

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

_____ О.В. Непомнящий

« ____ » _____ 2024 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Проектирование программной архитектуры системы передачи
телеметрической информации

090401 Информатика и вычислительная техника

09.04.01.11 «Вычислительные системы и сети»

Руководитель	_____	_____	профессор, канд. техн. наук	О.В. Непомнящий
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	
Выпускник	_____	_____		Н.В. Коптяев
	<i>подпись</i>	<i>дата</i>		
Рецензент	_____	_____	доцент, канд. техн. наук	К.В. Раевич
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	
Консультант	_____	_____	генеральный директор ООО «ПК «Дельта»»	О.Г. Варыгин
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	
Нормоконтролёр	_____	_____	профессор, канд. техн. наук	О.В. Непомнящий
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	

Красноярск 2024

РЕФЕРАТ

Выпускная квалификационная работа посвящена проектированию программно-аппаратной архитектуры системы передачи телеметрической информации.

Пояснительная записка содержит 57 страниц текстового документа, 17 рисунков, 2 таблицы, 1 приложение, и 32 использованных источника.

СТРУКТУРИЗАЦИЯ, АРХИТЕКТУРА СИСТЕМЫ, ПЕРЕДАЧА ДАННЫХ, КОМПЛЕКС УСТРОЙСТВ, СЕТЬ, БАЗЫ ДАННЫХ, UBUNTU, APACHE, MYSQL, ОПЕРАЦИОННАЯ СИСТЕМА.

Цель работы: создание комплексной программно-аппаратной архитектуры наземного сегмента системы передачи телеметрической информации с использованием спутниковой связи, обеспечивающей эффективную передачу и обработку телеметрической информации, а также исследование, моделирование и проработка алгоритмов, связей, и технических решений для создания полностью функционирующего прототипа системы, предназначенного для разработки конечного продукта.

Задачи, решенные в процессе разработки:

- проведён аналитический обзор предметной области;
- определены критерии сравнения и сравнение различных систем передачи телеметрической информации;
- предложена возможность использования спутниковой связи, базирующаяся на исследовании существующих аналогов подобных систем со схожим функционалом;
- разработана программная модель системы передачи телеметрической информации;
- на контроллер установлена ОС Linux;
- на сервер установлено необходимое программное обеспечение для старта разработки последующих модулей и программ;
- проведено тестирование разработанных модулей.

АННОТАЦИЯ

Магистерская диссертация на тему «Проектирование программной архитектуры системы передачи телеметрической информации» выполнена в Институте космических и информационных технологий Сибирского федерального университета.

Проводятся исследования существующих систем сбора и обработки телеметрической информации. Разрабатывается программная архитектура системы передачи телеметрической информации. Разрабатываются программные средства и технические решения в системе передачи телеметрической информации. Проводится тестирование разработанных программных средств и технических решений.

Отличительной особенностью разрабатываемой системы является предложенная архитектура системы передачи телеметрической информации, базирующаяся на требованиях, предъявляемых к данной системе, позволяющая разработать программно-аппаратный комплекс передачи и обработки телеметрической информации, с поддержкой канала спутниковой связи.

Содержание

Введение	4
1 Исследования существующих систем сбора и анализа телеметрической информации.....	7
1.1 Обзор предметной области	7
1.1.1 Система «Умный дом» от Яндекса	8
1.1.2 Home Assistant	10
1.1.3 Xiaomi Smart Home.....	12
1.2 Анализ полученных результатов исследования	14
1.2.1 Критерии сравнения систем передачи телеметрической информации	14
1.2.2 Сравнение различных систем передачи телеметрической информации	15
1.3 Выводы по главе	16
2 Разработка программно-аппаратной структуры системы передачи телеметрической информации	18
2.1 Разработка программно-аппаратной архитектуры системы передачи телеметрической информации	20
2.2 Определение технических требований к модулям системы передачи телеметрической информации	23
2.2.1 Технические требования к программному обеспечению сервера	23
2.2.2 Технические требования к программному обеспечению контроллера	24
2.2.3 Технические требования к пользовательскому мобильному приложению	26
2.3 Выводы по главе	29
3 Разработка технических решений и программных средств для разработки системы передачи телеметрической информации	30
3.1 Выбор инструментальных средств для разработки и настройки программной части сервера.....	30
3.2 Разработка модели системы	31
3.2.1 Сервер.....	31

3.2.2 Выбор контроллера	33
3.2.3 Тестирование ОС Android на контроллере	34
3.2.4 Установка ОС Linux на контроллер	36
3.3 Выводы по главе	39
4 Тестирование технических решений и программных средств.....	41
4.1 Тестирование установленной на контроллер системы	41
4.2 Тестирование программных средств	43
4.3 Выводы по главе	47
Заключение.....	49
Список сокращений.....	51
Список используемых источников	52
Приложение А. Листинги разработанных программ	55

ВВЕДЕНИЕ

Актуальность проблемы: в современных условиях развития информационных технологий стоит обратить внимание на широко распространенные и разнообразные системы сбора и передачи телеметрической информации, позволяющие охватить страны и континенты, которые базируются на глобальной спутниковой системе навигации, ориентации и т.д. Это обеспечивает сбор информации с объектов практически в любой точке земного шара. Интеграция беспроводных сетей, таких как Zigbee и Wi-Fi, а также использование спутникового интернета предоставляют возможности для создания глобально доступных и управляемых систем.

Для непосредственного сбора телеметрической информации используются датчики, а также устройства связи с объектом, в роли которых выступают различные контроллеры. Телеметрической информации, собранные с датчиков, в большинстве случаев хранятся на внешних серверах, которые доставляются посредством беспроводных или же проводных сетей.

Магистерская диссертация направлена на исследование и разработку программных решений, которые могут успешно сочетать в себе техническую сложность и гибкость функционала, обеспечивая не только передачу данных, но и адаптацию к меняющимся требованиям среды. Актуальность темы подчеркивается возможностью создания универсальных и масштабируемых систем передачи телеметрической информации в различных областях применения. А также актуальной является разработка программно-аппаратного комплекса с методиками и алгоритмами сбора и обработки телеметрической информации.

Примером подобного комплекса может служить проект разработки спутниковой системы передачи телеметрической информации. Данный проект является распределённым комплексом спутниковой связи, который состоит из двух подсистем, включающих в себя систему спутниковой связи, а также наземный программно-аппаратный комплекс передачи телеметрической

информации. В данной работе будет вестись непосредственно разработка наземной части комплекса, который включает в себя совокупность серверов, контроллеров, устройств приёма и передачи телеметрической информации, и смартфона с установленным на него мобильным приложением. Все эти устройства обмениваются данными посредством беспроводных сетей.

Цель диссертационной работы заключается в создании комплексной программной архитектуры наземного сегмента системы передачи телеметрической информации.

К системе предъявляются следующие требования:

- в качестве физического канала используется спутниковый канал;
- обеспечение контроля трафика;
- на одного абонента приходится максимум два контроллера;
- к одному контроллеру можно подключить до пяти датчиков и исполнительных устройств;
- автономность системы, разворачиваемой на удаленных точках;
- возможность управления со смартфона;
- хранение и автономное выполнение сценариев на контроллере;
- сценарии должны позволять использовать не более трёх логических операции «И», и не более двух логических операций «ИЛИ».

Для достижения поставленной цели сформулированы следующие задачи:

- исследовать существующие системы сбора и анализа телеметрической информации;
- разработать программную архитектуру системы передачи телеметрической информации;
- реализовать технические решения и программные средства для системы передачи телеметрической информации;
- провести тестирование разработанной модели и программных средств.

Основное внимание уделяется не только техническим аспектам разработки, но и высокоуровневому обеспечению гибкости и масштабируемости

системы, позволяя успешно адаптировать ее к различным сценариям использования.

Предполагаемая научная новизна исследования заключается в предложенной архитектуре системы передачи телеметрической информации, базирующейся на требованиях, предъявляемых к данной системе, позволяющей разработать программно-аппаратный комплекс передачи и обработки телеметрической информации с поддержкой канала спутниковой связи.

1 Исследования существующих систем сбора и анализа телеметрической информации

1.1 Обзор предметной области

Исходя из определения термина телеметрической информации – это передаваемая телеметрической системой информация, которая включает в себя информацию о результатах измерения физических параметров, состояния контролируемых объектов, изучаемых явлениях или событиях, а также информация, обеспечивающая работу наземных средств телеметрической системы. [1]

Телеметрическая система – это совокупность устройств, обеспечивающих сбор сигналов со средств первичного преобразования, формирование телеметрических сигналов, передачу их по каналу связи, регистрацию и отображение телеметрических сообщений на приемной стороне. [1] Можно сделать вывод, что как таковые системы передачи телеметрической информации весьма схожи с устройством систем «Умного дома», которые, в свою очередь, включают в себя совокупность устройств, собирающих и отображающих информацию о событиях отслеживаемых пользователем объектов на устройстве, которое по своей сути является приёмной стороной.

Системы передачи телеметрической информации позволяют собирать данные, передаваемые от различных устройств или датчиков и обрабатывать их, а также формировать управляющие сигналы для периферийных устройств, подключенных к системе. Системы, обладающие подобным принципом работы, находят своё применение в самых разных областях – от «умного дома» до автоматизации производств.

Большинство систем передачи телеметрической информации защищено коммерческой тайной, из-за чего невозможно оценить их функционал и проанализировать архитектуру. Исходя из этого, для обзора предметной области были выбраны системы домашней автоматизации от разных производителей.

Архитектура подобных систем включает в себя набор датчиков и исполнительных устройств, а также хаб для подключения устройств и сервер, что приближает их к разрабатываемому наземному программно-аппаратному комплексу передачи телеметрической информации. Также эти системы доступны на рынке, что позволяет изучить их.

1.1.1 Система «Умный дом» от Яндекса

Система «Умный дом» от компании Яндекс [2] является самой распространённой в России. Она позволяет объединять устройства разных производителей в общие сценарии, а также управлять ими голосом через колонки Станции или напрямую со смартфона. Похожие системы есть у компаний Apple, Google, VK и Сбер, однако у Яндекса один из самых обширных каталогов совместимых устройств и продвинутое приложение, которое поддерживает сложные сценарии автоматизации [3].

Умный дом Яндекса состоит из трёх основных составляющих:

- интерфейс, который включает в себя мобильное приложение на смартфоне, или умную колонку, которые передают команды пользователя на сервер;
- сервер, который занимается обработкой запросов пользователя, передачей их на устройства, сохраняет заданные пользователем сценарии;
- умные устройства, которые участвуют в сценариях и управляются дистанционно.

Данная система требует постоянного подключения к интернету, всеми подключенными устройствами управляет непосредственно внешний дата-центр. С одной стороны, это немного замедляет реакцию на команду, однако лишает пользователя необходимости разбираться в отдельных настройках и сопряжении каждого отдельного элемента или модуля.

Для создания базовой системы умного дома Яндекса, пользователю достаточно обзавестись мобильным приложением «Дом с Алисой» [4] или же

воспользоваться браузером, а также понадобится умная лампочка, розетка или любое устройство из списка совместимых [5].

Для добавления нового умного устройства используется мобильное приложение «Дом с Алисой». Дальнейшие настройки и взаимодействия с подключенными устройствами рекомендуется выполнять именно там.

Существует три вида подключаемых устройств:

- фирменные устройства Яндекса, которые подключаются напрямую в приложении;
 - устройства сторонних фирм, например, Xiaomi, которые нужно сначала активировать, а потом связать аккаунты производителя устройства и Яндекса;
 - сторонние датчики, лампочки и выключатели с протоколом Zigbee, которые привязываются сначала к устройству-хабу, а не к аккаунту напрямую.
- В случае если в качестве хаба выступает одна из фирменных станций Яндекса (Например Станция Миди), то процесс схож с таковым в устройствах от Яндекса.

Фирменные умные устройства Яндекс подключаются через домашнюю сеть Wi-Fi к аккаунту Яндекс. Также к плюсам фирменных датчиков можно отнести 100% совместимость и простоту выбора.

