключаться как по таймауту, так и по нажатию на кнопку. Пусть к файлу и номер кнопки для переключения режима передается в качестве входного аргумента. Разработайте еще одну программу, которая будет выводить значения цвета в стандартный поток ввода/вывода с датчика цвета, подключенного по интерфейсу i2c.</p> <p>Задание №3.2. Модернизируйте программы, чтобы данные о времени изменения режима работы светофора и оцениваемого цвета направлялись по неименованным каналам. Напишите третью программу, которая бы последовательно принимала сообщения по неименованным каналам.</p> <p>Задание №3.3. Усовершенствуйте первые две программы, добавив в нее обработчик сигнала SIGINT, по которому в третью программу будет отправлять сообщение об окончании работы первых двух программ.</p>
«Разработка многопоточных программ»	<p>Задание №4.1. Разработайте программу, которая в двух независимых потоках будет принимать сообщения по неименованным каналам от программ -1 и -2 и выводить результат в консоль.</p> <p>Задание №4.2. Модернизируйте программу, добавив еще один поток вычисления времени срабатывания датчика цвета как разницу между временами прихода данных от двух программ -1 и -2.</p> <p>Задание №4.3. Добавьте функцию отображения результатов в web-интерфейсе и в стандартный поток ввода/вывода.</p>

6. Вспомогательные материалы:

Спецификация на используемые модули:

- [микрокомпьютер Raspberry Pi 4](#);
- макетная плата Freescale PBMCSLК: [схема](#), [описание](#), [быстрый старт](#);
- набор светодиодов и резисторов;
- [AMS TCS3472](#).

Описание, настройка и использование периферийных устройств Raspberry Pi 4 доступно по следующим ссылкам:

- [Основы работы с GPIO в Raspberry Pi OS](#);
- [How to Control GPIO Hardware from C or C++](#);
- [исходные коды по настройке и использованию интерфейса SPI](#);
- [Pmod COLOR Reference Manual](#);
- [Pmod COLOR](#);
- [подключение pmods по интерфейсу I2C](#).

Вариант 4.

1. Описание задания:

Необходимо разработать прототип встраиваемой системы, как совокупность программного обеспечения и стенда на базе Raspberry Pi4, которая бы выводила на OLED-дисплей время в виде циферблата часов с часовой, минутной и секундной стрелками, а также позволяла настраивать время с помощью энкодера. Время и операция его изменения энкодером должны отправляться на web-сервер и отображаться в окне браузера, а также дублироваться по ssh-соединению на персональный компьютер.

2. Используемые отладочные комплекты:

- микрокомпьютер Raspberry Pi 4;
- макетная плата Freescale PBMCSLK;
- модуль Pmod ENC: Rotary Encoder PN 410-117;
- модуль Pmod OLEDrgb: 96 x 64 RGB OLED Display;
- персональный компьютер.

3. Требования к работе встраиваемой системы:

3.1 Встраиваемая система должна обеспечивать вывод времени за счет выполнения трех основных программ: 1 – программа, работающая с OLED-дисплеем и выводящая время в виде циферблата часов, 2 – программа, принимающая сигналы с энкодера и определяющая угол поворота секундной стрелки, 3 – программа, по нажатию на кнопку передающая значения углов поворота энкодера в программу -2 по одному из способов межпроцессного взаимодействия (именованные и неименованные каналы).

3.2 Программа-1 должна обеспечивать взаимодействие RPi 4 с модулем OLED-Display по интерфейсу SPI и выполняет следующие функции:

- Настройка и инициализация OLED-дисплея;
- запись и чтение значений регистров OLED-дисплея;
- отрисовка циферблата часов по текущему времени;
- изменение времени по данным угла поворота секундной стрелки;

3.3 Программа-2 должна обеспечивать взаимодействие RPi с датчиком угла поворота (энкодером) по GPIO и выполняет следующие функции:

- чтение значений GPIO и вычисление угла поворота;
- отправка значения угла поворота с временной меткой системы по именованным/не именованным каналам.

3.4 Программа-3 должна обеспечивать взаимодействие с программами-1,-2, web-сервером и стандартным вводом, работает в многопоточном режиме, а также выполняет следующие функции:

- обработка нажатия кнопки на изменение времени;
- обмен сообщениям по именованным/не именованным каналам с приложениями-1,-2;
- обмен сообщениями со стандартным потоком ввода/вывода, в том числе прием команд от пользователя (см. общие требования к работе ВС);
- выдача результатов на web-сервер;

3.5. Встраиваемая система обеспечивает выдачу событий изменения и самого времени на web-server в следующем формате: «время» (hh:mm:ss), «сигнал на изменение времени».

4. Общая схема и принципы работы встраиваемой системы:

Схема соединений всего используемого оборудования представлена на рисунке 4. Для соединения проводников в нескольких точках используйте безопасную макетную плату, установленную в Freescale P BMCUSLK и набор перемычек.

Таблица 4. Задания лабораторных работ из курсового проекта по варианту 4

Лабораторная работа	Сдача этапа курсового проекта
«Процессы и межпроцессное взаимодействие»	<p>Задание №3.1. Разработайте программу, которая будет обрабатывать данные датчика угла поворота по GPIO интерфейсу и вычислять угла поворота, записывать его в файл вместе со временем с шагом в 1 секунду. Пусть к файлу передается в качестве входного аргумента. Разработайте еще одну программу, которая будет отрисовывать циферблат часов на OLED-дисплее по интерфейсу SPI.</p> <p>Задание №3.2. Модернизируйте программу-2, чтобы данные об угле поворота направлялись по неименованным каналам. Напишите третью программу, которая бы последовательно принимала сообщения по неименованным каналам.</p> <p>Задание №3.3. Усовершенствуйте первые две программы, добавив в нее обработчик сигнала SIGINT, по которому в третью программу будет отправлять сообщение об окончании работы первых двух программ.</p>
«Разработка многопоточных программ»	<p>Задание №4.1. Разработайте программу, которая в двух независимых потоках будет принимать сообщения по именованным каналам от программ-1, -2.</p> <p>Задание №4.2. Модернизируйте программу, добавив еще один поток обработки нажатия кнопки для отправки сообщения программе-1 об изменении времени по значениям поворота стрелки циферблата.</p> <p>Задание №4.3. Добавьте функцию отображения результатов работы встраиваемой системы в web-интерфейс.</p>

6. Вспомогательные материалы:

Спецификация на используемые модули:

- [микрокомпьютер Raspberry Pi 4](#);
- макетная плата Freescale PBMCUSLK: [схема](#), [описание](#), [быстрый старт](#);
- [модуль Pmod ENC: Rotary Encoder PN 410-117](#);
- [модуль Pmod OLEDrgb: 96 x 64 RGB OLED Display](#);

Описание, настройка и использование периферийных устройств Raspberry Pi 4 доступно по следующим ссылкам:

- [How to Control GPIO Hardware from C or C++](#);
- [примеры исходных кодов для Pmod ENC от digilent](#);
- [основы работы с GPIO в Raspberry Pi OS](#);
- [исходные коды для работы с OLED-дисплеем](#);
- [исходные коды для работы с Pmod ENC](#).

Вариант 5.

1. Описание задания:

Необходимо разработать прототип встраиваемой системы, как совокупность программного обеспечения и стенда на базе Raspberry Pi4, которая управляла бы шаговым двигателем, изменяя угол поворота по данным от гироскопа, вращающегося в одной плоскости. Встраиваемая система должна обеспечивать изменение угла поворота в ручном режиме по нажатию на кнопки «влево», «вправо». Все действия должны записываться в лог-файл и отправляться на web-сервер, отображаться в окне браузера, а также дублироваться по ssh-соединению на персональный компьютер.

2. Используемые отладочные комплекты:

- микрокомпьютер Raspberry Pi 4;
- макетная плата Freescale PBMCUCLK;
- модуль 3-х осевого гироскопа на L3G4200D;
- шаговый двигатель 28BYJ-48;
- драйвер электродвигателя ULN2003;
- персональный компьютер.

