СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 6](#_Toc485602205)

[1 АНАЛИЗ ИСХОДНЫХ ДАННЫХ И ПОСТАНОВКА ЗАДАЧИ 7](#_Toc485602206)

[1.1 Постановка задачи проектирования 7](#_Toc485602207)

[1.2 Описание предметной области 7](#_Toc485602208)

[1.3 Анализ аналогов и прототипов 7](#_Toc485602209)

[2 РАЗРАБОТКА АЛГОРИТМОВ РЕШЕНИЯ ЗАДАЧ 8](#_Toc485602210)

[2.1 Функциональная структура 8](#_Toc485602211)

[2.2 Выбор и обоснование инструментальных средств разработки 8](#_Toc485602212)

[2.3 Структура объекта разработки 8](#_Toc485602213)

[2.3.1 Программная структура 8](#_Toc485602214)

[2.3.2 Структура ресурсов приложения 8](#_Toc485602215)

[2.4 Создание уровней приложения 8](#_Toc485602216)

[2.5 Взаимодействие пользователя с мобильным приложением 8](#_Toc485602217)

[2.6 Разработка пользовательского интерфейса для динамических уровней приложения 9](#_Toc485602218)

[3 РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ 9](#_Toc485602219)

[3.1 Реализация функциональных частей приложения 9](#_Toc485602220)

[3.1.1 Реализация работы с ресурсами приложения 9](#_Toc485602221)

[3.1.2 Реализация формирования и переключения экранов 9](#_Toc485602222)

[3.2 Реализация игровой части приложения 9](#_Toc485602223)

[3.2.1 Создание игрового уровня 9](#_Toc485602224)

[3.2.2 Описание основных механик игры 9](#_Toc485602225)

[3.3 Тестирование и отладка 9](#_Toc485602226)

[4 ЭКОНОМИЧЕСКАЯ ЧАСТЬ 10](#_Toc485602227)

[4.1 Обоснование необходимости выведения продукта на рынок 10](#_Toc485602228)

[4.2 Структура работ по созданию программного обеспечения 10](#_Toc485602229)

[4.3 Составление сметы затрат на разработку программного обеспечения 10](#_Toc485602230)

[4.4 Обоснование эффективности разработки программного средства 10](#_Toc485602231)

[ЗАКЛЮЧЕНИЕ 11](#_Toc485602232)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 12](#_Toc485602233)

[ПРИЛОЖЕНИЯ 13](#_Toc485602234)

# ВВЕДЕНИЕ

Электронный магазин — это веб-ресурс . Подобно обычному магазину, электронный магазин реализует следующие основные функции: представление товаров (услуг) покупателю, обработку заказов, продажу и доставку товаров.

Интернет-магазин объединяет элементы прямого маркетинга с образом посещения традиционного магазина. Отличительной чертой интернет-магазинов по сравнению с обычной формой торговли является то, что интерактивный магазин может предложить значительно большее количество товаров и услуг, и обеспечить потребителей значительно большим объемом информации, необходимым для принятия решения о покупке. Кроме того, за счет использования компьютерных технологий возможна персонализация подхода к каждому из клиентов, исходя из истории его посещений магазина и сделанных ранее покупок.

Основные проблемы реализации интернет-магазина лежат на стыке технологий Интернета и традиционной коммерческой деятельности. В обычной торговле покупатель привык к тому, что есть возможность оценить товар визуально, определить его качество и характеристики. В электронной торговле он такой возможности лишен. Зачастую визуальной информации достаточно, но здесь вступают в действие эмоциональные и психологические факторы. Часто существуют проблемы с доставкой товаров, особенно если их цена невелика.

По последним данным аудитория в интернете стремительно растет, а продажи через интернет в крупных городах достигают до 25%, при этом специалисты подчеркивают тенденцию к росту продаж именно через интернет. Ежегодно количество интернет-магазинов увеличивается, так как это действительно прибыльно и удобно для покупателя, не говоря о экономии бюджета и времени. Интернет-магазин работает круглые сутки и может продавать определенные товары в автоматическом режиме без участия продавца. К преимуществам так же можно отнести то, что не надо закупать товар заранее, а это существенная экономия, на складских помещениях. Нужно лишь договориться с поставщиками, и в нужный момент, просто выкупить товар, который у вас закажут. По сравнению с обычным магазином, территория продаж которого ограничивается населением города или района, территория охвата интернет-магазина увеличивается на всю Россию и русскоязычную аудиторию в других странах, ведь товар можно доставлять не только курьерской службой, но и почтой.

Целью данной дипломной работы является разработка интернет-магазина по продаже свадебной и вечерней бижутерии. Данная система будет работать в рамках определенного предприятия и выполнять функции основной площадки для ведения коммерческой деятельности.

Что касается продажи свадебной бижутерии, то можно отметить, что эта ниша еще не достаточно заполнена и потому конкурентоспособность мероприятия по созданию онлайн площадки по продаже именно этой продукции достаточно высока. Кроме того, данная продукция интересна как для женской, так и для мужской половины человечества, что только подчеркивает актуальность нашего магазина.

АНАЛИЗ ИСХОДНЫХ ДАННЫХ И ПОСТАНОВКА ЗАДАЧИ

## Описание предметной области

При перевозке некоторых грузов, к примеру продуктов питания, лекарств, химических реактивов и др., требуется соблюдать температурный режим. Контролировать температуру хочет владелец груза – чтобы убедиться в его сохранности во время доставки. Это выгодно грузоперевозчику – дополнительный контроль снижает цену страхования груза и помогает при разборе нештатных ситуаций.

На рынке существует множество систем мониторинга. Чтобы конкурировать с другими решениями, нужно максимально упростить работу пользователя с устройством.

Предполагается что для работы пользователю потребуется устройство на платформе Android и приложение. Приложение должно взаимодействовать с устройством без проводов и интернета. Для этого можно использовать технологию Bluetooth.

В основе устройства – печатная плата с микроконтроллером. Микроконтроллер действует по инструкциям прошивки, которая загружается при помощи программатора.

Android – операционная система для смартфонов, планшетов, электронных книг, цифровых проигрывателей, наручных часов, фитнес-браслетов, игровых приставок, ноутбуков, нетбуков, смартбуков, очков Google Glass, телевизоров и других устройств (в 2015 году появилась поддержка автомобильных развлекательных систем и бытовых роботов).

Мобильное приложение (англ. «Mobile app») – программное обеспечение, предназначенное для работы на смартфонах, планшетах и других мобильных устройствах. Многие мобильные приложения предустановлены на самом устройстве или могут быть загружены на него из онлайновых магазинов приложений, таких как App Store, BlackBerry App World, Google Play, 1mobile market, Windows Phone Store, Яндекс.store и других, бесплатно или за плату.

Bluetooth, блюту́с – производственная спецификация беспроводных персональных сетей (Wireless personal area network, WPAN). Bluetooth обеспечивает обмен информацией между такими устройствами, как персональные компьютеры (настольные, карманные, ноутбуки), мобильные телефоны, интернет-планшеты, принтеры, цифровые фотоаппараты, мышки, клавиатуры, джойстики, наушники, гарнитуры и акустических систем на надёжной, бесплатной, повсеместно доступной радиочастоте для ближней связи. Bluetooth позволяет этим устройствам сообщаться, когда они находятся в радиусе до 10 м друг от друга (дальность сильно зависит от преград и помех), даже в разных помещениях.