Сторонние устройства необходимо добавить в отдельное приложение от их производителя, после чего объединить аккаунты с Яндексом. Все устройства одного производителя добавляются разом, повторное подключение их поодиночке не обязательно. Данный подход имеет неоспоримый плюс в виде отсутствия необходимости переключения региона (после добавления устройства в приложение Яндекса) для того, чтобы воспользоваться устройством. Данная проблема характерна для устройств компании Xiaomi, выпущенных для внутреннего рынка Китая.

Устройства Zigbee подключаются посредством хаба, который получает команды от пользователя и распределяет их между устройствами. Некоторые колонки от Яндекс имеют встроенный модуль Zigbee. Но при отсутствии

колонки можно использовать отдельный фирменный хаб Яндекс или стороннего производителя наподобие Aqara.

Создание сценариев. Сценарии – это заранее созданные пользователем программы, которые содержат в себе две переменные:

- условие — заданное время, срабатывание датчика или определенная голосовая команда;
- результат — включение или выключение устройств, проигрывание музыки, изменение настроек освещения и так далее.

Сценарии могут создаваться как в одиночку, парами и даже группами, для дополнения уже существующего сценария или же нескольких сценариев. Однако отдельный сценарий не является сложным, поддерживается только условие «ЕСЛИ» (), «ТО» (), где «()» какое-либо устройство или датчик.

Можно создать следующие виды сценариев:

- сценарии, которые запускаются голосом по ключевой фразе;
- сценарии работы устройства по расписанию;
- сценарии, по определённому событию: по сигналу от датчика или нажатию беспроводной кнопки;
- сценарии, при которых система присылает уведомление на телефон, если поступает сигнал от датчиков;
- сценарии, которые запускаются по голосу, без специальных команд (работает только в системах с «умной колонкой» «Миди») [6,7].

1.1.2 Home Assistant

Home Assistant – это программное обеспечение с открытым исходным кодом для автоматизации умного дома, которое ориентируется на локальное управление и конфиденциальность. Проект существует более пяти лет и развивается силами открытого сообщества энтузиастов. Он хорошо подходит для работы на устройствах Raspberry Pi [8] или на локальном сервере [9, 10].

Для начала работы, на компьютер нужно установить Home Assistant Operating System — система доступна на бесплатной основе на официальном сайте [11]. Для её установки нужна карта MicroSD или флешка. Windows, Linux или другая полноценная операционная система для установки не требуется.

После этого сервер, который может работать на базе Raspberry Pi, нужно подключить к домашней сети и оставить включенным. Дальнейшая настройка и использование происходит через интерфейс в браузере на обычном персональном компьютере или смартфоне. После первоначальной настройки к системе умного дома станет возможным подключение любого устройства [12].

Настройка Home Assistant происходит сложнее, чем с готовыми системами наподобие Яндексa. Как минимум, для начала пользования устройствами, придется начать с настройки собственного сервера: некоторые устройства придется перепрошить, чтобы заменить им необходимость обращаться к серверу производителя на альтернативную возможность подключения к домашнему серверу. В связи с чем могут возникнуть трудности на этапе подключения устройства. Ведь для перепрошивки потребуется компьютер и специальный программатор, помимо этого, на устройство может не оказаться альтернативной прошивки, что сделает невозможным его дальнейшее использование в составе настраиваемой системы [12].

Автоматизации являются основным инструментом управления умного дома. Основная функция автоматизаций – вызов сервисов при достижении тех или иных условий. Проще говоря, автоматизации позволяют задать действие, которое будет применяться в момент возникновения определенного события. Они состоят из триггера и действия, также может включаться еще и дополнительное условие. В приложении можно выбрать уже готовую автоматизацию, или же создать свою [10, 13].

Сценарии позволяют объединить в одну комбинацию несколько автоматизаций. Этот инструмент значительно упрощает работу с приложением. Однако, чем автоматизации сложнее и чем их больше, тем чаще случаются сбои. Рано или поздно сценарии придется обновлять. Например, после замены

неисправного старого девайса на новый, он окажется неучтенным в старых сценариях и придется производить ревизию [14].

Из этого следует, что:

- чем меньше автоматизаций, тем система надёжнее и её проще обслуживать.

- сценарии должны быть простыми, во избежание сбоев.

Уведомления, как отправка сообщений пользователю, также являются необходимой частью умного дома и позволяют ему сообщать о событиях и сигналах, полученных от устройств. Home Assistant даёт на выбор несколько вариантов возможностей:

Внутренние уведомления Home Assistant [15], которые позволяют настроить отставку уведомлений непосредственно в интерфейс Home Assistant.

Внешние уведомления [16], которые направляются в сторонние платформы через установку дополнительных интеграций – например, Slack, Telegram, Facebook Messenger [10, 13].

1.1.3 Xiaomi Smart Home

Компания Xiaomi занимается производством различной электроники от носимой портативной электроники (смартфоны, наушники, планшеты, ноутбуки), до бытовой техники (чайники, увлажнители, микроволновки, холодильники) и автомобилей. Компания имеет и свою собственную систему «Умный дом», однако её можно больше отнести к системе домашней автоматизации [17].

Систему автоматизации Xiaomi можно разделить на 3 типа устройств:

- устройства, работающие по протоколу Zigbee, с питанием от батарейки и требующие наличия одного из шлюзов управления для объединения устройств в единую сеть;

- устройства, работающие по протоколу Wi-Fi, с питанием от электросети;

- устройства, подключаемые по Bluetooth.

Однако стоит учитывать один немаловажный фактор: каждый датчик привязан к определённому региону и не будет работать с устройствами, предназначенными для другого, не совпадающего с ним региона. Поэтому нужно ответственно подходить к покупке устройств и следить, чтобы они были для того региона, в котором будет располагаться система.

Управление Xiaomi Smart Home происходит через приложение Xiaomi MiHome [18], доступное как для Android, так и для iOS.

Существует возможность голосового управления с помощью Яндекс Алисы.

Рассмотрим принцип работы сценариев в системе Xiaomi. Существуют условия «ЕСЛИ» () «ТО» (), «ЕСЛИ» () «И» () «ТО», а также «ЕСЛИ» () «ИЛИ» () «ТО». К сожалению, в одном сценарии невозможно совместить условия «И» и «ИЛИ». Для этого необходимо прибегнуть к настройке нескольких сценариев.

Условия запуска сценария: все сцены в умном доме прописаны и запускаются от определенных условий.

- запуск вручную – сценарий запускается после нажатия пользователем кнопки в приложении;
- таймер – можно задать запуск сценария в определенные дни и определенное время;
- при звонке – при звонке на мобильный телефон будет выполняться действие;
- при получении сообщения – аналогично предыдущему триггеру;
- при изменении погоды на улице – можно использовать только данные о погоде в Китае;
- срабатывание датчика или устройства – в зависимости от наличия у вас устройств, это может быть «Обнаружение протечки», нажатие кнопки, тревога на Шлюзе и т.д. [19].

Действия сценариев Mi Home зависят от выбранных условий и устройств. Так же доступно:

- запустить сценарий – можно настроить запуск сценария на определенное условие;
- включить\выключить сценарий – сценарии могут быть как включены, так и выключены, в зависимости от выбранных условий;
- задержка – отложенное выполнение действия (можно установить задержку до 1 часа);
- действие устройства – заданное действие в smart устройстве.

Можно комбинировать выполнения нескольких сценариев от одного условия – создавая сцены.

Хранение данных и обработка каждого действия или запроса происходит на стороне Китайских серверов, что замедляет взаимодействие с устройствами, а в случае сбоев на серверах и вовсе парализует работу и взаимодействие с системой.

1.2 Анализ полученных результатов исследования

Для определения требований к разрабатываемой системе передачи телеметрической информации необходимо выполнить анализ рассмотренных существующих решений. Для проведения анализа необходимо выделить ряд критериев, позволяющих сравнить эти системы.

1.2.1 Критерии сравнения систем передачи телеметрической информации

Для того, чтобы проанализировать рассмотренные выше системы передачи телеметрической информации, введены следующие критерии:

- Сложные сценарии. Под сложными сценариями подразумеваются сценарии, которые содержат более одного условия. Необходимо для возможности более точной и гибкой настройки системы.

- Возможность регулировки трафика. Необходимо для контроля трафика в условиях передачи данных по ограниченному каналу.
- Место хранения информации. Под информацией понимаются данные с датчиков и служебная информация системы.
- Место исполнения сценариев. Этот критерий определяет, где реализуется логика работы сценариев.

1.2.2 Сравнение различных систем передачи телеметрической информации

Для анализа рассмотренных ранее систем передачи телеметрической информации составлена таблица (Таблица 1) с использованием приведённых выше критериев.

Таблица 1 – Сравнение систем передачи телеметрической информации

Критерий	Умный дом Яндекс	Home Assistant	Xiaomi Smart Home
Сложные сценарии	-	-	+
Возможность регулировки трафика	-	-	-
Место хранения информации	Сервер компании	Локальный сервер	Сервер компании
Место исполнения сценариев	Удаленно	Локально	Удаленно

Из таблицы 1 видно, что возможностью создания сложных сценариев обладает только Xiaomi Smart Home. Система Home Assistant отчасти обладает возможностью создания сложных сценариев, но не отличается высокой надёжностью, поэтому не подходит под этот критерий, а система умный дом от Яндекс имеет возможность создания только простого сценария.

У всех рассмотренных систем отсутствует возможность регулировки трафика, так как условия работы этих систем не подразумевают ограничений по объёму передаваемых данных.

Во всех рассмотренных системах информация хранится на сервере – в случае Умного дома от Яндекса и Xiaomi Smart Home это сервер компании, таким образом, для работы этих систем требуется постоянное подключение к сети Интернет, что исключает их автономность. В случае Home Assistant информация хранится на локальном сервере.

Логика работы сценариев реализована аналогичным образом: в случае Умного дома от Яндекса и Xiaomi Smart Home – удалённо, а в случае Home Assistant – локально.

Проведённый анализ показал, что Home Assistant является более автономной системой из рассмотренных, однако не имеет возможностей для создания сложных сценариев, как Xiaomi Smart Home. Все три рассмотренные системы не предоставляют возможности регулировать передаваемый и принимаемый трафик, что для разрабатываемой системы является критичным.

Полученные результаты анализа позволяют разработать план для дальнейшей работы по теме исследования.

1.3 Выводы по главе

Выполнено исследование существующих систем сбора и анализа телеметрической информации. Для определения требований к разрабатываемой системе проведён обзор предметной области.

Определены основные системы передачи телеметрической информации. Все они относятся к классу «умный дом». Среди этих систем: система умный дом от компании Яндекс, Home assistant, Xiaomi Smart Home.

Проведён анализ полученных результатов исследования. Для его выполнения введены критерии сравнения систем передачи телеметрической информации.

На основании проведённого обзора источников и введённых критериев выполнено сравнение различных систем передачи телеметрической информации. В ходе сравнения выявлено, что все представленные системы не

имеют возможности регулировать передаваемый и принимаемый трафик, что для разрабатываемой системы передачи телеметрической информации является критичным. Возможность создания сложных сценариев имеется только в системе от компании Xiaomi. Системы Умный дом от Яндекса и Xiaomi Smart Home нуждаются в постоянном подключении к интернету, что снижает их автономность.

В связи с полученными результатами анализа существующих систем передачи телеметрической информации необходимо учитывать выявленные недостатки и особенности для дальнейшей разработки системы.

Полученные в ходе проведённой работы результаты позволяют перейти к разработке программной структуры системы передачи телеметрической информации.

2 Разработка программно-аппаратной структуры системы передачи телеметрической информации

На сегодняшний день в нашей стране разрабатываются разные проекты, такие как «Скиф» (среднеорбитальная система спутниковой связи) [20], или же система «Гонец» (многофункциональная система спутниковой связи) [21]. К таким проектам можно отнести и проект ООО «ПК «Дельта»», находящейся в г. Красноярск. Проект представляет собой распределённый комплекс спутниковой связи, который позволит абоненту при помощи спутникового канала связи получать данные от датчиков и управлять устройствами, установленными на больших расстояниях от абонента. Такой подход удобен для управления автоматизированными системами, расположенными в труднодоступных районах, где нет иных способов связи, кроме спутниковой.