3. Требования к работе встраиваемой системы:

3.1 Встраиваемая система должна управлять шаговым двигателем за счет выполнения трех основных программ: 1 – программа, работающая с драйвером электродвигателя, управляя его движением, а также взаимодействующая с web-сервером и консолью, 2 – программа, принимающая сигналы с гироскопа и определяющая угол поворота электродвигателя, 3 – программа, по нажатию на кнопку передающая значения инкремент/декремент угла поворота шагового двигателя в программу -1 по одному из способов межпроцессного взаимодействия (именованные и неименованные каналы).

3.2 Программа-1 должна обеспечивать взаимодействие RPi 4 с драйвером электродвигателя ULN2003 с помощью GPIO, обеспечивает взаимодействие с программами-2,-3, web-сервером и стандартным вводом, работает в многопоточном режиме и выполняет следующие функции:

- настройка и инициализация GPIO для взаимодействия с драйвером;
- чтение значения угла поворота двигателя и направления;
- обмен сообщениям по именованным/не именованным каналам с приложениями-2,-3;
- обмен сообщениями со стандартным потоком ввода/вывода, в том числе прием команд от пользователя (см. общие требования к работе ВС);

- выдача результатов на web-сервер.

3.3 Программа-2 должна обеспечивать взаимодействие RPi с гироскопом L3G4200D по интерфейсу I2C и выполняет следующие функции:

- инициализирует и настраивает интерфейс I2C;
- считывает значения угла поворота в одной из плоскостей вращения модуля L3G4200D с частотой 1кГц;
- отправка значения приращения угла поворота с временной меткой системы по именованным/не именованным каналам с частотой 10 Гц.

3.4 Программа-3 должна обеспечивать считывание данных с кнопок отладочной платы Freescale для изменения угла поворота шагового двигателя, при этом выполняются следующие функции:

- обработка нажатия кнопки «влево» или «вправо» с защитой отдребезга в 1 секунду;
- передаваемое через аргумент значение угла поворота по нажатию кнопки;
- обмен сообщениями со стандартным потоком ввода/вывода для изменения стандартного угла поворота по нажатию на кнопку, а также прием команд от пользователя (см. общие требования к работе ВС);
- обмен сообщениям по именованным/не именованным каналам с приложением-1.

3.5. Встраиваемая система должна обеспечивать выдачу измеренных значений угла, нажатия кнопок и времени на web-server в следующем формате: «время» (hh:mm:ss), «данные от гироскопа или кнопки» (GYR/BTN:[градусы]).

4. Общая схема и принципы работы встраиваемой системы:

Схема соединений всего используемого оборудования представлена на рисунке 5. Для соединения проводников в нескольких точках используйте безопасную макетную плату, установленную в Freescale PBMUSLK и набор перемычек.

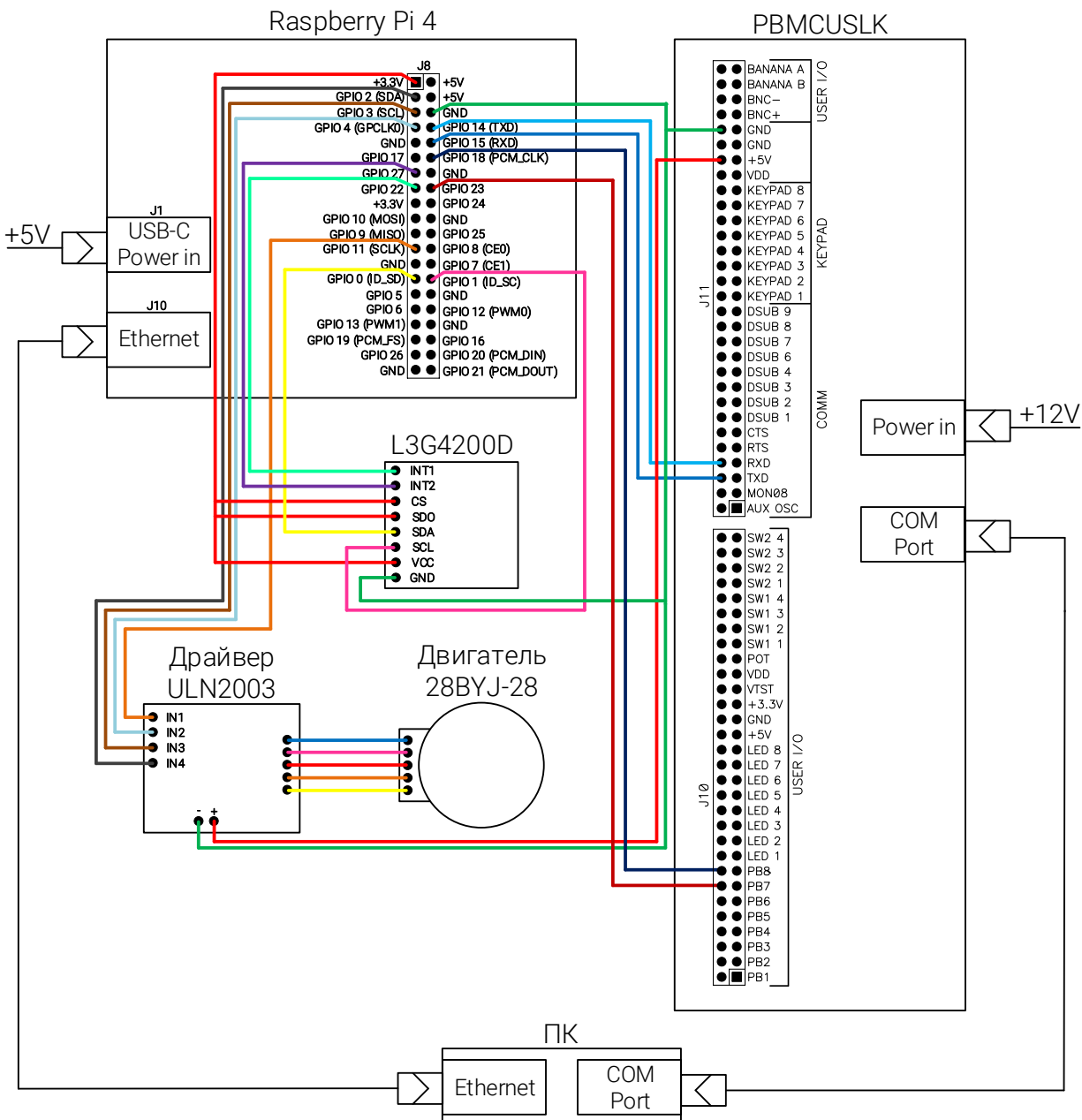


Рисунок 5. Схема соединений по варианту 5

5. Рекомендуемый порядок выполнения курсового проекта:

Выполнение курсового проекта рекомендуется делать поэтапно, разобравшись первоначально с работой периферии и интерфейсов, далее реализовать отдельные программы работы с периферией, далее модифицировать их согласно требованиям к функционированию встраиваемой системы.

Выполнение курсового проекта может осуществляться в рамках лабораторных занятий по дисциплине «Операционные системы», при этом, с

разрешения преподавателя, можно заменить индивидуальные задания лабораторной работы на следующие пункты курсового проекта, представленные в таблице 5:

Таблица 5. Задания лабораторных работ из курсового проекта по варианту 5

Лабораторная работа	Сдача этапа курсового проекта
«Процессы и межпроцессное взаимодействие»	<p>Задание №3.1. Разработайте программу, которая будет управлять шаговым двигателем по GPIO интерфейсу и записывать приращение поворота в файл вместе со временем с частотой 10 Гц. Пусть к файлу передается в качестве входного аргумента. Разработайте еще одну программу, которая будет настраивать и принимать данные с гироскопа по интерфейсу i2c и вычислять угол поворота по одной из плоскостей.</p> <p>Задание №3.2. Модернизируйте программу-2, чтобы данные об угле поворота направлялись по неименованному каналу. Напишите третью программу, которая бы по нажатию кнопок отправляла бы фиксированное значение угла поворота по неименованному каналу.</p> <p>Задание №3.3. Добавьте в 2 и 3 приложения функционал отключения и генерации сигнала SIGINT.</p>
«Разработка многопоточных программ»	<p>Задание №4.1. Модернизируйте программу-1, которая в двух независимых потоках будет принимать сообщения по неименованным каналам от программ-2, -3.</p> <p>Задание №4.2. Модернизируйте программу, добавив еще один поток обработки стандартного потока ввода/вывода для управления работой встраиваемой системы (старт и остановка, изменение скорости опроса и значения углов).</p> <p>Задание №4.3. Добавьте функцию отображения результатов работы встраиваемой системы в web-интерфейс.</p>

6. Вспомогательные материалы:

Спецификация на используемые модули:

- [микрокомпьютер Raspberry Pi 4](#);
- макетная плата Freescale PBMICUSLK: [схема](#), [описание](#), [быстрый старт](#);
- [модуль 3-х осевого гироскопа на L3G4200D](#);
- [шаговый двигатель 28BYJ-48](#);
- [драйвер электродвигателя ULN2003](#).