Печа́тная пла́та (англ. printed circuit board, PCB, или printed wiring board, PWB) – пластина из диэлектрика, на поверхности и/или в объёме которой сформированы электропроводящие цепи электронной схемы. Печатная плата предназначена для электрического и механического соединения различных электронных компонентов. Электронные компоненты на печатной плате соединяются своими выводами с элементами проводящего рисунка обычно пайкой.

Микроконтро́ллер (англ. Micro Controller Unit, MCU) – микросхема, предназначенная для управления электронными устройствами.

Типичный микроконтроллер сочетает на одном кристалле функции процессора и периферийных устройств, содержит ОЗУ и (или) ПЗУ. По сути, это однокристальный компьютер, способный выполнять относительно простые задачи.

Отличается от микропроцессора интегрированными в микросхему устройствами ввода-вывода, таймерами и другими периферийными устройствами.

Прошивкой (англ. firmware, fw) называют содержимое энергонезависимой памяти компьютера или любого цифрового вычислительного устройства – микрокалькулятора, сотового телефона, GPS-навигатора и т. д., в которой содержится его программа.

Словом «прошивка» иногда называют образ ПЗУ, предназначенный для записи в память соответствующего устройства с целью обновления его микропрограммы, а также собственно процесс записи этого образа в энергонезависимую память устройства.

Программа́тор – аппаратно-программное устройство, предназначенное для записи/считывания информации в постоянное запоминающее устройство (однократно записываемое, флеш-память, ПЗУ, внутреннюю память микроконтроллеров и ПЛК).

## Постановка задачи проектирования

Тема дипломного проекта – беспроводной датчик температуры.

Необходимо спроектировать электрическую схему и печатную плату датчика температуры.

Устройство должно работать как от аккумулятора, так и от внешнего источника питания. При наличии аккумулятора и внешнего источника питания, аккумулятор должен заряжаться.

Для работы датчика нужно написать прошивку, которая будет осуществлять измерение температуры и передачу её пользователю.

Помимо самого датчика необходимо разработать приложение для приема и отображения температуры. Оно должно работать на мобильном устройстве на платформе android.

Для передачи данных необходимо использовать Bluetooth, т.к. он есть в большинстве современных мобильных устройств.

## Анализ аналогов и прототипов

Ближайшими аналогами являются устройства фирм Digital Comtech и Samsara. От разрабатываемого устройства они отличаются необходимостью использования проводов для подключения датчика (Digital Comtech), а так же платной подпиской для использования приложения (Samsara).

## Анализ требований к проекту

Для обработки данных о температуре и передачи их на мобильное устройство можно использовать микроконтроллер с Bluetooth.

Измерение температуры будет производиться с помощью цифрового датчика температуры. Цифровой датчик содержит внутри аналоговую схему для измерения сигнала и цепь преобразования аналогового сигнала в цифровой.

В качестве аккумулятора будет использоваться распространенный форм-фактор литий-ионных батарей – 18650. Такие аккумуляторы часто используются в портативных устройствах и их легко найти в продаже. Для данной химии аккумуляторов существуют специальные микросхемы зарядки и защиты, они так же будут задействованы в разрабатываемом устройстве.

Принимать данные от микроконтроллера будет мобильное устройство на платформе Android. Необходимо разработать удобное приложение, которое будет отображать информацию с датчиков и хранить историю измерений.

## Выбор среды и средств разработки

Для разработки электронной части устройства будет использован Altium Designer. Это система, позволяющая реализовывать проекты электронных средств на уровне схемы с последующей передачей информации проектировщику печатной платы. Отличительной особенностью программы является проектная структура и сквозная целостность ведения разработки на разных уровнях проектирования. Иными словами, изменения в разработке на уровне платы могут мгновенно быть переданы на уровень схемы и так же обратно. Так же в качестве приоритетного направления разработчиков данной программы стоит отметить интеграцию ECAD и MCAD систем. Теперь разработка печатной платы возможна в трёхмерном виде с двунаправленной передачей информации в механические САПР (Solid Works, Pro/ENGINEER, NX и др.). В состав программного комплекса Altium Designer входит весь необходимый инструментарий для разработки, редактирования и отладки проектов на базе электрических схем и ПЛИС. Редактор схем позволяет вводить многоиерархические и многоканальные схемы любой сложности, а также проводить смешанное цифро-аналоговое моделирование. Библиотеки программы содержат более 90 тысяч готовых компонентов, у многих из которых имеются модели посадочных мест, SPICE и IBIS-модели, а также трёхмерные модели. Любую из вышеперечисленных моделей можно создать внутренними средствами программы.

Для разработки прошивки микроконтроллера будет использована Segger Embedded Studio. Это комплексное решение «все в одном» для управления, создания, тестирования и развертывания встроенных приложений: от генератора проектов, с помощью которого можно легко приступить к работе с обычными микроконтроллерами ARM, до мощного диспетчера проектов и редактора исходного кода, включенного в комплект поставки. Компилятор C/C++ и интегрированный отладчик с расширенными окнами отладочной информации и прямой интеграцией с программатором J-Link, вплоть до функций контроля версий для автоматического развертывания ваших приложений.

Segger Embedded Studio использует стиль, аналогичный Microsoft Visual Studio. Это интуитивно понятное использование, с которым разработчики ПК знакомы в мире встраиваемых технологий.

Для разработки мобильного приложения будет использована Android Studio. Это интегрированная среда разработки (IDE) для работы с платформой Android. Android Studio, основанная на программном обеспечении IntelliJ IDEA от компании JetBrains, - официальное средство разработки Android приложений. Данная среда разработки доступна для Windows, OS X и Linux. 17 мая 2017, на ежегодной конференции Google I/O, Google анонсировал поддержку языка Kotlin, используемого в Android Studio, как официального языка программирования для платформы Android в добавление к Java и С++.

## Разработка технического задания

На основании анализа предметной области были выделена следующая функциональность устройства:

измерение температуры;

передача данных о температуре по Bluetooth;

работа от аккумулятора;

работа от внешнего источника питания;

зарядка аккумулятора.

На основании анализа предметной области были выделена следующая функциональность приложения:

работа на платформе Android;

получение данных о температуре по Bluetooth;

отображение текущих данных;

сохранение истории в базу данных;

просмотр истории измерений.

ПРОЕКТИРОВАНИЕ АППАРАТНОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ



## Функциональная структура печатной платы

На основе приведённых характеристик и требований технического задания (приложение А) была разработана функциональная структура для платы датчика температуры. Её можно увидеть на рисунке 2.1 и приложении Е.



Рисунок 2.1 – функциональная структура платы датчика температуры

Рассмотрим подробнее роль каждого блока.

Конструкция аккумулятора и корпуса не содержат механической защиты для предотвращения неправильной установки. Защита от неправильной установки аккумулятора позволяет избежать повреждения схемы при переполюсовке аккумулятора.