Анализ предметной области, проведенный в главе 1 данной работы, показал, что рассмотренные системы передачи телеметрической информации обладают схожей архитектурой: все они имеют сервер, который является логическим центром системы, и к которому подключаются все периферийные устройства. При этом если сервер Home Assistant развернут локально, соединение с серверами других систем требуют постоянного подключения к сети Интернет. В условиях соединения по каналу спутниковой связи такой подход недопустим, так как пропускная способность спутникового канала невысока. В связи с этим было принято решение сделать логическим центром системы не сервер, а контроллер, что является отличительной особенностью разрабатываемой программно-аппаратной структуры. Контроллер – это устройство, к которому будут подключаться периферийные устройства. К категории периферийных устройств в разрабатываемой системе относятся датчики для сбора телеметрической информации, и исполнительные устройства, которые принимают управляющие сигналы от контроллера при помощи подключенных к нему внешних приемо-передатчиков.

В связи с принятым архитектурным решением, для контроллера следует разработать программное обеспечение, позволяющее выполнять все функции, такие, как:

- добавление сценариев и их работа;
- получение и обработка данных от датчиков;
- отправка управляющих сигналов для исполнительных устройств;
- формирование и запись логов работы контроллера;
- получение и обработка запросов с сервера или от пользователя.

Сервер также необходим в разрабатываемой системе, но его функции, по сравнению с рассмотренными аналогами, меняются, и составляют:

- хранение служебной и необходимой пользовательской информации;
- передача пользовательских запросов на контроллер;
- обновление ПО на контроллерах.

Пользователю необходимо подключаться к системе, чтобы иметь возможность управлять подключенными к контроллеру устройствами и получать информацию с датчиков. Для этого необходимо разработать пользовательское приложение для смартфона, которое позволит пользователю:

- настраивать контроллер;
- создавать сценарии для контроллера;
- получать актуальную информацию о работе контроллера.

Таким образом, определены основные устройства системы передачи телеметрической информации: контроллер, сервер и пользовательское устройство. Также имеется спутниковая часть системы, но для непосредственно процесса передачи телеметрической информации она играет роль канала передачи данных. В связи с этим условимся разделять систему передачи телеметрической информации на две части: систему спутниковой связи и наземный программно-аппаратный комплекс передачи телеметрической информации. В данной работе описывается процесс разработки программной архитектуры и технических решений для наземной части системы передачи телеметрической информации.

2.1 Разработка программно-аппаратной архитектуры системы передачи телеметрической информации

На основе требований к системе передачи телеметрической информации, определённым заданием на ВКР и указанным во введении данной пояснительной записки, разработана программно-аппаратная архитектура, схема которой представлена на рисунке 1.

Все устройства комплекса передачи телеметрической информации обмениваются данными посредством беспроводных сетей, однако существует возможность подключения контроллера к сети Интернет с помощью проводного соединения для доступа к серверу.

Главными компонентами системы (Рисунок 1) являются контроллер, сервер и пользовательское устройство (смартфон), а также спутниковая сеть передачи данных, которая служит каналом связи между контроллером и сервером. Однако, при подключении к сети Интернет через Wi-Fi или LAN-соединение, контроллер может обмениваться данными с сервером без участия спутника. Пользовательское приложение, установленное на смартфон, имеет возможность подключения к контроллеру через сервер.

Основная функция контроллера в предложенной системе: сбор телеметрической информации с подключенных к нему датчиков, а также обработка, хранение и по запросу пользователя, либо по сценарию, передача данных через сервер, либо же напрямую в приложение на смартфоне. Помимо датчиков, к контроллеру подключены и исполнительные устройства, управление которыми осуществляется при помощи преобразованных посредством модуля приемо-передатчика сигналов, поступающих с контроллера. Ещё одна важная функция: работа со сценариями: их создание, запуск и постоянное отслеживание выполнения заданных пользователем условий. Сценарии позволяют реализовать необходимую пользователю автоматизацию, а также контролировать объём передаваемых по сети данных от датчиков.

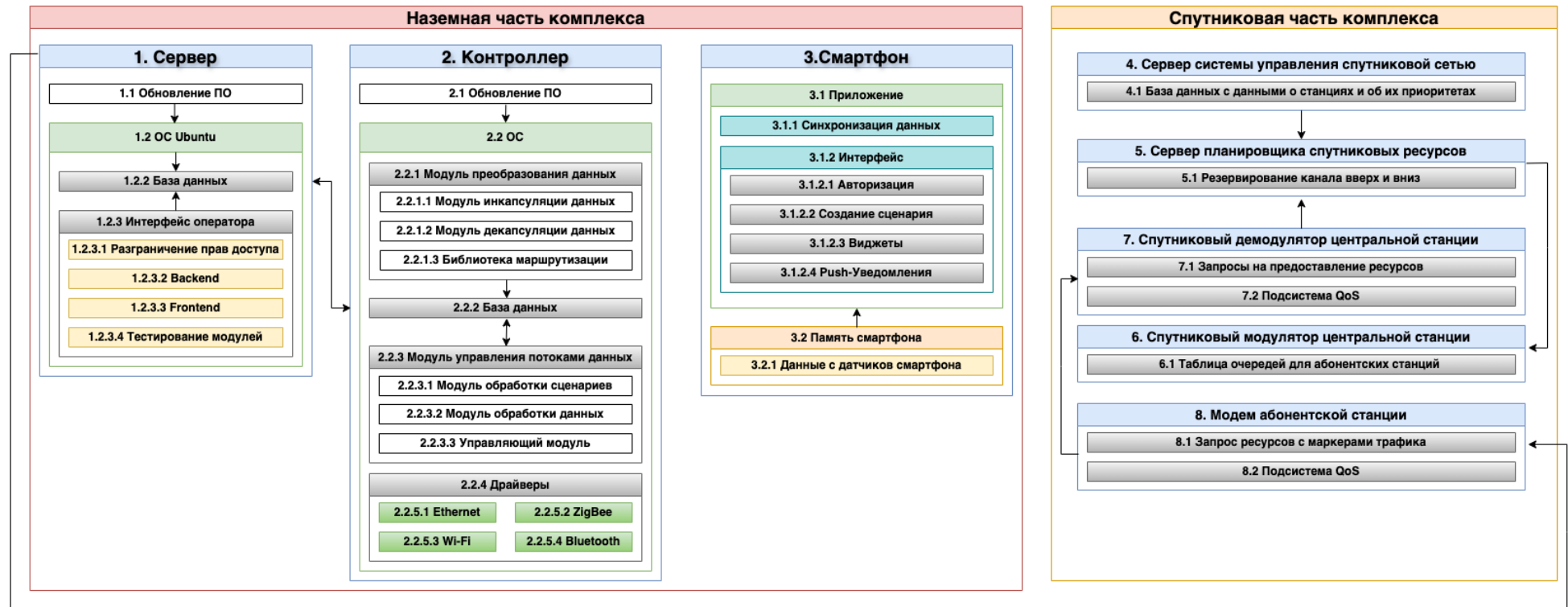


Рисунок 1 – Структурная схема программно-аппаратной архитектуры системы передачи телеметрической информации

На контроллер будет установлена ОС Linux, со встроенными в систему приложениями, необходимыми для работы подключаемых к контроллеру периферийных устройств и модулей. Контроллер подключается к интернету через Wi-Fi точку доступа, либо подключаемый спутниковый или же сотовый модем.

На смартфон пользователя должно быть установлено мобильное приложение, или же должен использоваться специальный бот телеграм, в котором планируется возможность регистрации нового пользовательского аккаунта, авторизации в аккаунт, просмотра поступающей на контроллер информации с датчиков, создание сценариев для контроллера по сбору данных и управлению исполнительными устройствами, а также возможность хранения записей данных с датчиков, по установленному пользователем временному промежутку.

Сервер используется для хранения системной и пользовательской информации, и для дублирования информации с контроллера. Также на сервере хранятся файлы актуального системного ПО для контроллера и внешних периферийных устройств.

Между контроллером, сервером и пользовательскими устройствами передаются такие данные как: показания датчиков, сигналы управления от контроллера на внешние управляющие устройства, информация о состоянии и работе контроллера, а также подключенных периферийных устройств. В том числе такие данные, которые относятся к сценариям, надстройки системы контроллера, которые пользователь вводит вручную. Эти данные необходимо синхронизовать и обработать для корректного отображения на всех подключенных устройствах к одной учетной записи.

Для начала работы, пользователю необходимо скачать приложение на смартфон. Войти, или зарегистрировать новый аккаунт. Аккаунт используется для синхронизации данных между всеми зарегистрированными в нём устройствами, просмотра технической информации о подключенных датчиках, добавления новых датчиков. После авторизации у пользователя появляется

возможность настройки конфигурации подключенного датчика, а также возможность выбора одного из предустановленных сценариев или же настроить полностью свой. После чего подключенные периферийные устройства будут управляться контроллером согласно сценарию.

Пользователь должен иметь доступ к интернету посредством подключения к тому же маршрутизатору, находясь в зоне его покрытия, или же любым другим доступным способом, например, через мобильную связь. Управление подключенными к контроллеру датчиками может быть осуществлено из любой точки земного шара, где есть подключение к интернету.

Разрабатываемая система может применяться для отслеживания показаний температуры отопительного котла, противопожарных датчиков, а также температуры и давления в контурах отопления, и теплого пола в загородном доме. Использование данной системы позволит пользователю, который, например, живет за городом, но на работу уезжает в город, отслеживать и корректировать нужные для него параметры, находясь на расстоянии, что может значительно упростить жизнь, а также повысить её безопасность.

2.2 Определение технических требований к модулям системы передачи телеметрической информации

Для разработки программной архитектуры наземного программно-аппаратного комплекса передачи телеметрической информации определены технические требования к устройствам, входящих в комплекс, и программному обеспечению, устанавливаемому на эти устройства.

2.2.1 Технические требования к программному обеспечению сервера

Сервер разрабатывается для наземной части проекта спутниковой системы передачи телеметрической информации. Он должен иметь необходимые для работы модули:

1. Прокси-сервер.
2. База данных (БД). В состав БД входят:
 - 2.1. Встроенные в БД инструменты шифрования.
 - 2.2. Таблицы БД, а том числе:
 - 2.2.1. Данные учетной записи пользователя (УЗП).
 - 2.2.2. Данные об устройствах, привязанных к УЗП (S/N, ID).
 - 2.2.3. Данные о работе устройств, привязанных к УЗП
 - 2.2.4. Каталог актуальных датчиков и контроллеров.
3. Логи работы сервера.
4. Файлы актуальных прошивок устройств (Контроллеры, датчики, мобильное приложение).
5. Web-интерфейс оператора. Можно выделить следующие основные функции:
 - 5.1. Модерирование учетных записей (блокировка/удаление подозрительных аккаунтов).
 - 5.2. Просмотр состояния контроллера (В случае сбоя. Доступ к логам работы проблемного устройства).

Инструментальные средства и программное обеспечение для разработки системы, отвечающей заявленным требованиям, выбираются по усмотрению разработчика.

2.2.2 Технические требования к программному обеспечению контроллера

Контроллер системы передачи телеметрической информации должен выполнять все основные функции этой системы. Программное обеспечение контроллера должно состоять из следующих основных программных компонентов:

1. Протокол взаимодействия базы данных контроллера с интерфейсом (Мобильное приложение). Должен включать в себя такие функции как:

1.1. При подключении контроллера к первому подключившемуся смартфону блокируется возможность редактирования другими устройствами, которым достаётся лишь просмотр информации.

1.2. Перенос сценариев, созданных пользователем, на контроллер.

1.3. Обмен данными при авторизации.

2. Протокол взаимодействия базы данных контроллера с сервером:

2.1. При подключении контроллера к первому подключившемуся смартфону блокируется возможность редактирования другими устройствами, только просмотр информации.

3. Модуль работы со сценариями, он же модуль управления потоками данных. Должен обладать следующими функциями:

3.1. обработка сценариев в текстовом виде;

3.2. обработка данных, поступающих от датчиков, указанных в сценарии;

3.3. согласно сценарию, запуск необходимых действий, таких, как:

3.3.1. отправка данных на устройство пользователя, на сервер, или в базу данных;

3.3.2. отправка управляющих сигналов на исполнительные устройства.