Описание, настройка и использование периферийных устройств Raspberry Pi 4 доступно по следующим ссылкам:

- [основы работы с GPIO в Raspberry Pi OS](#);
- [управление шаговым двигателем через драйвер ULN2003](#);
- [исходный код C для работы с L3G4200D по интерфейсу I2C](#);
- [How to Control GPIO Hardware from C or C++](#);
- [исходные коды управления драйвером ULN2003](#).

Вариант 6.

1. Описание задания:

Необходимо разработать прототип встраиваемой системы, как совокупность программного обеспечения и стенда на базе Raspberry Pi4, которая управляла бы шаговым двигателем, изменяя угол поворота по данным от энкодера или по данным, вводимым с 16-клавишной клавиатуры. Все действия должны записываться в лог-файл и отправляться на web-сервер, отображаться в окне браузера, а также дублироваться по ssh-соединению на персональный компьютер.

2. Используемые отладочные комплекты:

- микрокомпьютер Raspberry Pi 4;
- макетная плата Freescale PBMCUSLK;
- модуль Pmod ENC: Rotary Encoder PN 410-117;
- шаговый двигатель 28BYJ-48;
- драйвер электродвигателя ULN2003;
- устройство ввода Pmod KYPD: 16-button Keypad PN 410-195;
- персональный компьютер.

3. Требования к работе встраиваемой системы:

3.1 Встраиваемая система должна обеспечивать управление шаговым электродвигателем и по значениям углов поворота энкодера или введенным значениям с 16-кнопочной клавиатуры, вывод данной информации в консоль и в web-сервер за счет выполнения трех основных программ: 1 – программа, управляющая электродвигателем, 2 – программа, принимающая сигналы с датчика поворота (энкодера), 3 – программа, получающая данные от клавиатуры, программ 1 и 2 по одному из способов межпроцессного взаимодействия (именованные и неименованные каналы) и выводящее результат работы светофора и определенного цвета по последовательному каналу и на web-сервер.

3.2 Программа-1 должна обеспечивать взаимодействие RPi 4 с драйвером электродвигателя ULN2003 с помощью GPIO, принимает данные об установке угла по именованным каналам и выполняет следующие функции:

- настройка и инициализация GPIO для взаимодействия с драйвером;
- установка и хранение угла поворота двигателя от 0 до 360 градусов;
- прием значение установки угла поворота двигателя по именованным каналам;
- вычисление приращения угла и передача нужных значений по интерфейсу GPIO драйверу.

3.3 Программа-2 должна обеспечивать взаимодействие RPi 4 с датчиком поворота (энкодер) по выводам GPIO и выполняют следующие функции:

- конфигурация выводов GPIO для чтения значений с датчика поворота (энкодера);
- преобразование данных датчика в значение угла поворота;
- отправка данных об угле энкодера по именованным каналам с частотой 100 Гц.

3.4 Программа-3 должна обеспечивать взаимодействие RPi 4 с 16-кнопочной клавиатурой, обеспечивает взаимодействие с программами-1,-2, web-сервером и стандартным вводом, работает в многопоточном режиме, а также выполняет следующие функции:

- чтение значений с клавиатуры и перевод их в установочный угол поворота для электродвигателя;
- обмен сообщениями по именованным/не именованным каналам с приложениями-1,-2;
- обмен сообщениями со стандартным потоком ввода/вывода, в том числе прием команд от пользователя (см. общие требования к работе ВС);
- выдача результатов на web-сервер;

3.5 Встраиваемая система должна обеспечивать выдачу измеренных значений угла, нажатия кнопок и времени на web-server в следующем формате: «время» (hh:mm:ss), «данные от гироскопа или кнопки» (GYR/BTN:[градусы]).

4. Общая схема и принципы работы встраиваемой системы:

Схема соединений всего используемого оборудования представлена на рисунке 6. Для соединения проводников в нескольких точках используйте беспаячную макетную плату, установленную в Freescale PBMCUSLK и набор перемычек.

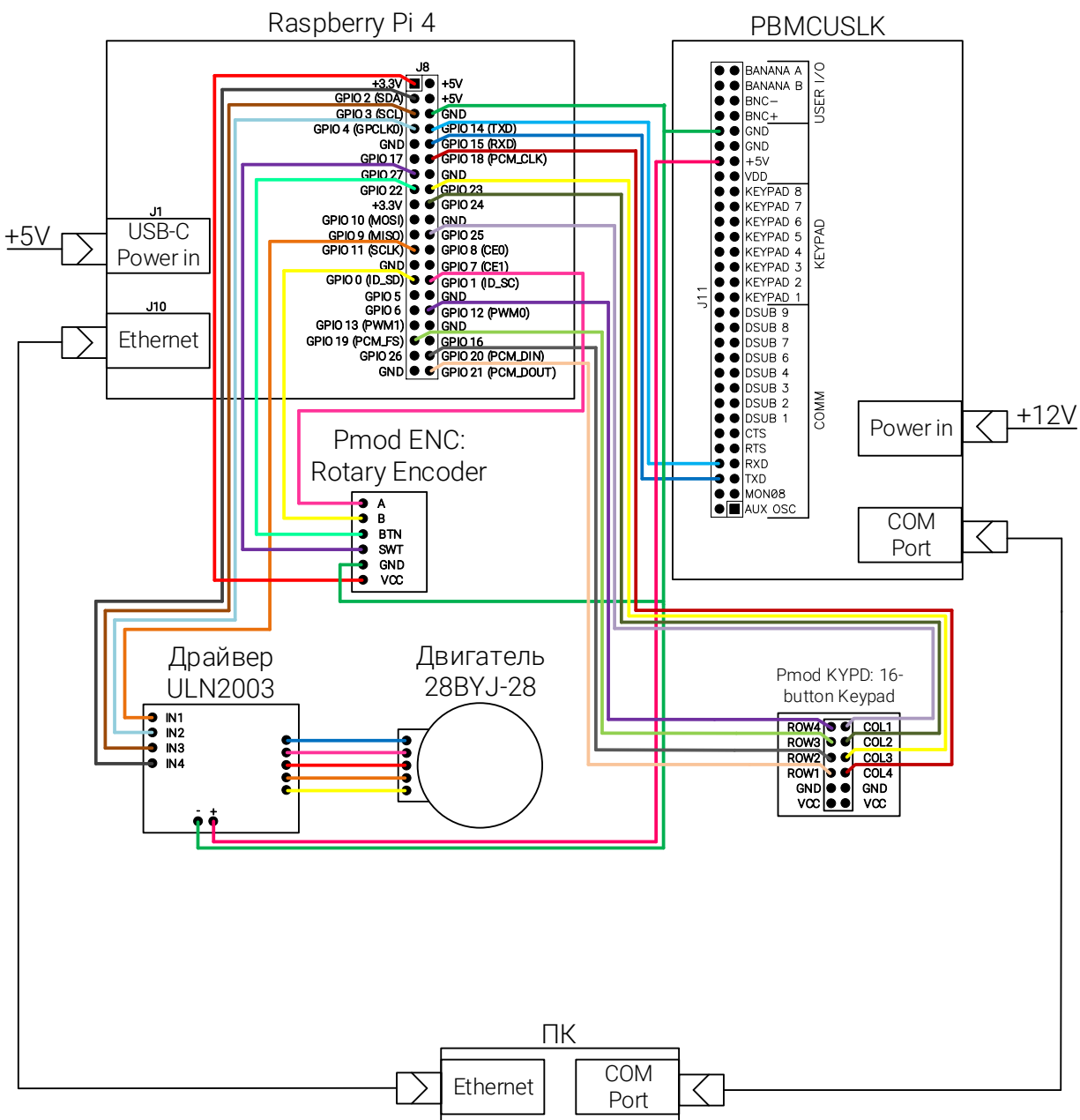


Рисунок 6. Схема соединений по варианту 6

5. Рекомендуемый порядок выполнения курсового проекта:

Выполнение курсового проекта рекомендуется делать поэтапно, разобравшись первоначально с работой периферии и интерфейсов, далее реализовать отдельные программы работы с периферией, далее модифицировать их согласно требованиям к функционированию встраиваемой системы.