Планируется использовать стандартный литий-ионный аккумулятор типоразмера 18650. Литий-ионные аккумуляторы очень чувствительны к перезарядке и переразрядке. В таких ситуациях аккумулятор быстро изнашивается. Для предотвращения будет использована защита от переразрядки и перезарядки.

Для удобства использования устройства, в него будет встроена зарядка аккумулятора. Помимо зарядки, эта часть схемы позволит питать устройство от внешнего источника питания при отсутствии аккумулятора. Микросхема зарядки будет сообщать микроконтроллеру о подключении внешнего питания и о зарядке аккумулятора с помощью логических выводов.

Для измерения температуры будет использован цифровой сенсор. Информация о температуре будет передаваться на микроконтроллер по линии данных.

## Функциональная структура прошивки микроконтроллера



Рисунок 2.2 – функциональная структура прошивки микроконтроллера

Рассмотрим подробнее роль каждого блока.

Модуль контроля статусов зарядки считывает информацию с логических входов микроконтроллера, которые подключены к логическим выводам микросхемы зарядки. Внешние прерывания настраиваются чтобы не опрашивать выводы постоянно, а выполнять функцию-обработчик только при изменении состояния. Текущее состояние можно прочитать из модуля в любой момент.

Модуль измерения заряда аккумулятора настраивает встроенный аналогово-цифровой преобразователь на вход питания. Микроконтроллер запитан напрямую от аккумулятора, напряжение питания равно напряжению на аккумуляторе. Напряжение регулярно измеряется и конвертируется в процент заряда. При малом потреблении, напряжение на литий-ионном аккумуляторе стабильно и зависимость напряжения от оставшейся емкости можно принять за линейную.

Датчик температуры передает данные по интерфейсу 1-wire. В SDK микроконтроллера нет готовой реализации драйвера 1-wire. При разработке драйвера принято разделять логические уровни. Сначала реализуются функции передачи информации, они используются в функциях отправки команд. На следующем уровне реализуется работа с датчиком, отправка необходимых команд, чтение ответов и конвертация полученных результатов.

Модуль передачи данных собирает всю необходимую информацию в структуру данных и отправляет по Bluetooth. Функции работы с Bluetooth реализованы в SDK микроконтроллера.

## Функциональная структура мобильного приложения

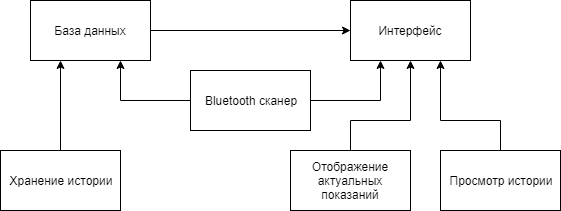


Рисунок 2.3 – функциональная структура мобильного приложения

Рассмотрим подробнее основные модули мобильного приложения.

Bluetooth сканер обрабатывает получаемые пакеты, отсеивая информацию от датчиков среди остальных устройств. Происходит парсинг полученных пакетов в объекты и передача их в базу данных и в пользовательский интерфейс.

База данных хранит историю измерений.

В интерфейсе можно увидеть последние измерения и историю измерений за указанный период.

## Диаграмма развертывания

Диагра́мма развёртывания (англ. Deployment diagram) в UML моделирует физическое развертывание артефактов на узлах.[1] Например, чтобы описать веб-сайт, диаграмма развертывания должна показывать, какие аппаратные компоненты («узлы») существуют (например, веб-сервер, сервер базы данных, сервер приложения), какие программные компоненты («артефакты») работают на каждом узле (например, веб-приложение, база данных), и как различные части этого комплекса соединяются друг с другом (например, JDBC, REST, RMI).

Диаграмма развертывания разрабатываемого проекта показана на рисунке 2.4.

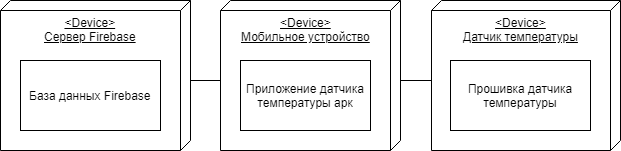


Рисунок 2.4 – диаграмма развертывания

## Проектирование пользовательского интерфейса

С помощью графического редактора был разработан макет пользовательского интерфейса, показанный на рисунке 2.5.

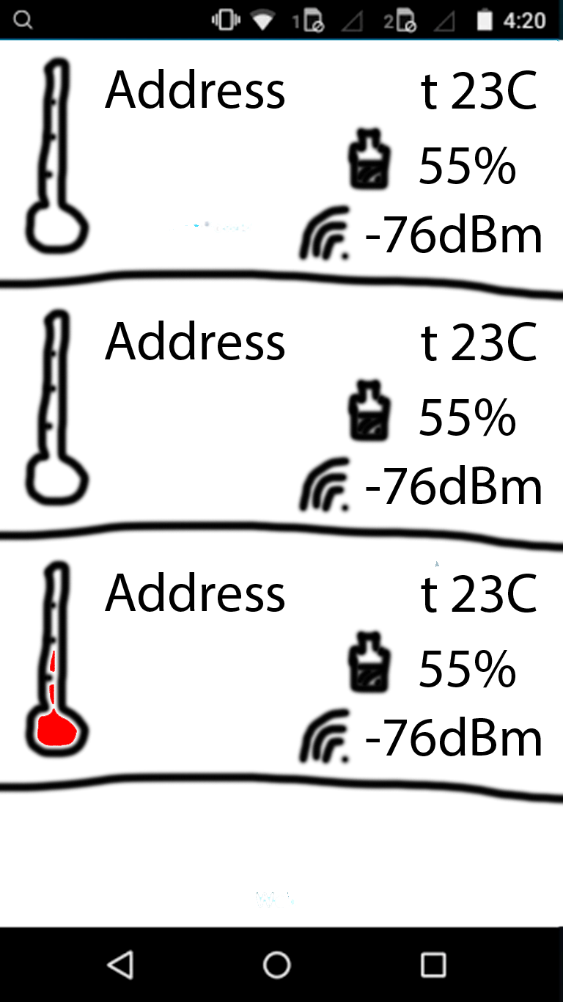


Рисунок 2.5 – Макет пользовательского интерфейса

РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ

## Реализация электрической схемы

На основании требований были подобраны компоненты устройства. Итоговые схемы представлены в приложении Д.

Для зарядки была выбрана микросхема BQ24040DSQT от производителя Texas Instruments. Её характеристики представлены в таблице 3.1. Помимо представленных в таблице характеристик, эта микросхема поддерживает профиль температуры JEITA. Это означает, что микросхема не будет заряжать аккумулятор за пределами диапазона его рабочей температуры, а при приближении к границе диапазона будет снижать ток зарядки.

Таблица 3.1 – характеристики микросхемы зарядки

|  |  |
| --- | --- |
| Тип аккумуляторов | Литий-ионные, литий-полимерные |
| Максимальный ток заряда, А | 1 |
| Входное напряжение, В | 5 |
| Регулировка тока заряда | Присутствует |
| Контроль температуры аккумулятора | С помощью внешнего термистора |
| Индикация | Присутствия питания, статуса зарядки |
| Работа без аккумулятора | Возможна |
| Защита | От перегрева, короткого замыкания |

Для защиты входа микросхемы зарядки от переполюсовки был использован диод Шоттки. Так же на вход был установлен защитный диод (супрессор) – это диод, позволяющий ограничить импульсные всплески перенапряжения, которые возникают от внешних воздействий, таких как: электростатические разряды (ESD), грозовые разряды, подключение индуктивной нагрузки и др.