4. База данных. Содержит следующие данные:

4.1. данные о пользователе;

4.2. список подключенных датчиков;

4.3. информация о последних полученных данных с каждого датчика;

4.4. список подключенных исполнительных устройств;

4.5. список сценариев для контроллера;

4.6. информация о настройках контроллера.

5. Модуль сбора и хранения логов работы контроллера.

6. Модуль обновления ПО. Доступные способы обновления:

6.1. OTA (over-the-air) обновление;

6.2. Перепрошивка flash-памяти через аппаратный интерфейс UART, или через использование загрузчика. (Для отладки устройства на этапе разработки ПО).

7. Модуль преобразования данных должен выполнять следующие функции:

7.1. Декапсуляция данных, получаемых от датчиков, т.е. отделение данных от служебной информации.

7.1.1. Возможные протоколы передачи данных от датчиков: Zigbee, Wi-Fi.

7.2. Обработка полученных данных.

7.3. Инкапсуляция данных в IP-пакеты для отправки на сервер или пользовательские устройства.

8. Модуль маршрутизации данных должен сделать возможной передачу данных по следующим маршрутам:

8.1. Контроллер – сервер;

8.2. Контроллер – пользовательское устройство;

8.3. Контроллер – периферийные устройства.

Инструментальные средства и программное обеспечение для разработки системы, отвечающей заявленным требованиям, выбираются по усмотрению разработчика.

2.2.3 Технические требования к пользовательскому мобильному приложению

Разрабатываемое пользовательское мобильное приложение (МП) должно работать под управлением OS Android версии от 7.0 и выше. МП разрабатывается для наземной части проекта спутниковой системы передачи телеметрической информации.

Приложение должно состоять из следующих основных программных компонентов:

1. Интерфейс. Интерфейс должен быть минималистичным и интуитивно понятным для пользователя. Должен включать в себя такие функции как:

1.1. Авторизация. Функционал, который должен быть в этом меню:

1.1.1. Вход в уже существующий аккаунт;

- 1.1.2. Создание нового аккаунта;
- 1.1.3. Сброс пароля (кнопка «Забыли пароль?»);
- 1.1.4. Вход через SMS (После ввода номера телефона пользователю будет предложено ввести логин, после чего на номер высылается сгенерированный пароль из 8 символов, который будет необходимо заменить в настройках в течение недели (условно) и дописать недостающие данные в настройках (см. п. 1.2.))
- 1.1.5. Другие способы входа (Например вход через VK или Google)
- 1.2. Создание нового аккаунта. От пользователя требуются ввести такие данные и действия как:
 - 1.2.1. Логин (До 32 символов);
 - 1.2.2. Пароль (До 128 символов);
 - 1.2.3. Емейл (До 50 символов);
 - 1.2.4. Номер мобильного телефона (формата: +78001234567, 12 символов);
 - 1.2.5. Ответ на секретный вопрос (выбор одного из трёх предложенных вопросов);
- 1.3. Профиль пользователя. Содержит уточняющую информацию о пользователе с возможностью ее правки:
 - 1.3.1. ФИО (64+64+64 символа)
 - 1.3.2. Город (50 символов)
 - 1.3.3. Данные банковской карты для оплаты тарифного плана (19 цифр)
- 1.4. Сценарии.
 - 1.4.1. Тип сценария
 - 1.4.1.1. По умолчанию (10 шт.)
 - 1.4.1.2. Пользовательский (5 шт.)
 - 1.4.2. Функции:
 - 1.4.2.1. Запуск сценария вручную. (Запуск из приложения напрямую);
 - 1.4.2.2. Запрос параметров (Состояния котроллера/датчиков)
 - 1.4.2.3. Время. (Запуск в определенные назначенные дни или часы);

1.4.2.4. Функции (И, ИЛИ, ЕСЛИ, ТО, ПЕРЕСЛАТЬ, ВКЛЮЧИТЬ, ВЫКЛЮЧИТЬ);

1.4.2.5. При звонке. (При звонке на мобильный телефон выполняется действие);

1.4.2.6. При получении сообщения. (Суть схожа с 1.4.5.3.);

1.4.2.7. При срабатывании датчика или устройства.;

1.4.2.8. Функция сохранения;

1.4.2.9. Функция редактирования;

1.4.2.10. Опрос подключенных датчиков (для отображения в конструкторе сценариев)

1.4.3. Условия запуска/остановки сценария: события и функции;

1.4.3.1. Ограничение на количество операций с: И (3 операции); ИЛИ (2 операции); ЕСЛИ (1 операция); ТО (1 операция).

1.5. Добавление нового устройства.

1.5.1. Добавление нового контроллера. Открывается меню, после нажатия на кнопку «Добавить новое устройство». После чего будет открыт каталог, где после выбора нужного устройства будет доступно на выбор одно из двух действий:

1.5.1.1. «Добавление нового устройства с помощью камеры» (QR код на нижней части контроллера или датчика);

1.5.1.2. «Добавление нового устройства с помощью ввода его S/N» (В случае если доступ к камере невозможен) (Поле для ввода текста, ограниченного количеством символов в серийном номере (ID датчика));

1.5.2. Добавление нового датчика. Действия аналогичны с таковыми в пункте 1.5.1.

2. Запрос с сервера данных:

2.1. Об информации с датчиков, подключенных к контроллеру (Температура, влажность. В т.ч. данных о самих датчиках S/N, состояние и т.д.).

2.2. О принудительном получении информации по наличию нового обновления ПО (Кнопка проверки обновлений в настройках в разделе

«Обновление ПО». После нажатия кнопки может открываться магазин, откуда было установлено приложение, где можно будет в дальнейшем его обновить.).

Инструментальные средства и программное обеспечение для разработки системы, отвечающей заявленным требованиям, выбираются по усмотрению разработчика.

2.3 Выводы по главе

Предложена программно-аппаратная архитектура системы передачи телеметрической информации. При этом:

1. Определены основные устройства в наземной части системы: сервер, контроллер, пользовательское устройство.

2. Логический центр системы перенесён с сервера на контроллер, что отличает разрабатываемую систему от аналогов, а также позволяет минимизировать передаваемый между контроллером и сервером трафик, что является важным при использовании спутникового канала связи.

3. Определены основные функции каждого устройства в наземной части системы.

4. На основе требований к наземной части системы, определённых заданием на ВКР, разработана структурная схема программно-аппаратной архитектуры системы передачи телеметрической информации, наглядно представляющая предложенные архитектурные решения.

5. На основе разработанной архитектуры определены технические требования к программному обеспечению сервера, контроллера, пользовательских устройств. Описаны основные программные модули, которые должны быть разработаны для реализации основных функций разрабатываемой системы.

Полученные результаты позволяют перейти к разработке технических решений и программных модулей для реализации их в системе передачи телеметрической информации.

3 Разработка технических решений и программных средств для разработки системы передачи телеметрической информации

После разработки программно-аппаратной структуры и описания основных требований к аппаратным и программным модулям, для реализации были выбраны несколько подсистем в модулях 1. Сервер и 2. Контроллер (Рисунок 1), а именно, база данных и элементы интерфейса оператора на сервере, и операционная система контроллера. Эти подсистемы являются необходимыми для функционирования разрабатываемого программно-аппаратного комплекса, поэтому было принято решение сделать их минимальную реализацию.

3.1 Выбор инструментальных средств для разработки и настройки программной части сервера

Для реализации программных модулей на сервере, который является важной частью системы, необходимо подготовить и установить нужные модули для работы. Начнем с развертывания сервера.

В качестве инструментальных средств для развертывания сервера и работы с базами данных (БД) были выбраны:

1. Язык программирования PHP [22]. Язык был выбран для разработки программных модулей, предназначенных для обработки запросов и данных. Выбор пал на этот язык программирования ввиду своей универсальности и кроссплатформенности, а также распространённости в данной области.

2. LAMP [23]. Программа представляет собой набор серверных программ, который устанавливается туда комплектом приложений, необходимых для развертывания сервера. Программа устанавливает непосредственно сервер Apache [24], БД MySQL [25] с элементами управления ей, а также инструменты для работы с PHP. Причиной, по которой был выбран этот набор, является его распространённость, надёжность и удобство установки.

3. Операционная система Linux Ubuntu 22.04 LTS [26]. Бесплатный дистрибутив Linux, который был выбран в качестве ОС для сервера ввиду возможности бесплатного использования, а также большой распространенности.

4. База данных MySQL [25]. Свободная реляционная система управления базами данных. Данная СУБД была выбрана в связи с её свободным распространением, работе с реляционными базами данных, а также большим количеством инструкций и подробной документации.

5. PhpMyAdmin [27]. Инструмент для управления БД MySQL, написанный на языке php. Был выбран в связи с возможностью бесплатного использования и крайне большой распространённостью.

После выбора перечисленных выше программных средств, на рабочее место был установлен стационарный компьютер с возможностью выхода в интернет, на который впоследствии было установлено выбранное ПО. После чего была начата дальнейшая разработка.

3.2 Разработка модели системы

На начальном уровне разработки прорабатывается программная модель системы передачи телеметрической информации. По ходу разработки учитываются идеи, в том числе и на будущее расширение возможного функционала. Исходя из этого, некоторые модули могут находиться лишь на ранней стадии разработки или тестирования.

Рассмотрим модули системы поподробнее.

3.2.1 Сервер

Как упоминалось ранее в пункте 3.1, на сервер была установлена операционная система (ОС) Linux Ubuntu 22.04 LTS. Далее на него был установлен комплекс программного обеспечения (ПО) LAMP, после чего была произведена базовая настройка БД, с добавлением пароля для пользователя root.

Также были созданы новые таблицы с тестовыми данными. Произведена настройка PhPMyAdmin (Рисунок 2), с использованием пользователя root. А также общее тестирование работоспособности системы.

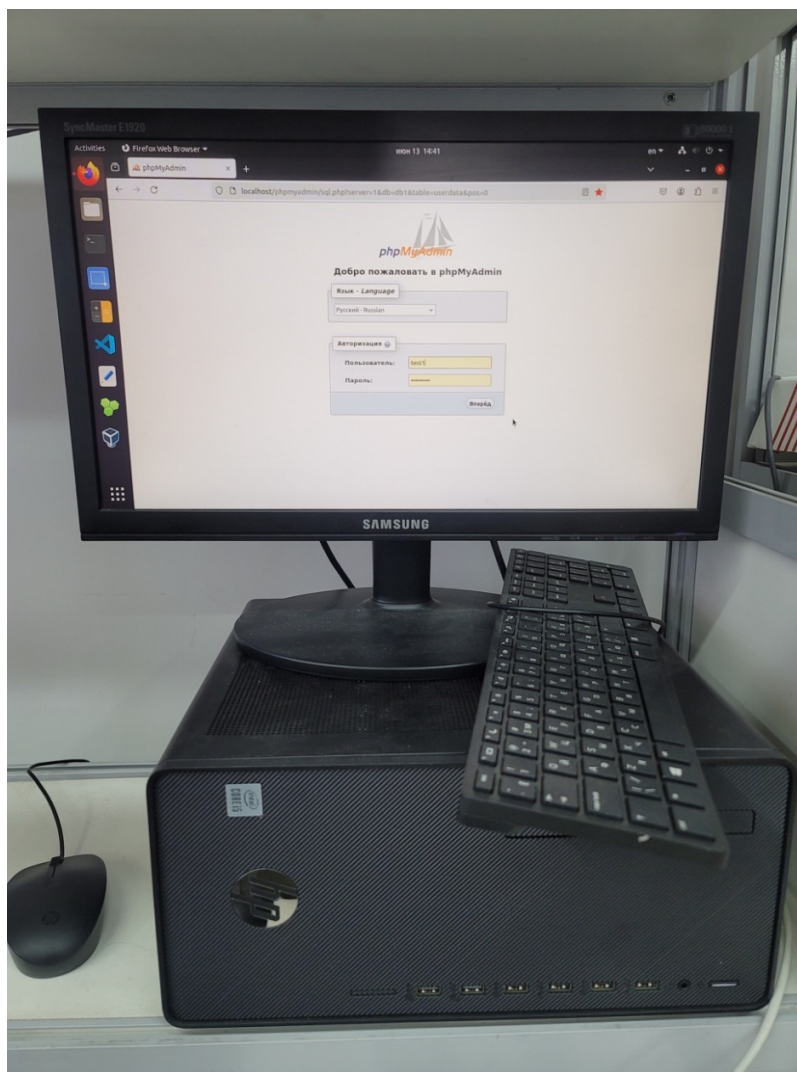


Рисунок 2 – Сервер с открытой на нём панелью PhPMyAdmin

Комплекс ПО, описанный в пункте 3.1, был выбран в результате нескольких неудачных попыток установки и настройки серверного ПО, которое не подходило из-за несовместимости между программами, или же с характеристиками контроллера.