Выполнение курсового проекта может осуществляться в рамках лабораторных занятий по дисциплине «Операционные системы», при этом, с

разрешения преподавателя, можно заменить индивидуальные задания лабораторной работы на следующие пункты курсового проекта, представленные в таблице 6:

Таблица 6. Задания лабораторных работ из курсового проекта по варианту 6

Лабораторная работа	Сдача этапа курсового проекта
«Процессы и межпроцессное взаимодействие»	<p>Задание №3.1. Разработайте программу-3, которая будет считывать данные угла поворота двигателя с 16-кнопочной клавиатуры. Разработайте еще одну программу-2, которая будет настраивать и принимать данные с энкодера по интерфейсу i2c и вычислять угол поворота.</p> <p>Задание №3.2. Напишите третью программу-1, которая бы управляла шаговым двигателем по GPIO. Модернизируйте программы-2,-3, что они передавали свои данные по неименованным каналам.</p> <p>Задание №3.3. Добавьте в 2 и 3 приложения функционал отключения и генерации сигнала SIGINT.</p>
«Разработка многопоточных программ»	<p>Задание №4.1. Модернизируйте программы-1, которая в двух независимых потоках будет принимать сообщения по неименованным каналам от программ-2, -3.</p> <p>Задание №4.2. Модернизируйте программу, добавив еще один поток обработки стандартного потока ввода/вывода для управления работой встраиваемой системы (старт и остановка, изменение скорости опроса и значения углов).</p> <p>Задание №4.3. Добавьте функцию отображения результатов работы встраиваемой системы в web-интерфейс.</p>

6. Вспомогательные материалы:

Спецификация на используемые модули:

- [микрокомпьютер Raspberry Pi 4;](#)
- макетная плата Freescale PBMCSLК: [схема](#), [описание](#), [быстрый старт](#);
- [модуль Pmod ENC: Rotary Encoder PN 410-117;](#)
- [шаговый двигатель 28BYJ-48;](#)
- [драйвер электродвигателя ULN2003;](#)
- [устройство ввода Pmod KYPD: 16-button Keypad PN 410-195.](#)

Описание, настройка и использование периферийных устройств Raspberry Pi 4 доступно по следующим ссылкам:

- [основы работы с GPIO в Raspberry Pi OS;](#)
- [управление шаговым двигателем через драйвер ULN2003;](#)
- [примеры исходных кодов для Pmod ENC от digilent;](#)
- [основы работы с GPIO в Raspberry Pi OS;](#)
- [исходные коды по работе с Pmod ENC;](#)
- [исходные коды управления драйвером ULN2003;](#)
- [исходные коды для работы с Pmod KYPD;](#)
- [описание и материалы по Pmod KYPD.](#)

Вариант 7.

1. Описание задания:

Необходимо разработать прототип встраиваемой системы, как совокупность программного обеспечения и стенда на базе Raspberry Pi4, которая реализует функции термовокса с помощью двух датчиков (один отвечает за громкость звука, второй за ноту) и стандартного звукового выхода. Все действия должны записываться в лог-файл и отправляться на web-сервер, отображаться в окне браузера, а также дублироваться по ssh-соединению на персональный компьютер.

2. Используемые отладочные комплекты:

- микрокомпьютер Raspberry Pi 4;
- макетная плата Freescale PBMUSLK;
- два ультразвуковых датчика HC-SR04;
- наушники;
- персональный компьютер.

3. Требования к работе встраиваемой системы:

3.1 Встраиваемая система должна обеспечивать воспроизведение звука одной октавы и пятью уровнями громкости, которые задаются двумя датчиками, вывод данной информации в консоль и в web-сервер за счет выполнения трех основных программ: 1 – программа, принимает и обрабатывает сигналы датчика № 1 (нота), 2 – программа, принимает и обрабатывает сигналы датчика №2 (громкость), 3 – программа, получает данные от программ 1 и 2 по одному из способов межпроцессного взаимодействия (именованные и неименованные каналы), высчитывает ноту и громкость, воспроизводит ее и выводит результат работы по последовательному каналу и на web-сервер.

3.2 Программы-1 и -2 должны обеспечивать взаимодействие RPi 4 с датчиком с помощью GPIO, преобразуют принятые сигналы в расстояние и выполняют следующие функции:

- настройка и инициализация GPIO для взаимодействия с датчиками;
- определение расстояния до преграды с частотой в 0,1 кГц;
- передача значений расстояния в см по неименованным каналам.

3.3 Программа-3 должны обеспечивать взаимодействие RPi 4 со стандартным аудиовыходом, с кнопками для калибровки, с программами-1,-2, web-сервером и стандартным вводом, работает в многопоточном режиме, а также выполняет следующие функции:

- обмен сообщениями по именованным/не именованным каналам с приложениями-1,-2 и прием дальностей;

- по нажатию на кнопку SEL переходит в режим калибровки расстояний для определения самой высокой ноты и максимальной громкости;
- обмен сообщениями со стандартным потоком ввода/вывода, в том числе прием команд от пользователя (см. общие требования к работе ВС);
- выдача результатов на web-сервер;

3.5 Встраиваемая система должна обеспечивать выдачу ноты и громкости, режима калибровки и времени на web-server и в консоль в следующем формате: «время» (hh:mm:ss), «нота», «громкость».

4. Общая схема и принципы работы встраиваемой системы:

Схема соединений всего используемого оборудования представлена на рисунке 7. Для соединения проводников в нескольких точках используйте беспаячную макетную плату, установленную в Freescale P BMCUSLK и набор перемычек.