Пассивные компоненты были подключены согласно технической спецификации на микросхему зарядки. К выводам, отвечающим за индикацию присутствия питания и статуса зарядки, были подключены светодиодные индикаторы, так же эти выводы подключены к микроконтроллеру. Схема зарядки представлена под номером 2.

Для защиты устройства от переполюсовки (неправильной установки) аккумулятора были применены два транзистора. Биполярный транзистор с PNP переходом управляет N-канальным MOSFET-ом, который разрывает цепь. Такая схема позволяет запитывать устройство и заряжать аккумулятор.

MOSFET (metal–oxide–semiconductor field-effect transistor) – полевой транзистор с изолированным затвором (МДП – транзистор), затвор которого отделён от канала тонким слоем диэлектрика (обычно двуокись кремния SiO2). MOSFET – служит в основном, как усилитель сигнала, либо как выпрямитель, для понижения напряжения до нужного значения.

Биполярный транзистор – трёхэлектродный полупроводниковый прибор, один из типов транзисторов. В полупроводниковой структуре сформированы два p-n-перехода, перенос заряда через которые осуществляется носителями двух полярностей – электронами и дырками. Именно поэтому прибор получил название «биполярный» (от англ. bipolar), в отличие от полевого (униполярного) транзистора. Применяется в электронных устройствах для усиления или генерации электрических колебаний, а также в качестве коммутирующего элемента (например, в схемах транзисторно-транзисторной логики).

Для защиты аккумулятора от переразрядки и перезарядки была выбрана микросхема BQ29700DSER от производителя Texas Instruments. Эта микросхема управляет сдвоенным N-канальным MOSFET-ом. С его помощью она отключает нагрузку или зарядку от аккумулятора. Эта микросхема измеряет напряжение на аккумуляторе и ток заряда/разряда. Если измеряемые величины выходят из допустимых пределов, микросхема выключает транзистор. Таким образом аккумулятор защищен от короткого замыкания, перезаряда и переразряда. Пассивные компоненты подключены согласно технической спецификации на микросхему защиты. Схема зарядки и защиты представлена под номером 3.

В качестве микроконтроллера был выбран Nordic nRF52840. Его характеристики представлены в таблице 3.2.

Таблица 3.2 – характеристики микроконтроллера nRF52840

|  |  |
| --- | --- |
| Bluetooth | Версия 5 |
| Разрядность, бит | 32 |
| Частота, МГц | 64 |
| Напряжение питания, В | 1,7 … 5,5 |
| Интерфейсы | SPI, UART, PWM, USB и др. |
| Разрядность аналогово-цифрового преобразователя, бит | 12 |
| Встроенный DC-DC преобразователь | Присутствует |
| Выходной ток встроенного преобразователя, мА | 25 |

К микроконтроллеру подключен датчик температуры и индикация статусов зарядки. Остальные компоненты подключены согласно технической спецификации, это показано на схеме номер 4.

Для измерения температуры был выбран цифровой датчик температуры DS18B20 от производителя Maxim Integrated. Характеристики датчика представлены в таблице 3.3.

Таблица 3.3 – характеристики датчика температуры DS18B20

|  |  |
| --- | --- |
| Напряжение питания, В | 3,0 … 5,5 |
| Диапазон измеряемых температур, °С | -55 ... +125 |
| Погрешность измерений, °С | ±0,5 (при t ∈ {-10°C … +85°C}  ±1 (при t ∈ {-30°C … +100°C}  ±2 (при t ∈ {-55°C … +125°C} |
| Потребление тока в режиме ожидания, нА | 750 … 1000 |
| Потребление тока во время измерения, мА | 1 … 1,5 |
| Разрешение, бит | 9 … 12 (задается программно) |
| Интерфейс передачи данных | 1-wire |

Разъем подключения и пассивные компоненты, необходимые для работы датчика показаны на схеме номер 5.

Соединение всех блоков показано на схеме номер 1. Так же на этой схеме изображены разъемы питания, контакты для программирования и делитель напряжения, который подключен к аналоговому входу микроконтроллера для измерения напряжения на аккумуляторе.

## Реализация прошивки микроконтроллера

Для реализации прошивки будем использовать язык Си и SDK от производителя микроконтроллера.

SDK (от англ. software development kit) – набор средств разработки, который позволяет специалистам по программному обеспечению создавать приложения для определённого пакета программ, программного обеспечения базовых средств разработки, аппаратной платформы, компьютерной системы, игровых консолей, операционных систем и прочих платформ.

SDK для микроконтроллера nRF52840 позволяет управлять всей периферией и радио-частью. Помимо библиотек SDK предоставляет примеры кода и инструкции для основных вариантов использования.

Функциональная структура прошивки показана в пункте 2.2.

Рассмотрим подробнее реализацию драйвера датчика температуры. Протокол передачи данных описан в технической документации датчика температуры.

Листинг 3.1 – функции записи и чтения бита

static inline void writeBit(bool bit) {

uint32\_t writeDelay = bit ? DELAY\_US\_WRITE\_1 : DELAY\_US\_WRITE\_0;

nrf\_gpio\_cfg\_output(DS\_PIN);

nrf\_gpio\_pin\_write(DS\_PIN, 0);

nrf\_delay\_us(writeDelay);

nrf\_gpio\_cfg\_input(DS\_PIN, NRF\_GPIO\_PIN\_NOPULL);

nrf\_delay\_us(DELAY\_US\_TIME\_SLOT - writeDelay);

}

static inline bool readBit() {

nrf\_gpio\_cfg\_output(DS\_PIN);

nrf\_gpio\_pin\_write(DS\_PIN, 0);

nrf\_delay\_us(DELAY\_US\_READ\_DATA);

nrf\_gpio\_cfg\_input(DS\_PIN, NRF\_GPIO\_PIN\_NOPULL);

nrf\_delay\_us(DELAY\_US\_READ\_DATA\_VALID);

bool result = (bool)nrf\_gpio\_pin\_read(DS\_PIN);

nrf\_delay\_us(DELAY\_US\_TIME\_SLOT - DELAY\_US\_READ\_DATA - DELAY\_US\_READ\_DATA\_VALID);

return result;

}

В листинге 3.1 представлены функции bool readBit() и void writeBit(bool bit). В них использованы библиотечные функции для управления GPIO и для приостановки выполнения программы.

Интерфейс ввода/вывода общего назначения (англ. general-purpose input/output, GPIO) – интерфейс для связи между компонентами компьютерной системы, к примеру микропроцессором и различными периферийными устройствами. Контакты GPIO могут выступать как в роли входа, так и в роли выхода – это, как правило, конфигурируется. GPIO контакты часто группируются в порты.