Сервер должен служить станцией авторизации новых пользователей, хранить данные их учетных записей, а также созданные пользователями сценарии. В том числе, сервер должен выдавать информацию по запросу

пользователя, например такую как: состояние датчика, редактирование уже существующего сценария. В том числе, предусмотрено хранение данных всех подключенных устройств к одной пользовательской учетной записи.

3.2.2 Выбор контроллера

В качестве контроллера рассматривалось устройство, которое должно иметь в себе процессор, несколько usb портов, для подключения внешних модулей приёмо-передатчиков, поддерживающих различные протоколы по типу LoRaVan, Zigbee, Zwave, а также для подключения устройств ввода и вывода сигналов, модуль Wi-Fi, или в случае отсутствия такового Ethernet порт, для доступа в интернет, модуль оперативной памяти (ОЗУ), модуль встроенной памяти (ПЗУ).

В результате поиска и анализа возможных вариантов, в качестве контроллера было решено использовать ТВ-медиаплеер Hyundai h-dmp100, характеристики которого представлены в таблице 2 [28]:

Таблица 2 – Технические характеристики Hyundai h-dmp100

ОС	Android 10
Видеоадаптер	Mali-G31
Процессор	Allwinner H616 ARM QuadCore
ОЗУ	2 Гб
ПЗУ	16 Гб
Порты	Hdmi, LAN, USB 2.0 (2шт), Microsd слот, аудио-видео
Поддержка интерфейсов	Wi-Fi, Bluetooth
Стоимость	3990 р.

Устройство имеет четырёхъядерный процессор Allwinner H616 с тактовой частотой одного ядра 1.5 ГГц, совместно с графическим процессором Mali-G31, 2 Гб оперативной и 16 Гб встроенной памяти, чего вполне достаточно для комфортной работы в, предустановленной заводом изготовителем,

операционной системе Android 10, а также имеет все необходимые для работы порты. Наличие двух USB портов добавляет устройству модульности и позволяет подключать нужные внешние периферийные устройства, такие как внешние приёмо-передатчики, отсутствующие в заводской комплектации устройства. Посредством LAN порта можно производить подключение к сети интернет, где отсутствует возможность подключения через Wi-Fi, или же где сигнал такового нестабилен. Приятным дополнением являются наличие слота под Micro-SD карту, что позволяет расширить объём встроенной памяти, без необходимости разбора устройства.

3.2.3 Тестирование ОС Android на контроллере

Завод-изготовитель устанавливает на контроллер ОС Android 10 версии с предустановленным лаунчером (оболочкой системы) (Рисунок 3).



Рисунок 3 – Лаунчер контроллера Hyundai H-DMP100

В ходе исследования данной системы командой разработчиков проекта был сделан вывод, что необходимо рассмотреть возможность замены текущей системы на какой-либо полноценный дистрибутив Linux.

ОС Android построена на ядре Linux, однако для полноценного использования возможностей системы, а также установки и настройки необходимого программного обеспечения, необходима установка стороннего приложения, эмулирующего терминал. В качестве такого приложения выбран эмулятор терминала Termux [29]. При тестировании данного приложения установлено, что команды взаимодействия с системой через терминал (Рисунок 4) отличаются от команд, используемых в ОС Debian, Ubuntu и подобных им, а необходимые для дальнейшей разработки приложения и модули не существуют, либо не работают в ОС Android из-за отсутствия необходимых библиотек и программных компонентов.

Для работы с файловой системой ОС Android установлено приложение Root Explorer (Рисунок 5), т.к. оно даёт возможность просмотра и модификации файлов, которые скрыты в предустановленном файловом менеджере.

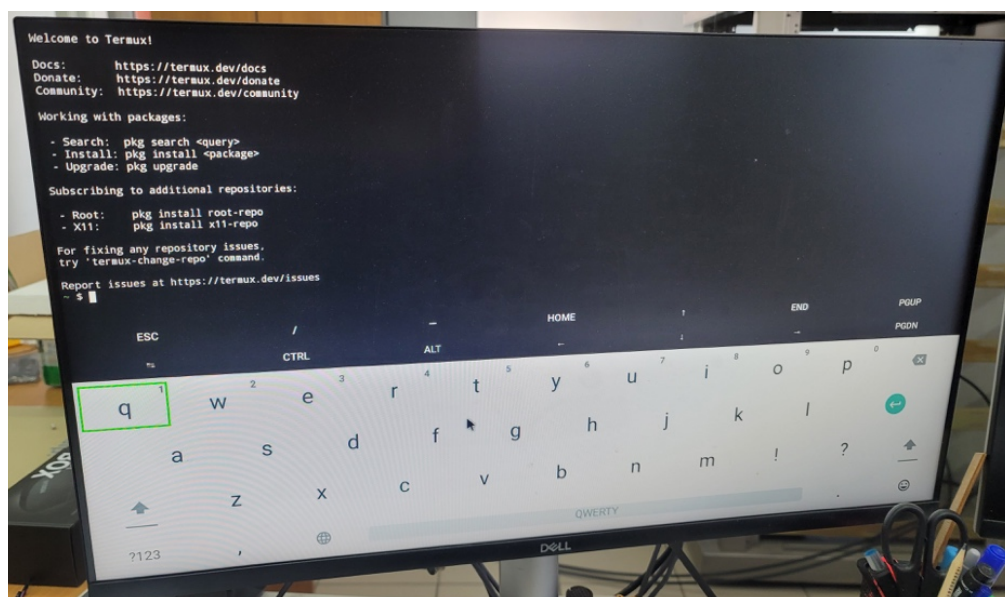


Рисунок 4 – Интерфейс приложения Termux на контроллере

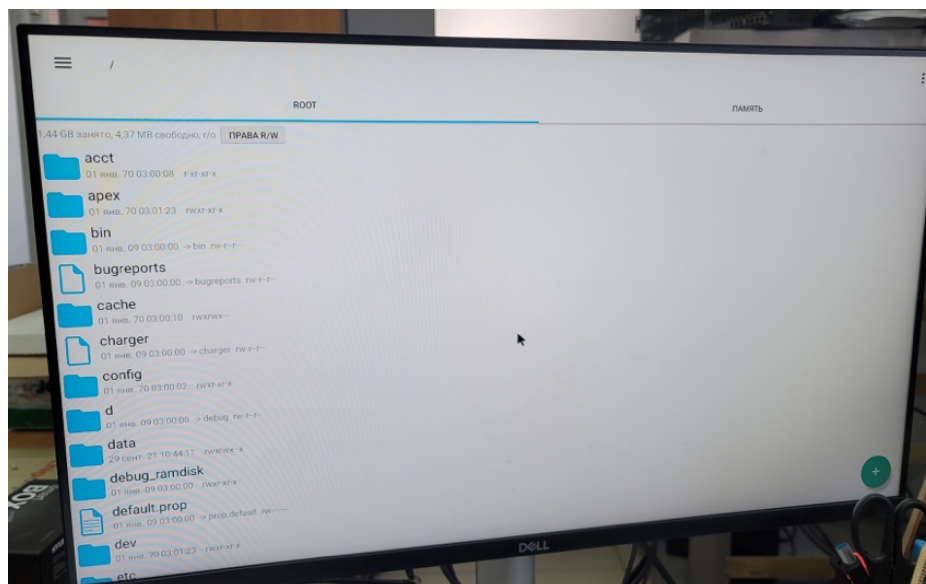


Рисунок 5 – Файловый менеджер Root Explorer на контроллере

Для функционирования программных модулей, устанавливаемых на контроллер необходимо стороннее программное обеспечение, например, MySQL, или Zigbee2MQTT. Несоответствия в командах и отсутствие многих программных пакетов и библиотек существенно затрудняет процесс установки и настройки этого ПО, замедляя при этом процесс разработки модулей и подсистем из состава системы передачи телеметрической информации.

Для того, чтобы исключить подобные трудности и задержки в работе над проектом, было принято решение установить на контроллер ОС Linux для процессора с архитектурой ARM.

3.2.4 Установка ОС Linux на контроллер

Для установки сторонней ОС на контроллер потребуется модификация встроенного заводом-изготовителем загрузчика ОС. Загрузчик (он же Bootloader) является аналогом Bios в компьютерах. Он грузится в первую очередь, а затем даёт сигнал на запуск операционной системы. Зачастую производители блокируют доступ к загрузчику, так как не желают, чтобы пользователь устанавливал другую операционную систему на их устройство, кроме предусмотренной ими [30]. Доступ к загрузчику контроллера Hyundai

также заблокирован, однако имеется возможность потенциального запуска обновления системы через слот MicroSD.

Был написан новый загрузчик и пересобрана вместе с ним версия aarch64 GNU/Linux версии 6.5.0-rc5. Возможность запуска загрузчика, а также легковесность рассматриваемого дистрибутива, позволила скопировать файлы системы на карту памяти объемом 2 Гб, и загрузиться прямым в систему посредством специального USB UART донгла. Для подключения этого донгла, необходимо разобрать устройство, корпус которого состоит из двух половинок, крепящиеся между собой при помощи клипс по всему периметру их прилегания, а также металлической пластины с термопрокладкой, встроенной в верхнюю половинку для лучшего отвода тепла от процессора (Рисунок 6).

На плате присутствуют специальные контакты отладки (VCC, TX, RX, GND), которые используются для отладки на производстве, в нашем случае VCC использоваться не будет. Через эти контакты и будут проводиться дальнейшие манипуляции с устройством. Для удобства и надёжности конструкции, к ним был припаян цельный трёхконтактный разъём, в который вставляются три провода, имеющие на концах коннекторы, которые подключаются к плате и к UART донглу соответственно (Рисунок 7, 8).



Рисунок 6 – Контроллер Hyundai H-DMP100 в разборе

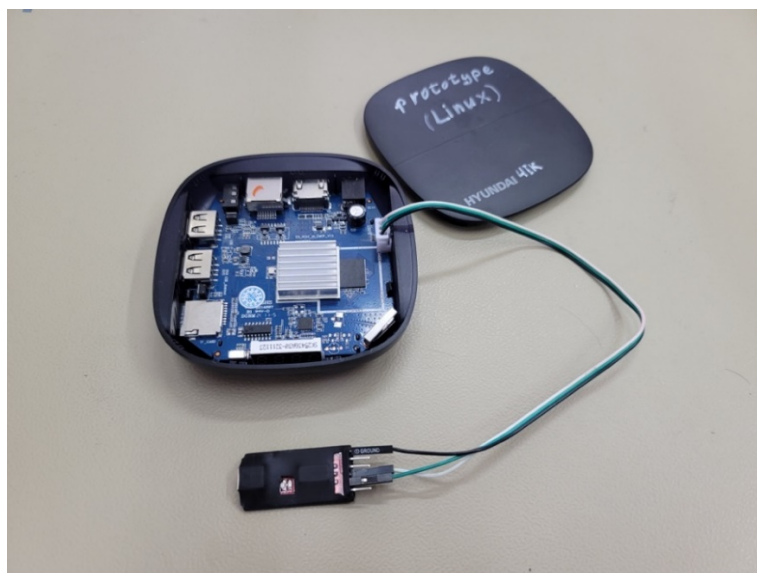


Рисунок 7 – Контроллер Hyundai H-DMP100 с подключенным UART донглом

При подключении, способ которого описан выше, контроллер загружается с карты памяти, на которой установлен дистрибутив aarch64 GNU/Linux версии 6.5.0-rc5. Таким образом, на контроллер была установлена новая операционная система, на которую в дальнейшем можно устанавливать необходимое программное обеспечение.