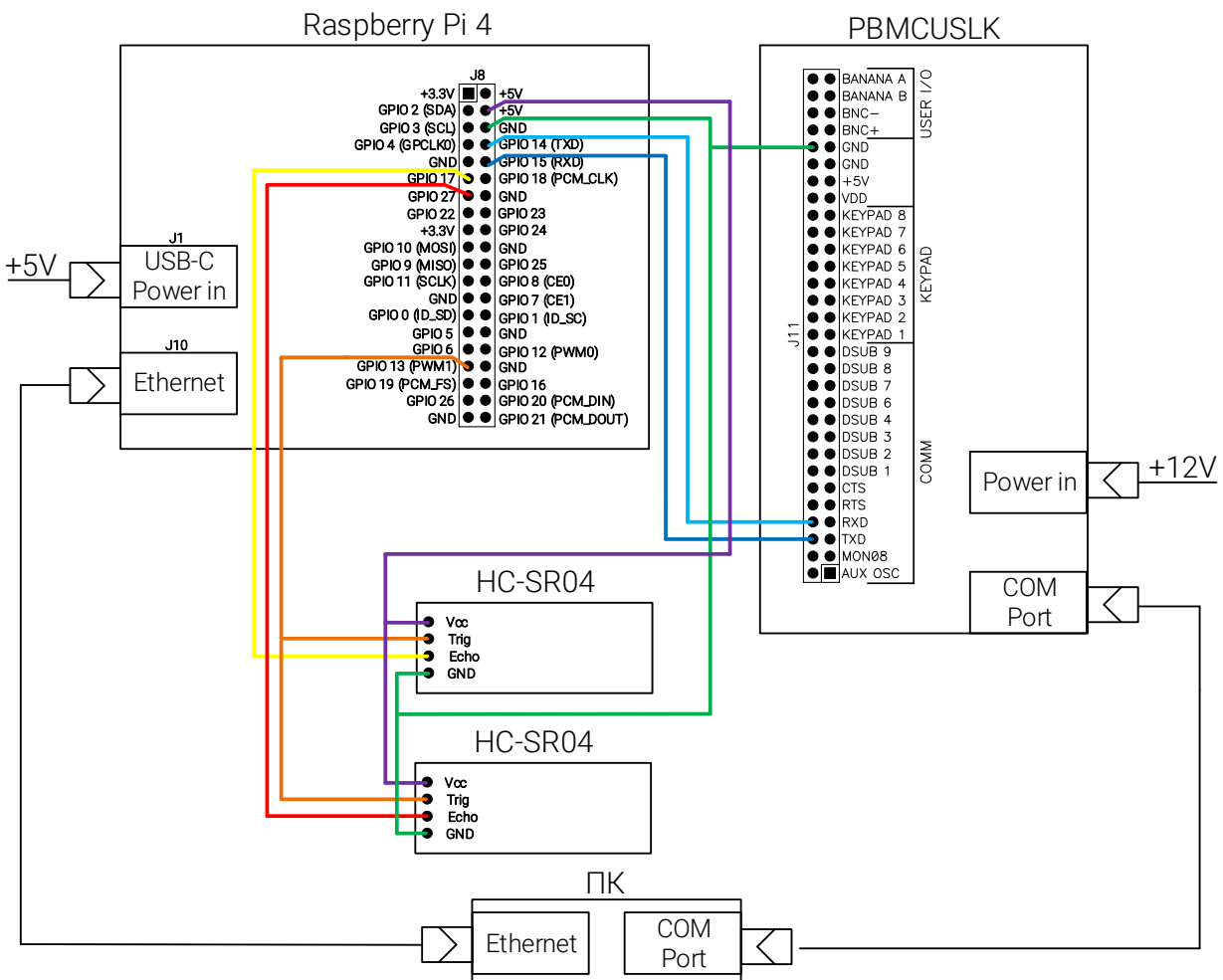


Рисунок 7. Схема соединений по варианту 7

5. Рекомендуемый порядок выполнения курсового проекта:

Выполнение курсового проекта рекомендуется делать поэтапно, разобравшись первоначально с работой периферии и интерфейсов, далее реализовать отдельные программы работы с периферией, далее модифицировать их согласно требованиям к функционированию встраиваемой системы.

Выполнение курсового проекта может осуществляться в рамках лабораторных занятий по дисциплине «Операционные системы», при этом, с разрешения преподавателя, можно заменить индивидуальные задания лабораторной работы на следующие пункты курсового проекта, представленные в таблице 7:

Таблица 7. Задания лабораторных работ из курсового проекта по варианту 7

Лабораторная работа	Сдача этапа курсового проекта
«Процессы и межпроцессное взаимодействие»	Задание №3.1. Разработайте программы-1 и -2, которые будут принимать показания дальномеров, пересчитывать их в расстояния и записывать в файл вместе со временем данных с частотой 10 Гц. Пусть к файлам передается в качестве входного аргумента. Задание №3.2. Модернизируйте программы-1,-2, чтобы данные о расстоянии направлялись по именованному каналу. Напишите третью программу, которая бы по нажатию кнопок выдавала бы на стандартный аудиовыход ноту и по нажатию на SEL – меняла громкость. Задание №3.3. Добавьте в приложения функционал отключения и генерации сигнала SIGINT.
«Разработка многопоточных программ»	Задание №4.1. Модернизируйте программу-3, которая в двух независимых потоках будет принимать сообщения по именованным каналам от программ-1, -2. Задание №4.2. Модернизируйте программу, добавив еще один поток обработки стандартного потока ввода/вывода для управления работой встраиваемой системы (старт и остановка, изменение скорости опроса и значения углов). Задание №4.3. Добавьте функцию отображения результатов работы встраиваемой системы в web-интерфейс.

6. Вспомогательные материалы:

Спецификация на используемые модули:

- [микрокомпьютер Raspberry Pi 4](#);
- макетная плата Freescale PBMUSLK: [схема](#), [описание](#), [быстрый старт](#);
- [ультразвуковой дальномер HC-SR04](#);

Описание, настройка и использование периферийных устройств Raspberry Pi 4 доступно по следующим ссылкам:

- [основы работы с GPIO в Raspberry Pi OS](#);
- [How to Control GPIO Hardware from C or C++](#);
- [исходные коды для работы с HC-SR04](#).

Вариант 8.

1. Описание задания:

Необходимо разработать прототип встраиваемой системы, как совокупность программного обеспечения и стенда на базе Raspberry Pi4, которая реализует функции системы контроля доступа по RF-метке и/или паролю. Должны быть реализованы функции смены пароля и RF-метки, данные работы выводятся на LCD-экран, статус работы отображается на led светодиодах. Все действия должны записываться в лог-файл и отправляться на web-сервер, отображаться в окне браузера, а также дублироваться по ssh-соединению на персональный компьютер.

2. Используемые отладочные комплекты:

- микрокомпьютер Raspberry Pi 4;
- макетная плата Freescale PBMCUSLK;
- устройство ввода Pmod KYPD: 16-button Keypad PN 410-195;
- RFID-модуль RC522;
- персональный компьютер.

3. Требования к работе встраиваемой системы:

3.1 Встраиваемая система должна обеспечивать контроль доступа по RF-метке и паролю, создание/редактирование/удаление данных о верной метке и пароле, отображение статуса ввода на LCD-экран, вывод информации о работе в консоль и в web-сервер. Встраиваемая система содержит три основные программы: 1 – программа, принимает и обрабатывает сигналы RFID-модуля, 2 – программа, осуществляет вывод информации на LCD-экран, 3 – программа, получает данные от кнопок клавиатуры, программ 1 и 2 по одному из способов межпроцессного взаимодействия (именованные и неименованные каналы), реализует логику создания-редактирования-удаления-принятия-отказа паролей и меток, выводит результат работы по последовательному каналу и на web-сервер.

3.2 Программа-1 должна обеспечивать взаимодействие RPi 4 с RFID-модулем по интерфейсу SPI и выполняет следующие функции:

- настройка и инициализация RFID-модуля;
- обеспечивает считывание данных с RFID-модуля по GPIO прерыванию и по опросу (конфигурируется командой через стандартный ввод\вывод);
- передача считанных данных метки по именованным каналам.

3.3 Программа-2 должна обеспечивать взаимодействие RPi 4 с LCD-экраном по SPI интерфейсу и выполняет следующие функции:

- настройка и инициализация LCD-экрана по SPI;

- прием двух строк для отображения на экране по именованному каналу.

3.4 Программа-3 должна обеспечивать взаимодействие RPi 4 с 16-кнопочной клавиатурой и выполняет следующие функции:

- считывает значения нажимаемых кнопок;
- реализует логику проверки пароля и метки, формирование соответствующих строк для LCD-экрана;
- позволяет создавать пароли, удалять их по командам с 16-кнопочной клавиатуры.
- обмен сообщениям по именованным/не именованным каналам с приложениями-1,-2;
- обмен сообщениями со стандартным потоком ввода/вывода, в том числе прием команд от пользователя (см. общие требования к работе ВС);
- выдача результатов на web-сервер;

3.8. Встраиваемая система должна обеспечивать выдачу режима работы, нажатия кнопок и времени на web-server в следующем формате: «Режим: (открыт, закрыт, ключ подошел, пароль подошел)», символы ввода пароля «*», время ввода «hh:mm:ss».

4. Общая схема и принципы работы встраиваемой системы:

Схема соединений всего используемого оборудования представлена на рисунке 8. Для соединения проводников в нескольких точках используйте беспаячную макетную плату, установленную в Freescale PBMCUSLK и набор перемычек.