Рассмотрим функцию записи бита void writeBit(bool bit). Контакт, подключенный к датчику температуры (его номер определен директивой DS\_PIN), конфигурируется как выход с помощью библиотечной функции nrf\_gpio\_cfg\_output(DS\_PIN). С помощью функции nrf\_gpio\_pin\_write(DS\_PIN, 0) его значение устанавливается в 0, это значит что напряжение на этом контакте будет равно нулю. Далее с помощью функции nrf\_delay\_us(writeDelay), останавливается выполнение кода на указанное количество микросекунд. Переменная writeDelay хранит временной интервал прописанный директивой DELAY\_US\_WRITE\_1 или DELAY\_US\_WRITE\_0, это промежуток времени, в течении которого контакт должен находится в состоянии нуля для записи бита 1 или 0 соответственно. Далее функция nrf\_gpio\_cfg\_input(DS\_PIN, NRF\_GPIO\_PIN\_NOPULL) переводит контакт в состояние входа без подтяжки, таким образом на контакте датчика устанавливается логическая единица (благодаря подтягивающему резистору, который размещен на схеме). Функция nrf\_delay\_us(DELAY\_US\_TIME\_SLOT - writeDelay) ожидает до завершения временного интервала, отведенного на передачу бита данных.

Рассмотри функцию чтения бита bool readBit(). Контакт конфигурируется на выход устанавливается в единицу. Далее выполнение останавливается на временной интервал для чтения бита. После этого контакт конфигурируется на вход, выжидается еще один временной интервал и состояние контакта считывается с помощью функции nrf\_gpio\_pin\_read(DS\_PIN). Она возвращает 0 или 1 в зависимости от напряжения на контакте. Функция вернет 1, если напряжение равно напряжению логической единицы (обычно это напряжение находится в диапазоне от половины напряжения питания до напряжения питания), если напряжение меньше логической единицы – 0. Далее вызывается функция задержки выполнения кода для ожидания до завершения временного интервала передачи бита.

На рисунке 3.1 показана временная диаграмма чтения и записи битов 0 и 1, которую предоставляет производитель датчика в технической спецификации.

В листинге 3.2 представлены функции чтения и записи байта. С помощью логических операций и битовых сдвигов байты записываются и считываются побитово.

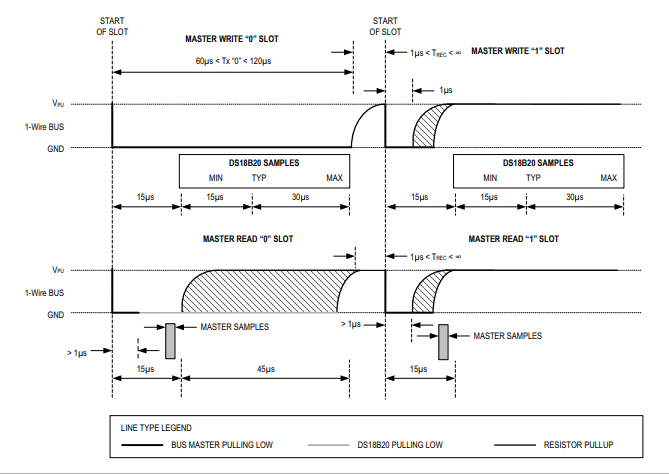


Рисунок 3.1 – временная диаграмма чтения и записи битов для датчика DS18B20

Листинг 3.2 – функции чтения и записи байта

1. static inline void writeByte(uint8\_t byte) {
2. for (int8\_t i = 8; i > 0; i--) {
3. writeBit(byte & 1);
4. byte >>= 1;
5. }
6. }
7. static inline uint8\_t readByte(void) {
8. uint8\_t result = 0;
9. for (int8\_t i = 8; i > 0; i--) {
10. result >>= 1;
11. if (readBit()) {
12. result |= 0x80;
13. }
14. }
15. return result;
16. }

Далее с помощью описанных функций датчику отправляются команды. В общем виде чтение температуры выглядит следующим образом:

проверить линию (по умолчанию она должна быть подтянута к единице);

отправить команду измерения температуры;

отправить команду чтения памяти датчика;

прочитать память датчика;

конвертировать полученные данные в температуру.

Рассмотрим подробнее измерение заряда аккумулятора. Работа с аналогово-цифровым преобразователем (АЦП) представлена в листинге 3.3.

Листинг 3.3 – настройка и использование АЦП

1. #define DIVIDER 5.0

#define GAIN 6.0

#define REFERENCE 0.6

#define RESOLUTION 1024.0

float adc\_measure(void) {

uint16\_t value;

nrfx\_err\_t error = nrfx\_saadc\_sample\_convert(SAADC\_CH\_PSELP\_PSELP\_VDDHDIV5, &value);

if (error != NRFX\_SUCCESS) {

return DEFAULT\_VOLTAGE\_RESULT;

}

return value \* DIVIDER \* GAIN \* REFERENCE / RESOLUTION;

}

void adc\_init(void) {

nrf\_saadc\_channel\_config\_t channel\_config = NRF\_DRV\_SAADC\_DEFAULT\_CHANNEL\_CONFIG\_SE(SAADC\_CH\_PSELP\_PSELP\_VDDHDIV5);

channel\_config.acq\_time = NRF\_SAADC\_ACQTIME\_40US;

nrf\_drv\_saadc\_init(NULL, saadc\_callback);

nrf\_drv\_saadc\_channel\_init(0, &channel\_config);

nrf\_drv\_saadc\_calibrate\_offset();

}

Перед получением значений нужно инициализировать АЦП. Для этого заполняется структура nrf\_saadc\_channel\_config\_t, затем вызываются функции nrf\_drv\_saadc\_init, nrf\_drv\_saadc\_channel\_init и nrf\_drv\_saadc\_calibrate\_offset. В качестве канала для измерения выбран SAADC\_CH\_PSELP\_PSELP\_VDDHDIV5 – это встроенный канал для измерения напряжения питания.

Для получения напряжения используется функция float adc\_measure(void). В ней вызывается функция nrfx\_saadc\_sample\_convert, результат выполнения которой преобразуется в напряжение по формуле, которая учитывает разрешение, опорное напряжение, усиление и делитель.

Измеренное на аккумуляторе напряжение преобразуется в процент заряда по формуле (voltage - VOLTAGE\_MIN) / (VOLTAGE\_MAX - VOLTAGE\_MIN) \* 100, где voltage – измеренное значение напряжения, VOLTAGE\_MIN – напряжение разряженного аккумулятора (3,0 В), VOLTAGE\_MAX – напряжение заряженного аккумулятора (4,2 В)

Получение статусов от микросхемы зарядки происходит с помощью прерываний. Инициализация прерываний не отличается от инициализации других модулей: заполняется структура с параметрами и вызывается библиотечная функция инициализации.

Прерывание (англ. interrupt) — сигнал от программного или аппаратного обеспечения, сообщающий процессору о наступлении какого-либо события, требующего немедленного внимания. Прерывание извещает процессор о наступлении высокоприоритетного события, требующего прерывания текущего кода, выполняемого процессором. Процессор отвечает приостановкой своей текущей активности, сохраняя свое состояние и выполняя функцию, называемую обработчиком прерывания (или программой обработки прерывания), которая реагирует на событие и обслуживает его, после чего возвращает управление в прерванный код.