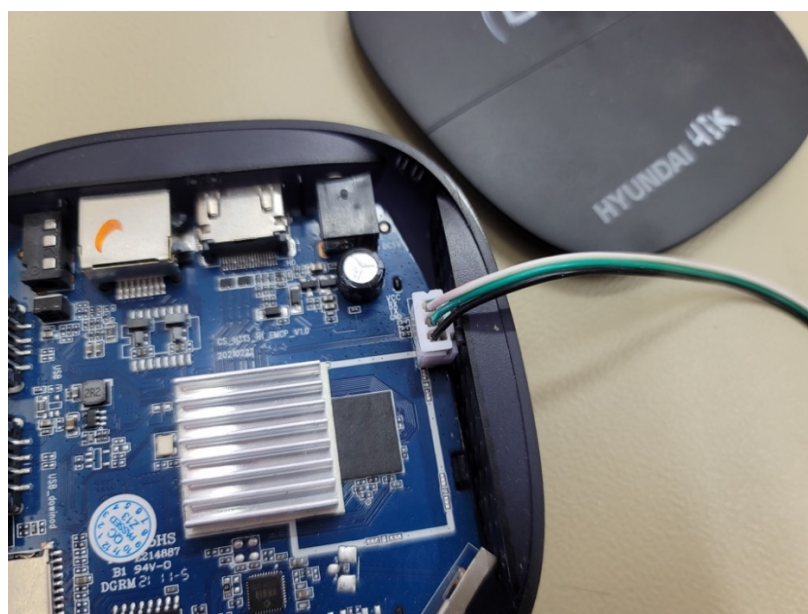


Рисунок 8 – Установленный разъём отладки на плате контроллера Hyundai H-DMP100

3.3 Выводы по главе

По завершению этапа разработки архитектуры наземной части системы передачи телеметрической информации была взята задача реализации аппаратной платформы контроллера, а также базы данных и элементов интерфейса оператора на сервере.

Программное обеспечение сервера является основой для работы других программных модулей и важной частью системы передачи телеметрической информации. Проведена следующая работа:

1. Выбраны программные средства для работы с сервером, в том числе средства разработки: язык PHP, ОС Ubuntu, база данных MySQL, а также инструмент управления БД PhpMyAdmin.

2. На персональный компьютер, выполняющий роль сервера, установлены вышеперечисленные программные средства при помощи утилиты LAMP. Произведена настройка данного программного обеспечения.

Так как в разрабатываемой системе контроллер является основным элементом, который реализует все основные её функции, подготовка аппаратной платформы контроллера является важной задачей. Для её выполнения выполнено следующее:

1. Выбрано аппаратное обеспечение, которое может использоваться в качестве контроллера передачи телеметрической информации: ТВ-медиаплеер Hyundai h-dmp100.

2. Проведено исследование операционной системы Android, изначально установленной на ТВ-медиаплеер Hyundai h-dmp100, с целью определения, подходит ли она в качестве ОС контроллера передачи телеметрической информации. В результате исследования сделан вывод, что Android – неподходящая ОС ввиду сложностей, возникших при установке и настройке необходимого программного обеспечения.

3. Для дальнейшей установки на контроллер выбрана ОС aarch64 GNU/Linux версии 6.5.0-rc5. Подготовлен дистрибутив для загрузки ОС с карты памяти.

4. Проведено исследование схемотехники контроллера, найдена шина UART, подключение к которой позволяет загрузить стороннюю операционную систему с карты памяти.

5. Для контроллера разработан загрузчик, позволяющий загружать контроллер с карты памяти.

6. Проведено подключение к контроллеру USB UART донгла.

7. Выполнена загрузка выбранной операционной системы с карты памяти.

Полученные результаты позволяют перейти к тестированию технических решений и программных средств.

4 Тестирование технических решений и программных средств

Под техническими решениями подразумевается контроллер, с установленной на него системой, описанной в пункте 3.2.4. Проведено тестирование работы установленной на контроллер операционной системы.

Под программными средствами подразумевается программное обеспечение, установленное на сервер и описанное в пункте 3.2.1.

4.1 Тестирование установленной на контроллер системы

Тестирование технических решений на контроллере проводилось после установки системы, по мере завершения этапов её сборки и установки на устройство. На данном этапе тестирования, проверяется работа ОС установленной на контроллер.

ОС с загрузчиком хранится на карте памяти формата MicroSD, объёмом 2 Гб. Целью этого этапа является проверка корректности работы системы со внутренними электронными компонентами контроллера, а также правильность работы загрузчика. Тестирование проводилось посредством ОС Linux, установленной на рабочем компьютере. Было установлено соединение с контроллером по интерфейсу UART через программу PuTTY [31], которая позволяет отслеживать логи работы подключенного устройства, а также направлять на устройство управляющие команды. Запуск ОС происходит посредством запуска загрузчика на карте памяти.

На рисунках 9, 10 приведен вывод загрузчика при запуске системы. Можно увидеть, что компоненты операционной системы запускаются в нормальном режиме, загрузка завершается корректно.

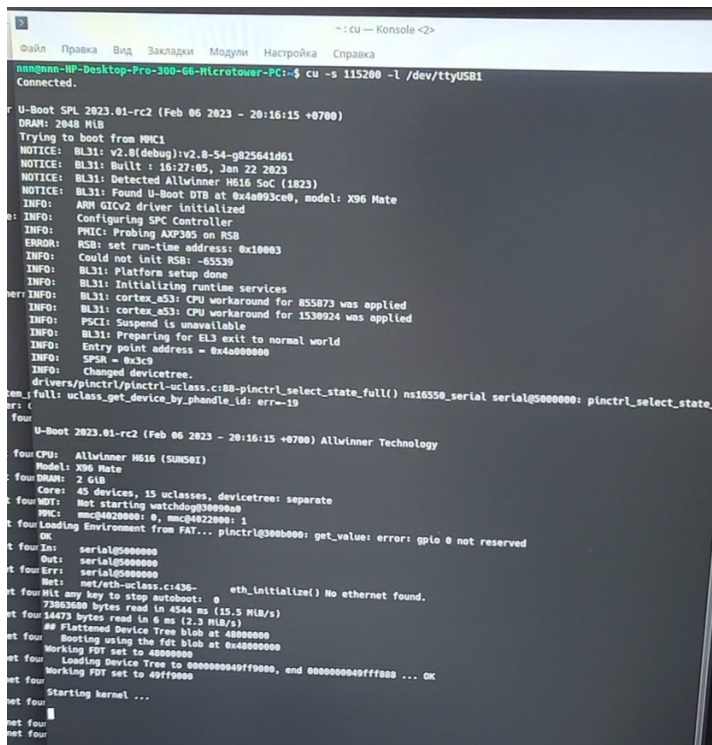


Рисунок 9 – Инициализация комплектующих контроллера при старте

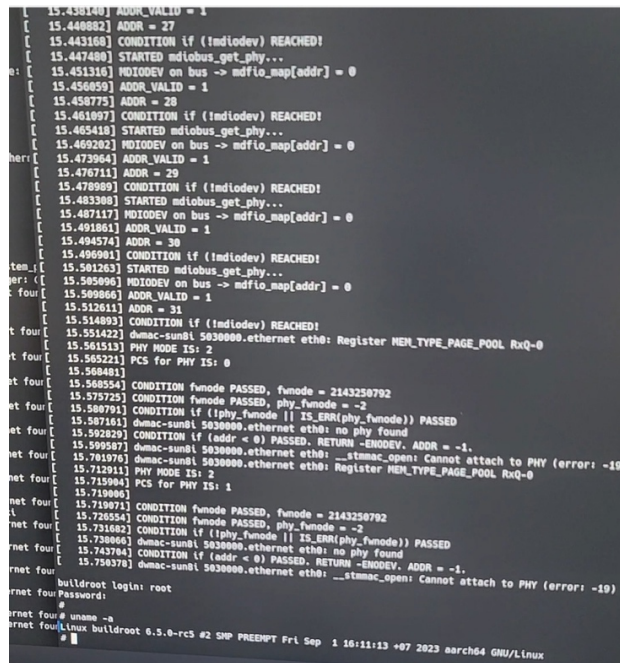


Рисунок 10 – Старт системы, авторизация и её версия

На рисунке 11 приведен скриншот работы с системой через консоль PuTTY. Авторизован пользователь root, обладающий соответствующими правами. Также на рисунке показан корневой каталог системы, что позволяет оценить целостность файловой системы.

```

[ 15.512611] ADDR = 31
[ 15.514893] CONDITION if (fmdiodev) REACHED!
[ 15.551422] dmac-sun8l 5030000.ethernet eth0: Register MEM_TYPE_PAGE_POOL RxQ-0
[ 15.561513] PHY MODE IS: 2
[ 15.565221] PCS for PHY IS: 0
[ 15.569481]
[ 15.568254] CONDITION fmode PASSED, fmode = 2143250792
[ 15.575725] CONDITION fmode PASSED, phy_fmode = -2
[ 15.580791] CONDITION if (!phy_fmode || IS_ERR(phy_fmode)) PASSED
[ 15.587161] dmac-sun8l 5030000.ethernet eth0: no phy found
[ 15.592829] CONDITION if (addr < 0) PASSED. RETURN -ENODEV. ADDR = -1.
[ 15.59587] dmac-sun8l 5030000.ethernet eth0: __stmmac_open: Cannot attach to PHY (err
[ 15.701976] dmac-sun8l 5030000.ethernet eth0: Register MEM_TYPE_PAGE_POOL RxQ-0
[ 15.712911] PHY MODE IS: 2
[ 15.715944] PCS for PHY IS: 1
[ 15.719893]
[ 15.719971] CONDITION fmode PASSED, fmode = 2143250792
[ 15.726554] CONDITION fmode PASSED, phy_fmode = -2
[ 15.731682] CONDITION if (!phy_fmode || IS_ERR(phy_fmode)) PASSED
[ 15.738666] dmac-sun8l 5030000.ethernet eth0: no phy found
[ 15.743704] CONDITION if (addr < 0) PASSED. RETURN -ENODEV. ADDR = -1.
[ 15.750378] dmac-sun8l 5030000.ethernet eth0: __stmmac_open: Cannot attach to PHY (err

buildroot login: root
Password:
_#
# uname -a
Linux buildroot 6.5.0-rc5 #2 SMP PREEMPT Fri Sep 1 16:11:13 +07 2023 aarch64 GNU/Linux
# [ 43.569801] audit: type=1334 audit(1071560184.284:16): prog-id=13 op=UNLOAD
# [ 43.572754] audit: type=1334 audit(1071560184.284:17): prog-id=12 op=UNLOAD
# [ 43.579948] audit: type=1334 audit(1071560184.284:18): prog-id=11 op=UNLOAD
# lscpu
# bash: lscpu: command not found
# ls -l /
total 4
drwxr-xr-x 1 root root 7 Aug 30 2023 bin -> usr/bin
drwxr-xr-x 15 root root 3700 Dec 20 18:15 dev
drwxr-xr-x 2 root root 800 Dec 20 18:15 etc
lrwxr-xr-x 1 root root 40 Aug 30 2023 home
drwxr-xr-x 1 root root 462 Sep 1 2023 init
lrwxr-xr-x 1 root root 7 Aug 30 2023 lib -> usr/lib
lrwxr-xr-x 1 root root 3 Aug 30 2023 lib64 -> lib
drwxr-xr-x 10 root root 11 Aug 30 2023 linuxrc -> bin/busybox
drwxr-xr-x 2 root root 200 Aug 30 2023 media
drwxr-xr-x 2 root root 40 Jun 16 2023 net
drwxr-xr-x 153 root root 40 Jun 16 2023 opt
drwxr-xr-x 3 root root 0 Jan 2 1979 proc
drwxr-xr-x 14 root root 60 Sep 1 2023 root
lrwxr-xr-x 1 root root 360 Dec 20 18:15 run
drwxr-xr-x 2 root root 0 Aug 30 2023 sbin -> usr/sbin
drwxr-xr-x 13 root root 40 Aug 30 2023 srv
drwxr-xr-x 1 root root 0 Jan 2 1979 sys
drwxr-xr-x 7 root root 160 Dec 20 18:15 tmp
drwxr-xr-x 8 root root 100 Sep 1 2023 usr
# ls --help

```

Рисунок 11 – Корневой каталог системы

Тестирование показало, что система загружается, корректно отображает компоненты контроллера, такие как объём ОЗУ, объём встроенной памяти, название процессора и платы. Отклик при вводе достаточно быстрый, задержек не наблюдается. Эти результаты показали, что контроллер может работать с полноценной ОС Arch Linux.