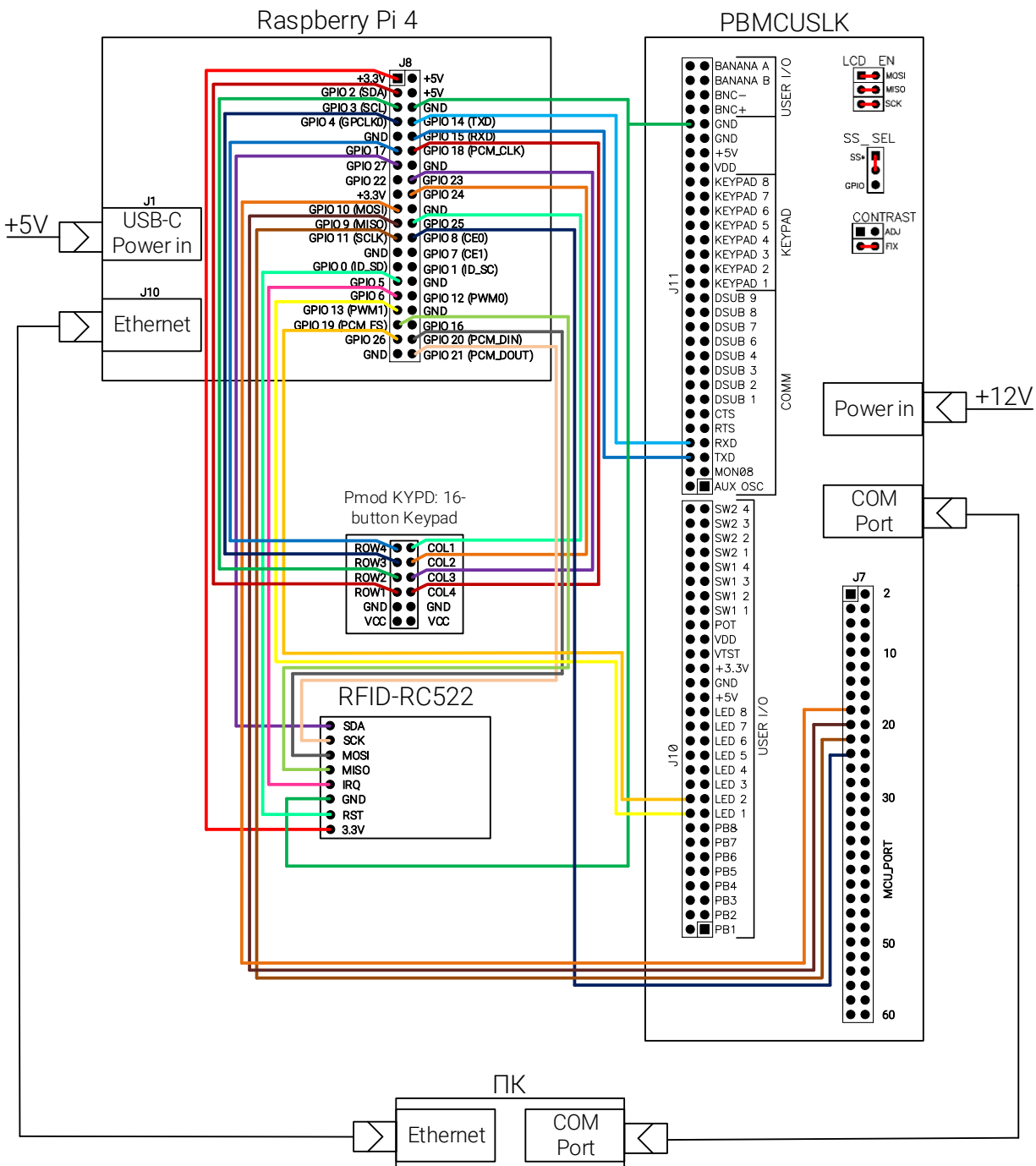


Рисунок 8. Схема соединений по варианту 8

5. Рекомендуемый порядок выполнения курсового проекта:

Выполнение курсового проекта рекомендуется делать поэтапно, разобравшись первоначально с работой периферии и интерфейсов, далее реализовать отдельные программы работы с периферией, далее модифицировать их согласно требованиям к функционированию встраиваемой системы.

Выполнение курсового проекта может осуществляться в рамках лабораторных занятий по дисциплине «Операционные системы», при этом, с разрешения преподавателя, можно заменить индивидуальные задания лабораторной работы на следующие пункты курсового проекта, представленные в таблице 8:

Таблица 8. Задания лабораторных работ из курсового проекта по варианту 8

Лабораторная работа	Сдача этапа курсового проекта
«Процессы и межпроцессное взаимодействие»	<p>Задание №3.1. Разработайте программу-1, которая будет инициализировать и принимать по SPI данные RF-метки и записывать попытки в файл вместе со временем данных. Пусть к файлам передается в качестве входного аргумента. Разработайте программу-2, которая будет выводить на LCD-дисплей две строки, вводимые с кнопок. Назначение кнопок и выводимую информацию определите самостоятельно.</p> <p>Задание №3.2. Модернизируйте программы-1,-2, чтобы данные в формате ASCII строки от приложения-1 передавались в приложение-2 по именованному каналу. Напишите третью программу, которая бы по нажатию кнопок клавиатуры выдавала бы на стандартный аудиовыход вводимые значения.</p> <p>Задание №3.3. Добавьте в приложения функционал отключения и генерации сигнала SIGINT.</p>
«Разработка многопоточных программ»	<p>Задание №4.1. Модернизируйте программу-3, которая в двух независимых потоках будет принимать сообщения по именованным каналам от программ-1, -2. Реализуйте логику взаимодействия программ, для чтения верных временных меток и паролей.</p> <p>Задание №4.2. Модернизируйте программу, добавив еще один поток обработки стандартного потока ввода/вывода для управления работой встраиваемой системы (старт и остановка, изменение скорости опроса и значения углов).</p> <p>Задание №4.3. Добавьте функцию отображения результатов работы встраиваемой системы в web-интерфейс.</p>

6. Вспомогательные материалы:

Спецификация на используемые модули расположены на удаленном репозитории Института МПСУ в разделе курса по операционным системам:

- [микрокомпьютер Raspberry Pi 4](#);
- макетная плата Freescale PBMUSLK: [схема](#), [описание](#), [быстрый старт](#);
- [устройство ввода Pmod KYPD: 16-button Keypad PN 410-195](#);
- [MFRC522](#).

Описание, настройка и использование периферийных устройств Raspberry Pi 4 доступно по следующим ссылкам:

- [описание работы с RFID модулем RC522](#);
- [команды SPI для LCD экрана платы Freescale PBMUSLK](#);
- [основы работы с GPIO в Raspberry Pi OS](#);
- [исходные коды для работы с Pmod KYPD](#);
- [описание и материалы по Pmod KYPD](#);
- [исходные коды для LCD-дисплея](#).

Вариант 9.

1. Описание задания:

Необходимо разработать прототип встраиваемой системы, как совокупность программного обеспечения и стенда на базе Raspberry Pi4, которая реализует функции полетного контролера. Данные об направлении движения, углов ориентации и высоты определяются с помощью модуля pmod NAV и датчика (высота от поверхности), данные выводятся на OLED дисплей. Все действия должны записываться в лог-файл и отправляться на web-сервер, отображаться в окне браузера, а также дублироваться по ssh-соединению на персональный компьютер.

2. Используемые отладочные комплекты:

- микрокомпьютер Raspberry Pi 4;
- макетная плата Freescale PBMCSLK;
- модуль Pmod NAV: 9-axis IMU Plus Barometer PN 410-326;
- модуль Pmod OLEDrgb: 96 x 64 RGB OLED Display;
- ультразвуковой датчик HC-SR04;
- персональный компьютер.

3. Требования к работе встраиваемой системы:

3.1 Встраиваемая система должна обеспечивать вывод навигационных параметров объекта за счет выполнения трех основных программ: 1 – программа, работающая с OLED-дисплеем и выводящая навигационные параметры, 2 – программа, принимающая сигналы с датчика и определяющая высоту объекта и передающая ее в программу-1, 3 – программа, принимающая и обрабатывающая данные навигационного модуля, передающая в программу-1 по одному из способов межпроцессного взаимодействия (именованные и неименованные каналы).