Прерывание настраивается на изменение состояния контакта GPIO, который подключен к выводу микросхемы зарядки. При изменении напряжения на контакте GPIO вызывается обработчик, который считывает значение и сохраняет его в переменную. Значение переменной хранит текущее состояние контакта и его можно прочитать из любого модуля прошивки.

Все данные собираются вместе и передаются в режиме вещания (доступны всем, кто осуществляет сканирования) с помощью BLE.

Беспроводная технология Bluetooth с низким энергопотреблением (англ. Bluetooth Low Energy, Bluetooth LE, BLE, представленная также как Bluetooth Smart) — выпущенная в декабре 2009 года версия спецификации ядра беспроводной технологии Bluetooth, наиболее существенным достоинством которой является сверхмалое пиковое энергопотребление, среднее энергопотребление и энергопотребление в режиме простоя.

## Реализация мобильного приложения

Для реализации мобильного приложения будем использовать язык Java и среду разработки Android Studio.

Для обработки данных, получаемых с датчика температуры, определим сущность получаемого пакета. Он будет хранить:

строку (String) с MAC – адресом датчика температуры;

число (float) хранящее значение температуры;

число (int) хранящее уровень сигнала (RSSI) датчика;

число (int) хранящее id текущего пакета;

число (int) хранящее уровень заряда батареи в процентах;

флаг (boolean) сигнализирующий о наличии внешнего питания;

временна́я метка (Timestamp) хранящая время и дату получения пакета.

MAC-адрес (от англ. Media Access Control – управление доступом к среде, также Hardware Address, также физический адрес) – уникальный идентификатор, присваиваемый каждой единице активного оборудования или некоторым их интерфейсам в компьютерных сетях.

Показатель уровня принимаемого сигнала, RSSI (англ. received signal strength indicator, RSSI) (телекоммуникации) – полная мощность принимаемого приёмником сигнала. Измеряется приёмником по логарифмической шкале в дБм (dBm, децибел относительно 1 милливатта).

Временна́я метка (также метка времени или timestamp с англ. — «временна́я печать») – это последовательность символов или закодированной информации, показывающей, когда произошло определённое событие. Обычно показывает дату и время (иногда с точностью до долей секунд).

Термин пришёл от сургучных печатей, используемых в офисах / на почте, чтобы отпечатать текущую дату (иногда и время) в подписи бумажных документов или записать, когда документ был принят. Типичные примеры метки времени – штемпель на письме.

Сейчас использование термина расширилось на цифровую информацию. Например, компьютерные файлы содержат метки, показывающие, когда последний раз меняли файл; цифровые камеры добавляют временные метки к изображениям.

Далее было создано приложение, которое сканирует BLE, находит пакеты датчика температуры и выводит их на экран. Получение пакетов от датчика описано в листинге 3.4.

Листинг 3.4 – сканирование BLE

1. ScanCallback scanCallback = new ScanCallback(){
2. @Override public void onScanResult(ScanResult result) {
3. TemperatureSensor ts = TemperatureSensor.parse(result);
4. if (ts != null) {
5. if (sensors.contains(ts))
6. sensors.remove(ts);
7. sensors.add(ts);
8. StringBuilder sb = new StringBuilder();
9. for (TemperatureSensor t : sensors){
10. sb.append(t.toString());
11. sb.append("\n");
12. }
13. textView.setText(sb.toString());
14. }
15. }};
16. BluetoothAdapter
17. .getDefaultAdapter()
18. .getBluetoothLeScanner()
19. .startScan(scanCallback);

Создается scanCallback – обработчик события получения BLE пакета. Функция onScanResult получает объект ScanResult с полученными данными. Функия TemperatureSensor.parse обрабатывает ScanResult и возвращает объект TemperatureSensor, если это пакет от датчика температуры, или null, если это пакет от другого устройства. Если получен валидный пакет, он добавляется в множество sensors. Это множество выводится на экран в текстовом виде.

После создания обработчика запускается сканирование с помощью функции startScan. Результат работы приложения можно увидеть на рисунке 3.2.



Рисунок 3.2 – приложение на стадии разработки

На основании требований и макета был разработан интерфейс мобильного приложения показанный на рисунке 3.3.

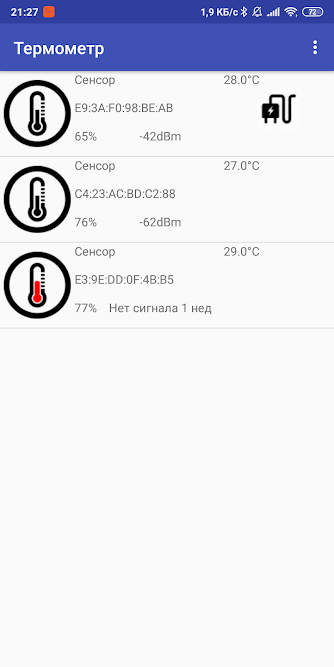


Рисунок 3.3 – интерфейс приложения

Датчик подключенный к зарядке помечается специальным значком. Датчик добавленный в избранное помечается красным цветом термометра. В меню можно отфильтровать избранные датчики. Меню показано на рисунке 3.4.



Рисунок 3.4 – меню приложения

Для хранения истории была подключена база данных Firebase.

Firebase – американская компания, поставщик облачных услуг, основанная в 2011 году Эндрю Ли и Джеймсом Тэмплином, и поглощённая в 2014 году корпорацией Google.

Основной сервис – облачная СУБД класса NoSQL, позволяющая разработчикам приложений хранить и синхронизировать данные между несколькими клиентами. Поддержаны особенности интеграции с приложениями под операционные системы Android и iOS, реализовано API для приложений на JavaScript, Java, Objective-C и Node.js.

API (программный интерфейс приложения, интерфейс прикладного программирования) (англ. application programming interface) – описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой.

База данных Firebase на платформе Android предоставляет удобные возможности для хранения данных. Если при добавлении записей в базу данных отсутствует интернет-соединение, то данные сохраняются в памяти и отправляются при появлении связи. Запросы на получение данных так же хранятся в памяти некоторое время и кэшированные записи можно получить без подключения к интернету.

Полученные по Bluetooth пакеты сохраняются в базе данных. При отсутствии сигнала от избранных датчиков, загружается их последнее состояние из базы данных.

Для отображения истории измерений создан экран отображающий график температуры с возможностью указания промежутка времени для фильтрации значений. Этот экран показан на рисунке 3.5.

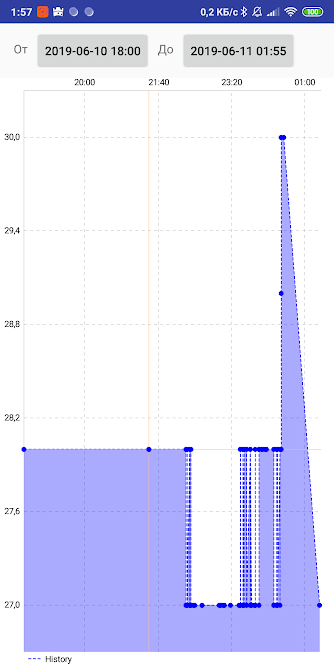


Рисунок 3.5 – просмотр истории измерений

## Реализация печатной платы

После проверки работы схемы на прототипе начата разработка печатной платы в среде Altium Designer. В этой программе печатная плата разделена по слоям, которые могут хранить следующую информацию:

Top/bottom/middle layer это карта верхнего/нижнего/промежуточных слоёв меди, которая будет создавать электрический контакт между компонентами, которые будут расположены на плате. Наличие хотя бы одного такого слоя обязательно для производства печатной платы.