4.2 Тестирование программных средств

Задача этого этапа тестирования – проверить работу установленных программных компонентов и правильность выполненной настройки.

Тестирование проводилось на ПК, исполняющем роль сервера в разрабатываемой системе, с установленным программным обеспечением, описание которого представлено в разделе 3.2.1 пояснительной записки.

Из рисунков 12 и 13 видно, что программы Apache и MySQL. запущены и работают нормально.

```
nnn@nnn-HP-Desktop-Pro-300-G6-Microtower-PC: ~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese
  Active: active (running) since Mon 2024-06-17 11:23:17 +07; 12min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 964 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUC
 Main PID: 1161 (apache2)
    Tasks: 6 (limit: 18915)
   Memory: 20.6M
   CGroup: /system.slice/apache2.service
           └─1161 /usr/sbin/apache2 -k start
             └─1181 /usr/sbin/apache2 -k start
               └─1182 /usr/sbin/apache2 -k start
                 └─1183 /usr/sbin/apache2 -k start
                   └─1185 /usr/sbin/apache2 -k start
                     └─1189 /usr/sbin/apache2 -k start

июн 17 11:23:17 nnn-HP-Desktop-Pro-300-G6-Microtower-PC systemd[1]: Starting Th
июн 17 11:23:17 nnn-HP-Desktop-Pro-300-G6-Microtower-PC apachectl[1007]: [Mon J
июн 17 11:23:17 nnn-HP-Desktop-Pro-300-G6-Microtower-PC apachectl[1007]: AH0055
июн 17 11:23:17 nnn-HP-Desktop-Pro-300-G6-Microtower-PC systemd[1]: Started The
lines 1-20/20 (END)
```

Рисунок 12 – Статус работы сервера Apache

```
nnn@nnn-HP-Desktop-Pro-300-G6-Microtower-PC: ~$ systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset:
  Active: active (running) since Mon 2024-06-17 11:23:18 +07; 14min ago
     Docs: https://dev.mysql.com/doc/mysql-systemd-start-pre/8.0/en
   Process: 965 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=ex
 Main PID: 1142 (mysqld)
   Status: "Server is operational"
    Tasks: 37 (limit: 18915)
   Memory: 438.5M
   CGroup: /system.slice/mysql.service
           └─1142 /usr/sbin/mysqld

июн 17 11:23:17 nnn-HP-Desktop-Pro-300-G6-Microtower-PC systemd[1]: Starting My
июн 17 11:23:18 nnn-HP-Desktop-Pro-300-G6-Microtower-PC systemd[1]: Started MyS
lines 1-13/13 (END)
```

Рисунок 13 – Статус работы базы данных MySQL

Чтобы проверить работу PhPMyAdmin, требуется в браузере ввести адрес localhost/phpmyadmin/index.php, ведущий на стандартную страницу программы (Рисунок 14).

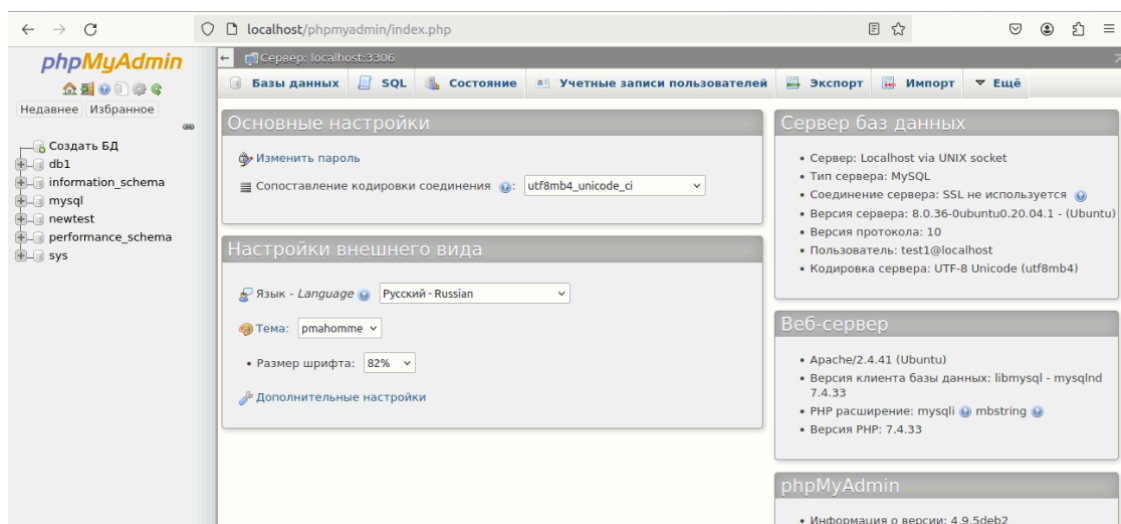


Рисунок 14 – Статус панели PhPMyAdmin и версии установленных модулей

Для записи и хранения данных при помощи СУБД MySQL создана база данных db1 с таблицей sensors. На рисунке 15 представлен список датчиков, добавленных в базу данных. Так как эта БД создана в целях проверки работоспособности сервера, в таблице Sensors имеются только поля name_id, являющееся первичным ключом, и name.

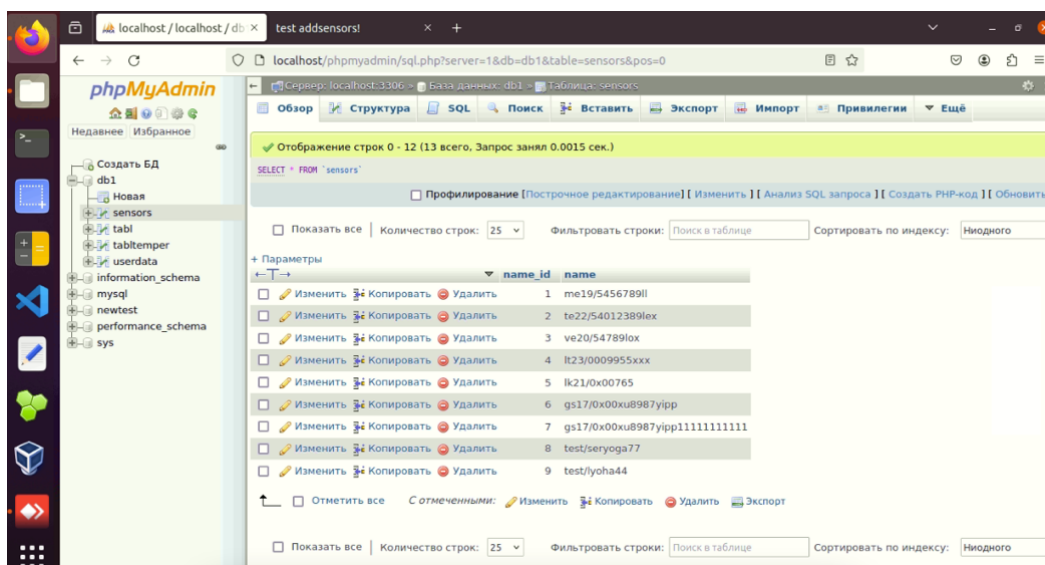


Рисунок 15 – База данных для записи с таблицей sensors

Запись новых данных в БД проводится при помощи ввода информации о добавляемом датчике в специальное поле на HTML-странице (Рисунок 16), и

разработанного PHP-скрипта, который срабатывает при нажатии на кнопку «Добавить!» и обрабатывает запрос.

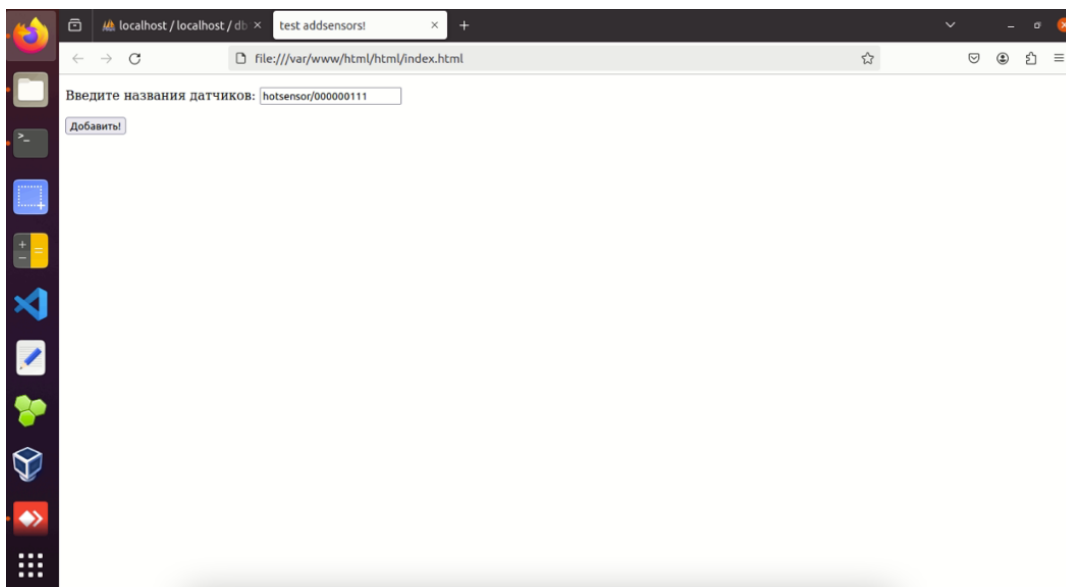


Рисунок 16 – HTML-страница с полем ввода данных

На рисунке 17 можно увидеть, что информация о датчике, введенная в форму (Рисунок 16), успешно добавлена в базу данных. Добавленному датчику присвоено name_id 10.

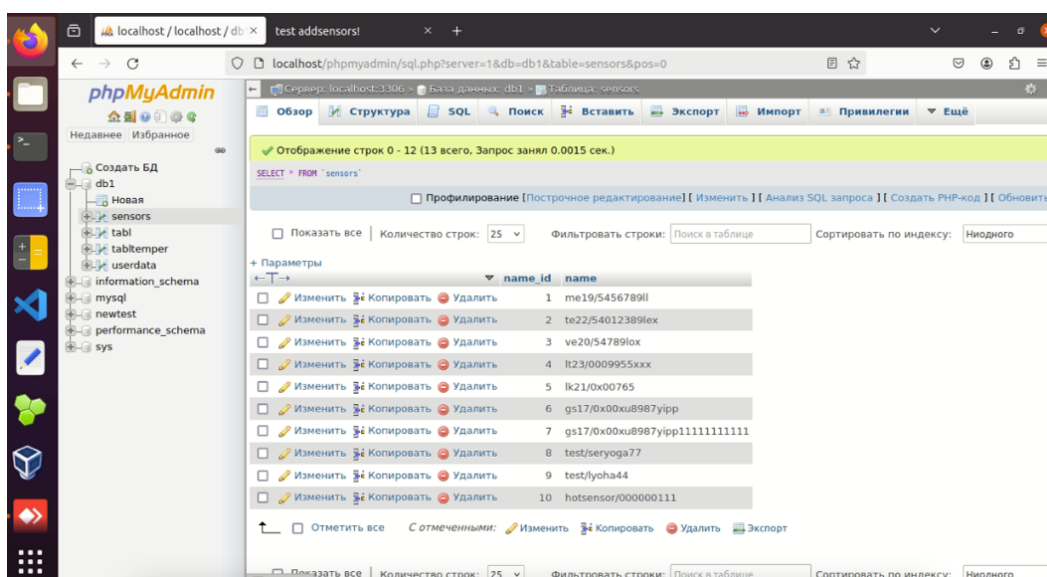


Рисунок 17 – Добавленный в БД новый датчик

Таким образом, проведенное тестирование показывает общую работоспособность сервера, и отдельных программных компонентов, таких как Apache, MySQL и PhpMyAdmin.