3.2 Программа-1 должна обеспечивать взаимодействие RPi 4 с модулем OLED-Display по интерфейсу SPI и выполняет следующие функции:

- настройка и инициализация OLED-дисплея;
- запись и чтение значений регистров OLED-дисплея;
- отрисовка параметров движения объекта;
- прием данных от навигационного модуля и датчика по именованным каналам;
- обмен сообщениями со стандартным потоком ввода/вывода, в том числе прием команд от пользователя (см. общие требования к работе ВС);
- выдача результатов на web-сервер;

3.3 Программа-2 должна обеспечивать взаимодействие RPi с датчиком по GPIO и выполняет следующие функции:

- запись и чтение значений GPIO и вычисление расстояния от поверхности;
- отправка значения расстояния с временной меткой системы по именованным/не именованным каналам.

3.4 Программа-3 должна обеспечивать взаимодействие с навигационным модулем по SPI, а также выполняет следующие функции:

- настройка и инициализация навигационного модуля по SPI;
- чтение и запись данных с SPI для получения навигационных параметров: углы ориентации (тангаж, рыскание, крен), ускорения по осям хуz, показания магнетометра;
- передача навигационных параметров в программу -2 по именованному каналу.

3.5 Встраиваемая система должна обеспечивать выдачу сообщений с навигационными параметрами встраиваемой системы на web-server в следующем формате: «время сообщения» (сек.), «измеряемые навигационные параметры» (град., м)

4. Общая схема и принципы работы встраиваемой системы:

Схема соединений всего используемого оборудования представлена на рисунке 9. Для соединения проводников в нескольких точках используйте безопасную макетную плату, установленную в Freescale PBMUSLK и набор перемычек.

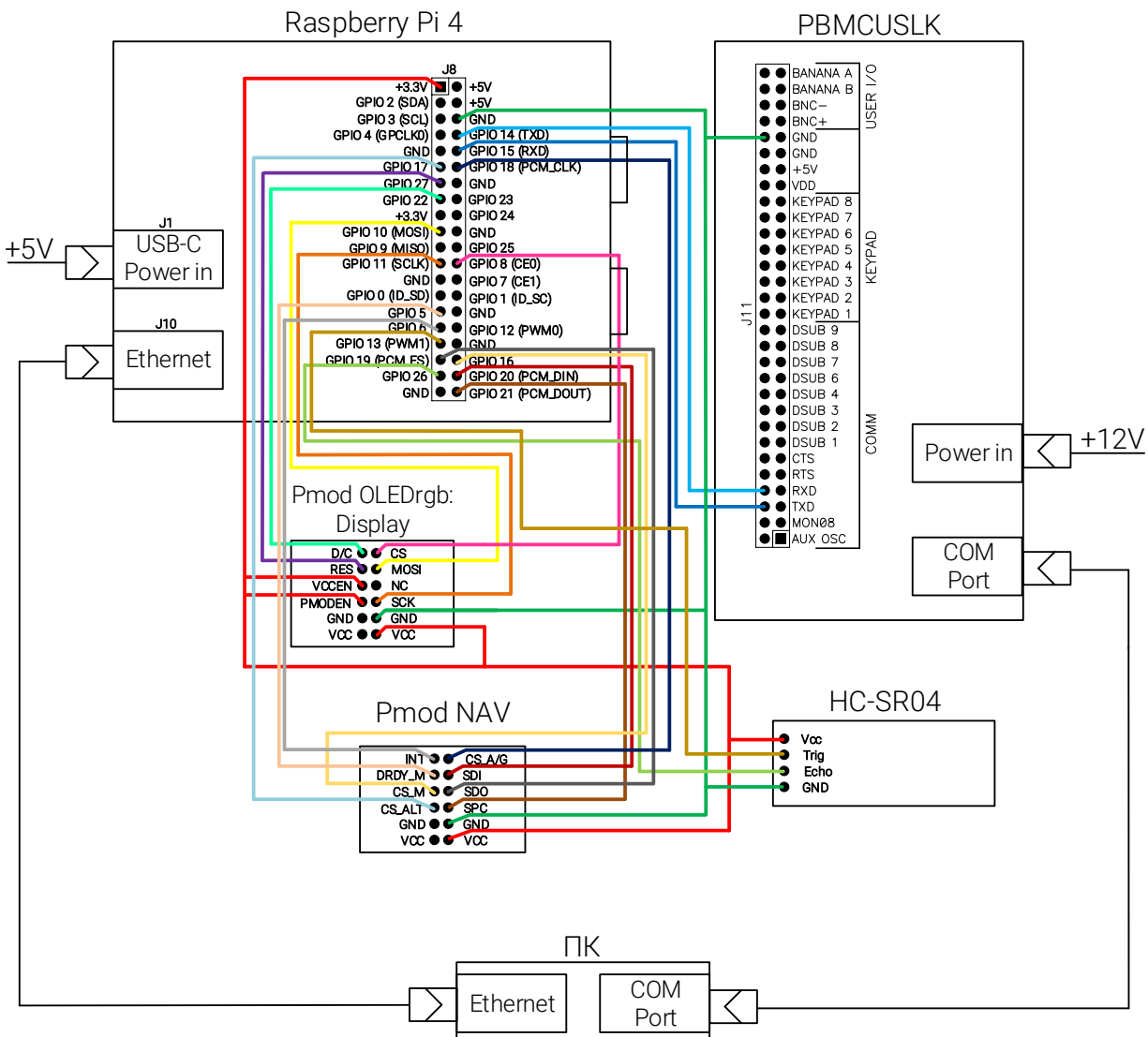


Рисунок 9. Схема соединений по варианту 9.

5. Рекомендуемый порядок выполнения курсового проекта:

Выполнение курсового проекта рекомендуется делать поэтапно, разобравшись первоначально с работой периферии и интерфейсов, далее реализовать отдельные программы работы с периферией, далее модифицировать их согласно требованиям к функционированию встраиваемой системы.

Выполнение курсового проекта может осуществляться в рамках лабораторных занятий по дисциплине «Операционные системы», при этом, с разрешения преподавателя, можно заменить индивидуальные задания лабораторной работы на следующие пункты курсового проекта, представленные в таблице 9:

Таблица 9. Задания лабораторных работ из курсового проекта по варианту 9

Лабораторная работа	Сдача этапа курсового проекта
«Процессы и межпроцессное взаимодействие»	<p>Задание №3.1. Разработайте программу-1, которая будет инициализировать и передавать по SPI данные для отображения строк данных на OLED-экране. Разработайте программу-2, которая будет принимать данные от датчика с частотой 10Гц.</p> <p>Задание №3.2. Модернизируйте программы-1,-2, чтобы данные в формате ASCII строки от приложения-2 передавались в приложение-1 по именованному каналу. Напишите третью программу, которая бы принимала значения углов поворота по SPI.</p> <p>Задание №3.3. Добавьте в приложения функционал отключения и генерации сигнала SIGINT.</p>
«Разработка многопоточных программ»	<p>Задание №4.1. Модернизируйте программу-1, которая в двух независимых потоках будет принимать сообщения по именованным каналам от программ-2, -3. Реализуйте логику взаимодействия программ, для вывода навигационных параметров.</p> <p>Задание №4.2. Модернизируйте программу, добавив еще один поток обработки стандартного потока ввода/вывода для управления работой встраиваемой системы (старт и остановка, изменение скорости опроса и значения углов).</p> <p>Задание №4.3. Добавьте функцию отображения результатов работы встраиваемой системы в web-интерфейс.</p>

6. Вспомогательные материалы:

Спецификация на используемые модули расположены на удаленном репозитории Института МПСУ в разделе курса по операционным системам:

- [микрокомпьютер Raspberry Pi 4](#);
- макетная плата Freescale PBMICUSLК: [схема](#), [описание](#), [быстрый старт](#);
- [микросхема акселерометров и гироскопов LSM9DS1](#);
- [микросхема барометра LPS25HB](#);
- [ультразвуковой датчик HC-SR04](#).