Top/bottom overlay это верхний и нижний слои шелкографии – графическая информация, которая будет напечатана на плате. Обычно это границы компонентов, их подписи, подписи разъёмов, информация о производителе и т.д. Необязательный для производства слой.

Multi-layer это расположение отверстий. Отверстия бывают монтажные – для крепления платы в корпус, и переходные – для электрического соединения слоёв платы. Этот слой нужен для производства печатной платы.

3D-model это 3D-модели компонентов. Этот слой не нужен для производства платы, но будет полезен при проектировании корпуса или совмещении нескольких печатных плат.

Contour это слой с обозначениями центров и границ компонентов. Нужен для автоматического монтажа компонентов. Такие слои называются механическими. Допускается создание нескольких слоёв с дополнительной информацией для сборки платы.

Top/bottom paste это карта мест, на которые нужно нанести паяльную пасту. Этот слой нужен для автоматического монтажа компонентов.

Top/bottom solder это карта мест, которые не требуют нанесения защитной маски. В этих местах верхний или нижний слой меди не будет изолирован. Обычно это площадки для монтажа компонентов. Этот слой нужен для производства платы.

На первом этапе создаются объекты PCB Library, их называют отпечатками (Footprint) компонентов. Под отпечатком подразумевается место на плате, к которому будет припаиваться компонент. Рисунок меди на плате совпадает с электрическими выводами компонента. Помимо отпечатка, объект PCB Library может содержать любые слои печатной платы: контур, 3D-модель, и т.д. Пример готового отпечатка (SMD-резистора) показан на рисунке 3.6.

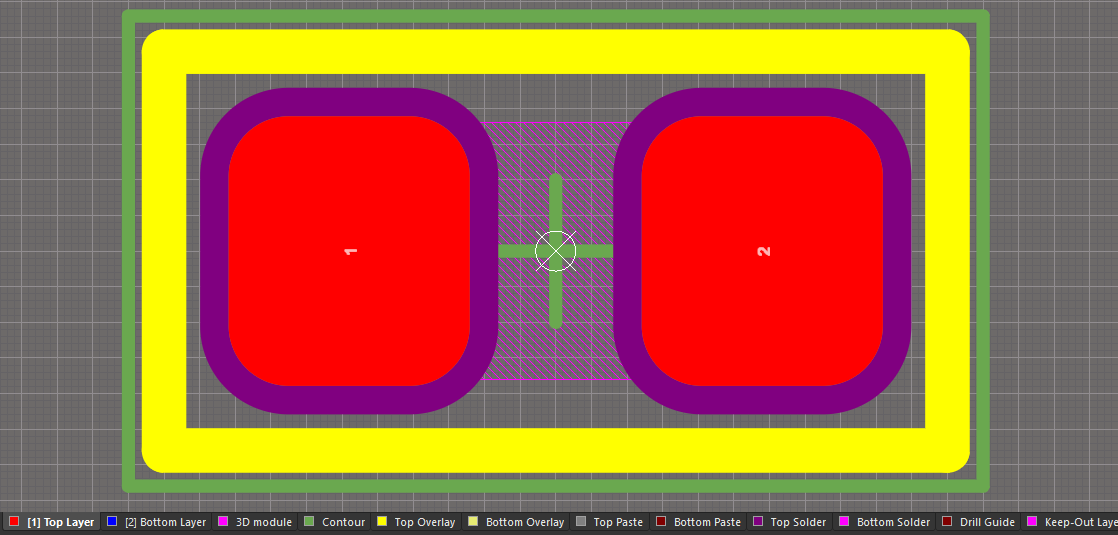


Рисунок 3.6 – отпечаток SMD-резистора

Поверхностный монтаж сокращённо называется SMT (от англ. Surface Mount Technology – Технология монтажа на поверхность, ТМП).

SMD – это Surface Mount Device, то есть компонент или устройство, монтируемое на поверхность. Таким образом, под SMD надо понимать именно компоненты и радиодетали, а не технологию в целом. Иногда SMD-элементы называют чип-компонентами, например, чип-конденсатор или чип-резистор.

Условно-графические обозначения создаются в SCH (Schematics) Library. К одному компоненту SCH Library можно привязать несколько отпечатков, к примеру резисторы могут быть поверхностного или сквозного монтажа и разных типоразмеров, при этом на схеме они будут обозначаться одинаково.

При создании схемы, на ней размещаются компоненты из библиотеки SCH. В этот момент выбирается конкретный отпечаток. Когда схема готова, отпечатки переносятся на плату.

В редакторе печатных плат настраиваются правила проектирования для каждой конкретной платы. В правилах указываются допустимые зазоры между компонентами, расстояния между проводниками, ширина и длина проводников, минимальные и максимальные размеры переходных отверстий и многое другое. Эти правила будут учитываться как при автоматическом, так и при ручном проектировании платы. С помощью правил учитываются требования производителя при заказе изготовления печатных плат.

Несмотря на возможность автоматической компоновки и трассировки печатной платы, многие делают это вручную. Это позволяет учесть рекомендации производителей микросхем и учесть нюансы, которые невозможно внести в правила. Средства Altium Designer позволяют вести интерактивную трассировку с проверкой правил на лету. Это позволяет ускорить процесс разработки печатной платы и делает работу более удобной.

На рисунке 3.7 изображен верхний слой готового дизайна печатной платы, на нем размещены все компоненты.

На рисунке 3.8 изображен нижний слой печатной платы.

На рисунке 3.9 изображена 3D модель печатной платы.