Полученные результаты говорят о успешности проведенного тестирования.

4.3 Выводы по главе

Проведено тестирование технических решений, применяемых для контроллера и программных модулей, установленных на сервере.

Тестирование проводилось в два этапа. Эти этапы включают в себя:

1. тестирование установленной на контроллер системы;
2. тестирование программных средств.

На первом этапе тестирования выявлено, что установленная на контроллер ОС работает корректно, все аппаратные модули определяются. Следовательно, установка выбранной ОС Arch Linux является целесообразным решением.

На втором этапе тестирования выявлено, что установленные программные модули, а именно сервер Apache, панель управления PhpMyAdmin и СУБД MySQL работают корректно.

Проведенное тестирование показало положительные результаты. Таким образом:

1. Контроллер подготовлен к дальнейшей установке необходимого программного обеспечения, в том числе драйверов подключаемых устройств, вспомогательного ПО и программных модулей, описанных в главе 2 пояснительной записки.

2. На сервере создана необходимая программная структура для дальнейшей разработки интерфейса оператора, протоколов взаимодействия с контроллером и пользовательскими устройствами, и наполнения базы данных.

Таким образом, проделанная работа является основой для дальнейшей работы над проектом наземной части комплекса архитектуры системы передачи телеметрической информации.

ЗАКЛЮЧЕНИЕ

Во время реализации данного проекта поэтапно решались задачи, поставленные в начале работы над магистерской диссертацией.

На первом этапе работы проведён обзор предметной области исследования. Произведено сравнение различных систем передачи телеметрической информации, и определены критерии сравнения выбранных систем. В ходе сравнения выявлено, что все представленные системы не имеют возможности регулировать передаваемый и принимаемый трафик, что для разрабатываемой системы передачи телеметрической информации является критичным. Возможность создания сложных сценариев имеется только в системе от компании Xiaomi. Системы Умный дом от Яндекса и Xiaomi Smart Home нуждаются в постоянном подключении к интернету, что снижает их автономность. Полученные результаты позволили сформировать перечень требований к разрабатываемой системе.

На втором этапе разработки программной структуры системы передачи телеметрической информации определены основные составляющие наземного программно-аппаратного комплекса: сервер, контроллер, пользовательское устройство. Принято решение о переносе логического центра системы с сервера на контроллер, что отличает разрабатываемую систему от аналогов, а также позволяет минимизировать передаваемый между контроллером и сервером трафик, что является важным при использовании спутникового канала связи. На основе требований, предъявляемых к системе, предложена программная архитектура системы, которая включает в себя ПО сервера, ПО контроллера, а также мобильное приложение. Разработаны технические требования к программному обеспечению сервера, контроллера, пользовательских устройств. Описаны основные программные модули, которые должны быть разработаны для реализации основных функций разрабатываемой системы. Разработанная архитектура и технические требования позволили перейти к разработке технических решений и программных модулей.

На третьем этапе выбраны программные средства для работы с сервером в системе передачи телеметрической информации. Произведена установка и настройка серверного ПО Apache, MySQL, PhpMyAdmin, при помощи программы LAMP. Выбрано аппаратное обеспечение, которое может использоваться в качестве контроллера передачи телеметрической информации. Проведены исследования операционных систем Android и Arch Linux с целью определения подходящей для установки на контроллер. Проведено исследование схемотехники контроллера, проведено подключение USB UART донгла. Для контроллера разработан загрузчик, позволяющий загружать его с карты памяти. Выполнена загрузка выбранной операционной системы на контроллер. Полученные результаты позволили перейти к тестированию технических решений и программных средств.

На заключительном этапе работы проведено тестирование установленной на контроллер ОС Arch Linux, а также тестирование программного обеспечения, установленного на сервер. Тестирование показало положительные результаты. Таким образом, реализована минимальная программная структура, позволяющая организовать дальнейшую работу по разработке системы передачи телеметрической информации и расширению её функционала.

Таким образом, все задачи, которые были поставлены на первом этапе выполнения ВКР, выполнены, вследствие чего можно сделать вывод о достижении цели работы.

СПИСОК СОКРАЩЕНИЙ

БД	– База данных
МП	– Мобильное приложение
ОЗУ	– Оперативное запоминающее устройство
ОС	– Операционная система
ПЗУ	– Постоянное запоминающее устройство
ПК	– Персональный компьютер
ПО	– Программное обеспечение
УЗП	– Учетная запись пользователя
ARM	– Advanced RISC Machine
GND	– Ground
HDMI	– High Definition Multimedia Interface
HTTP	– HyperText Transfer Protocol
ID	– Identifier
LAN	– Local area network
LAMP	– Linux, Apache, MySQL, PHP
OTA	– Over-the-air
PHP	– PHP: Hypertext Preprocessor
RX	– Receive
S/N	– Serial number
SD	– Secure Digital Memory Card
SMS	– Short Message Service
SQL	– Structured Query Language
TX	– Transmit
UART	– Universal Asynchronous Receiver
USB	– Universal Serial Bus

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. ГОСТ 19619-74 Оборудование радиотелеметрическое. Термины и определения. – URL: <https://docs.cntd.ru/document/1200015682> (дата обращения 06.07.2023).
2. Умный дом с Алисой – комфортный и безопасный. – URL: <https://alice.yandex.ru/smart-home> (дата обращения 08.07.2023).
3. Как настроить умный дом Яндекса с Алисой. – URL: <https://journal.tinkoff.ru/guide/yandex-smart-home/> (дата обращения 08.07.2023).
4. Мобильное приложение «Дом с Алисой». – URL: <https://apps.apple.com/ru/app/%D0%B4%D0%BE%D0%BC-%D1%81-%D0%B0%D0%BB%D0%B8%D1%81%D0%BE%D0%B9/id1582810683> (дата обращения 10.07.2023).
5. Тысячи совместимых устройств для умного дома с Алисой. – URL: <https://alice.yandex.ru/smart-home/devices> (дата обращения 10.07.2023).
6. «Умный дом» «Яндекса»: сценарии и примеры. – URL: <https://xn--90aifdm6al.xn--p1ai/blog/umnyj-dom-yandeksa-scenarii-i-primery> (дата обращения 10.10.2023).
7. Сценарии в Умном доме Яндекса. – URL: <https://yandex.ru/alice/support/smart-home/scenarios/create.html> (дата обращения 10.10.2023).
8. Raspberry pi 4. – URL: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/> (дата обращения 01.08.2023).
9. Home assistant. – URL: <https://www.home-assistant.io/> (дата обращения 06.08.2023).
10. Первое знакомство с Home Assistant. – URL: <https://habr.com/ru/articles/471822/> (дата обращения 06.08.2023).
11. Home assistant operating system. – URL: <https://www.home-assistant.io/installation/> (дата обращения 06.08.2023).

12. Что такое Home Assistant — как настроить умный дом без подключения к облаку. – URL: <https://journal.tinkoff.ru/guide/local-smart-home/> (дата обращения 06.08.2023).
13. Что такое Home Assistant? Обзор Приложения для Организации Умного Дома. – URL: <https://notify.events/ru/blog/home-assistant-review> (дата обращения 06.08.2023).
14. Четыре года с умным домом: личный опыт и интересные сценарии. – URL: <https://habr.com/ru/companies/banki/articles/765804/> (дата обращения 10.11.2023).
15. Persistent notification. – URL: https://www.home-assistant.io/integrations/persistent_notification/ (дата обращения 07.08.2023).
16. Notifications. – URL: <https://www.home-assistant.io/integrations/notify/> (дата обращения 07.08.2023).
17. Умный дом Xiaomi Smart Home : сайт. – URL: <https://xiaomi-smarhome.ru/> (дата обращения 10.09.2023).
18. Мобильное приложение Xiaomi MiHome : сайт. – URL: <https://xiaomi-smarhome.ru/xiaomi-mi-home/> (дата обращения 10.09.2023).
19. Mi Home настройка сценариев автоматизации. – URL: <https://xiaomi-smarhome.ru/mi-home-nastrojka-stsenarijev-avtomatizatsii/> (дата обращения 10.09.2023).
20. Спутниковый терминал «Скиф». – URL: <https://matrixwave.ru/sputnikoviy-terminal-skif> (дата обращения 10.11.2023).
21. Спутниковая система «Гонец». – URL: <https://gonets.ru/rus/> (дата обращения 10.11.2023).
22. PHP : сайт. – URL: <https://www.php.net/> (дата обращения 30.04.2024).
23. Установка LAMP. – URL: <https://help.ubuntu.ru/wiki/lamp> (дата обращения 01.05.2024).
24. Apache : сайт. – URL: <https://httpd.apache.org/> (дата обращения 07.05.2024).

25. База данных MySQL : сайт. – URL: <https://www.mysql.com/> (дата обращения 07.05.2024).
26. Ubuntu : сайт. – URL: <https://ubuntu.com/download/desktop> (дата обращения 06.07.2023).
27. PhpMyAdmin : сайт. – URL: <https://www.phpmyadmin.net/> (дата обращения 06.05.2024).
28. Медиаплеер Hyundai H-DMP100 технические характеристики. – URL: <https://www.citilink.ru/product/mediapleer-hyundai-h-dmp100-1385766/> (дата обращения 07.08.2023).
29. Termux : сайт. – URL: <https://termux.dev/en/> (дата обращения: 20.10.2023).
30. Что такое загрузчик в Android и как его разблокировать. – URL: <https://androidinsider.ru/smartfony/что-такое-zagruzchik-v-android-i-kak-ego-razblokirovat.html> (дата обращения 25.05.2024).
31. PuTTY : сайт. – URL: <https://www.putty.org/> (дата обращения: 20.10.2023).
32. СТО 7.5–07–2021 «Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности» (дата обращения 16.06.2024).

ПРИЛОЖЕНИЕ А

Листинги разработанных программ

Index.html

```
<html>
  <head>
    <title>test addsensors!</title>
    <meta charset="utf-8" />
  </head>
  <body>
<form action="addsensors.php" method="post">

  <p>Введите названия датчиков:
    <input type="text" name="name1"
      test:<?=$name1?>
    </p>

    <input type="submit" value="Добавить!">

  </body>
</html>
```

Addsensors.php

```
<?php
if (isset($_POST["name1"])) {

  $conn = mysqli_connect("localhost", "root", "]e]Yfv1809", "db1");
  if (!$conn) {
    die("Ошибка: " . mysqli_connect_error());
  }

  $name = mysqli_real_escape_string($conn, $_POST["name1"]);
  $sql = "INSERT INTO sensors (name) VALUES ('$name')";
  if(mysqli_query($conn, $sql)){
    echo "Данные успешно добавлены";
  } else{
    echo "Ошибка: " . mysqli_error($conn);
  }
  mysqli_close($conn);
}
?>
```

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ
Заведующий кафедрой
О.В. Непомнящий
« 20 » 06 2024 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Проектирование программной архитектуры системы передачи
телеметрической информации

090401 Информатика и вычислительная техника

09.04.01.11 «Вычислительные системы и сети»

Руководитель	 подпись	19.06.24 дата	профессор, канд. техн. наук должность, ученая степень	О.В. Непомнящий
Выпускник	 подпись	19.06.24 дата		Н.В. Коптяев
Рецензент	 подпись	19.06.24 дата	доцент, канд. техн. наук должность, ученая степень	К.В. Раевич
Консультант	 подпись	20.06.24 дата	генеральный директор ООО «ПК «Дельта»» должность, ученая степень	О.Г. Варыгин
Нормоконтролёр	 подпись	19.06.24 дата	профессор, канд. техн. наук должность, ученая степень	О.В. Непомнящий

Красноярск 2024