Описание, настройка и использование периферийных устройств Raspberry Pi 4 доступно по следующим ссылкам:

- [основы работы с GPIO в Raspberry Pi OS](#);
- [описание и материалы по Pmod NAV](#);
- [How to Control GPIO Hardware from C or C++](#);
- [исходные коды для работы с OLED-дисплеем](#);
- [исходные коды для работы с HC-SR04](#);
- [Исходные коды для работы с Pmod NAV](#).

Правила написания исходных кодов курсового проекта на Си.

Правило 1. Имена

Используется snake_case для всех переменных, функций, имен файлов. Заглавные буквы допускается использовать для общеизвестных аббревиатур в имени функции. Имена должны отражать назначение функций, переменных, файлов. Запрещено использовать транслитерацию в названиях функций, допустимо использовать в названиях переменных.

Неправильно:

```
2. void kakayotaCanfunctia(int VarA, can_t *Some_Var_B){
3.     Some_Var_B->PortNum = VarA;
4.     return;
5. }
```

Правильно:

```
1. void set_CAN_port(int port, can_t can_driver)
2. {
3.     can_driver->port = port;
4.     return;
5. }
```

Правило 2. Использование переменных

Запрещено использовать переменные более чем для одной цели, кроме временных локальных переменных и переменных циклов (имена temp, tmp, i, j, k, counter, count). Т.е. нельзя объявлять глобальную переменную, например, int flag и использовать ее в нескольких функциях, даже если это безопасно.

Неправильно:

```
1. int flag;
2.
3. void someFunctionA(/*...*/){
4.     if(flag){
5.         //...
6.     }
7.     return;
8. }
9.
10. void someFunctionB(/*...*/){
11.     if(flag){
12.         //...
13.     }
14.     return;
15. }
```


Правильно:

```
1. int flag_a;
2. int flag_b;
3.
4. void some_function_a(/* ... */){
5.     if(flag_a){
6.         //...
7.     }
8.     return;
9. }
10.
11. void some_function_b(/* ... */){
12.     if(flag_b){
13.         //...
14.     }
15.     return;
16. }
```

Правило 3. Дефайны, объединения, typedef

Запрещено использовать не именованные константы (магические числа).

Дефайны и поля объединений пишутся заглавными буквами.

Правило 4. Фигурные скобки

Функции обрамляются фигурными скобками следующим образом:

Открывающая фигурная скобка находится на новой строке после перечисления аргументов функции.

Закрывающая фигурная скобка находится на новой строке после оператора return.

Неправильно

```
1. void task_canfrank_receive(void *arg){
2.     // do_something
3.     return;}
```

Правильно

```
1. void task_canfrank_receive(void *arg)
2. {
3.     // do_something
4.     return;
5. }
```

Содержимое if, else, while, for, switch всегда обрамляется в фигурные скобки. Допускается не обрамлять тело оператора в скобки, когда оно помещается в одну

строку с самим оператором. Открывающая (и закрывающая для else) фигурная скобка находится на той же строке, что и оператор.

Неправильно

```
1. if(table_p != NULL)
2. {
3.     // Проверяем изменились ли данные
4.     temp[0]=table_p->data_p->data32[0];
5.     temp[1]=table_p->data_p->data32[1];
6.     //...
7. }
8. else
9.     debug_printf("Объект не найден!\n");
```

Правильно

```
1. if(table_p != NULL){
2.     // Проверяем изменились ли данные
3.     temp[0]=table_p->data_p->data32[0];
4.     temp[1]=table_p->data_p->data32[1];
5.     //...
6. } else debug_printf("Объект не найден!\n");
```

Правило 5. Пустые конструкции

Пустой цикл **for** или **while** должен содержать пустой блок {}, а не просто завершаться точкой с запятой.

Неправильно

```
1. for(i = 0; i < 100; i++)
2. for(i = 0; i < 100; i++);
```

Правильно

```
1. for(i = 0; i < 100; i++){};
```

Правило 6. Оформление файлов исходных кодов

В качестве отступов использовать 4 пробела. Документировать исходный код с помощью doxygen (описание файла с авторами и назначением, используемая лицензия, описание каждой функции и переменных). Обязательны комментарии каждого блока в теле функции, допускается использование блочных комментариев и строчных.

Критерии оценки курсового проекта

Итоговая оценка за курсовой проект выставляется на основании следующих показателей оценки, приведенных в таблице 10.

Таблица 10. Условия начисления баллов за курсовой проект

Показатель оценки	Критерий оценивания достижения показателя	Условия начисления баллов по критерию	Баллы
Сдача курсового проекта	Правильность работы стенда по варианту курсового проекта	Все требования к работе встраиваемой системы выполнены, результаты работы соответствуют требуемым	5
		Большая часть требований к работе встраиваемой системы выполнены и результаты работы соответствуют требуемым	4
		Большая часть требований к работе встраиваемой системы не выполнены, но результаты работы соответствуют требуемым	3
		Требования к работе встраиваемой системы не выполнены и/или результаты не соответствуют требуемым	0
	Правильность работы введенных дополнительных функций или изменений в работе встраиваемой системы по требованию преподавателя	Все дополнительные функции или изменения в работе встраиваемой системы внесены и работают как требовалось	5
		Большая часть дополнительных функций или изменений в работе встраиваемой системы внесены и работают как требовалось	4
		Большая часть дополнительных функций или изменений в работе встраиваемой системы не внесены, но внесенные работают как требовалось	3
		Ни одни из доп. функции или изменений не были внесены	0
	Правильность ответов на дополнительные вопросы	Были даны верно ответы по функционированию встраиваемой системы и по теории применяемых механизмов операционной системы	5
		Были даны верно ответы по функционированию встраиваемой системы, но по теории применяемых механизмов операционной системы были допущены ошибки	4
		Были даны верно ответы по теории применяемых механизмов операционной системы, но по	3

Реализация курсового проекта		функционированию встраиваемой системы были допущены ошибки	
		Не были даны верные ответы ни на один вопрос	0
	Эффективность работы программ курсового проекта	Утилизация ресурсов процессора, требуемая для исполнения всех программ проекта, не превышает 25%, а объем используемой памяти не превышает 50 МБ	5
		Одно из вышеперечисленных условий не выполняется, но функционально проект работает верно	4
		Ни одно из вышеперечисленных условий не выполняется, но функционально проект работает верно	3
		Ни одно из вышеперечисленных условий не выполняется и функционально проект не работает	0
	Использование системы контроля версий git	Исходные коды всех программ, скриптов и readme.md размещены на git, по каждому из этапов разработки проекта ведутся коммиты с содержательным описанием произведенных изменений	5
		Исходные коды всех программ, скриптов и readme.md размещены на git, но есть только один итоговый коммит	4
		Исходные коды всех программ, скриптов размещены на git, но описание readme.md отсутствует и есть только один итоговый коммит	3
		Система контроля версий не использовалась	0
	Читаемость исходных кодов программ и скриптов курсового проекта	Представленный код хорошо организован согласно предъявляемым требованиям и его просто читать, ко всем функциям и значимым участкам кода даны комментарии	5
		Представленный код хорошо организован, но оформлен не по предъявляемым требованиям; его просто читать, ко всем функциям и	4

		значимым участкам кода даны комментарии	
		Представленный код хорошо организован, но оформлен не по предъявляемым требованиям; его просто читать, но отсутствуют комментарии по коду	3
		Представленный код плохо организован и не оформлен по предъявляемым требованиям; его невозможно читать и отсутствуют комментарии по коду	0

Итоговая оценка за курсовой проект определяется как среднее арифметическое оценки по каждому критерию таблицы выше.

Список рекомендуемой литературы

- Таненбаум Э. Современные операционные системы: Пер. с англ. / Э. Таненбаум. - 2-е изд. - СПб.: Питер, 2005. - 1038 с. - (Классика Computer Science). - ISBN 5-318-00299-4; 0-13-031358-0: 505-73.