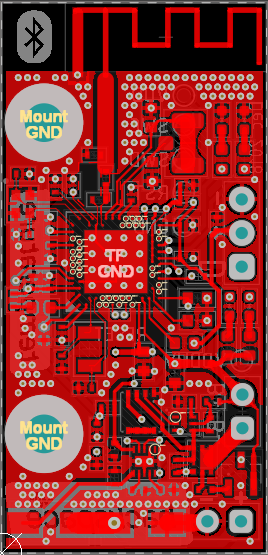


Рисунок 3.7 – верхний слой печатной платы

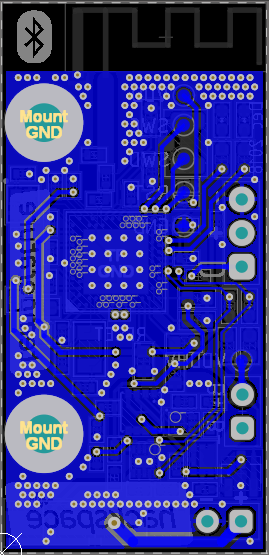


Рисунок 3.8 – нижний слой печатной платы

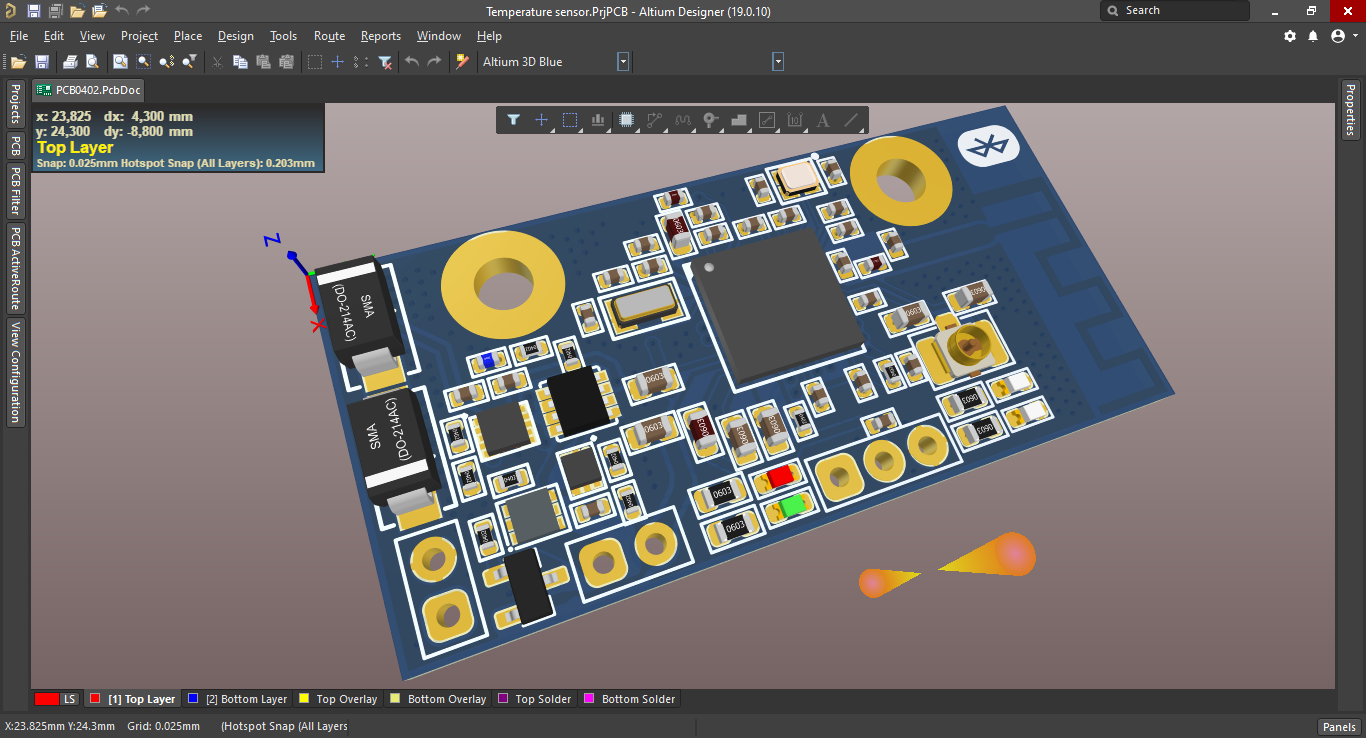


Рисунок 3.9 – объёмная модель платы

Когда плата готова, можно приступать к её производству. Для производства печатной платы генерируются Gerber-файлы. Производство печатной платы заказывается на специальных сервисах, например pcbway.com.

Gerber – файловый формат, представляющий собой способ описания проекта печатной платы для изготовления фотошаблонов на самом разнообразном оборудовании. Практически все современные системы автоматизации проектных работ для электроники позволяют генерировать выходные файлы в формате Gerber; с другой стороны – почти всё современное оборудование позволяет считывать данные в этом формате. Формат был изначально разработан корпорацией Gerber Systems (Хартфорд, штат Коннектикут, США), признанным производителем плоттеров, устройств широкоформатной печати и оборудования прецизионной резки, систем автоматического оптического контроля (AOI) для печатных плат, CAD/CAM систем.

По сути Gerber-файл представляет собой текстовое описание последовательности команд, направленных на прорисовку различных элементов топологии (контактных площадок, переходных отверстий, линий, дуг, текстовых надписей) с помощью графопостроителя. Фактически данные в формате Gerber представляют собой программный код, управляющий выбором инструмента рисования, перемещением его в точку с заданными координатами и выполнением самой операции рисования. При изготовлении фотошаблонов рисование на светочувствительной плёнке производится световым пятном заданной формы – апертурой.

Производство современных печатных плат автоматизировано, поэтому большинство операций (от создания рисунка проводников до монтажа компонентов) может быть выполнено на автоматизированных станках.

Фотографии готовой платы показаны на рисунках 3.10-3.11.



Рисунок 3.10 – верхняя часть платы

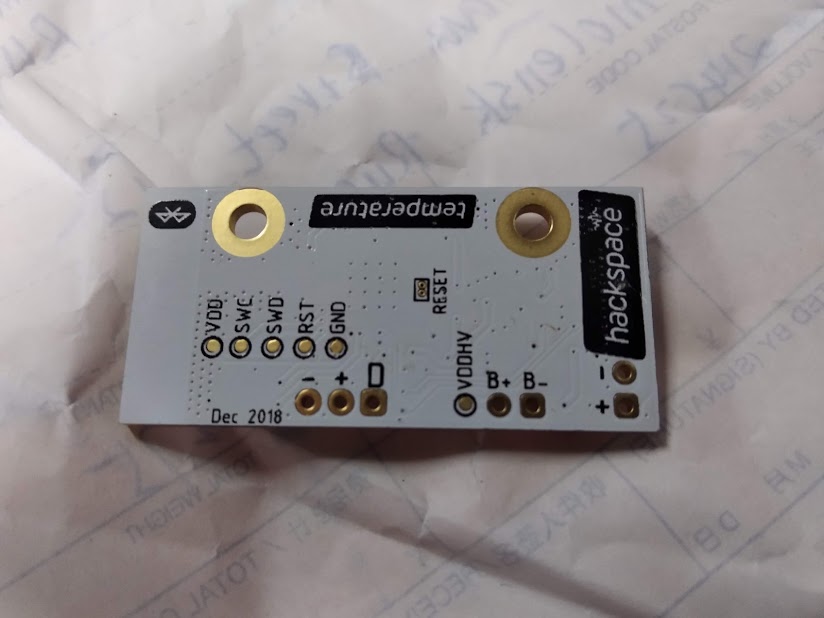


Рисунок 3.11 – нижняя часть платы

## Тестирование системы

Перед первым включением плат был проведен контроль сопротивлений основных линий питания и логических выходов на предмет короткого замыкания. После того как мы удостоверились в отсутствии короткого замыкания, плата была запущена с помощью лабораторного блока питания.

Лабораторный блок питания – вторичный источник электропитания, который позволяет регулировать выходное напряжение питания и ограничивать максимальный ток нагрузки.

Была проверена работа основных защит и зарядки аккумулятора.

После этого в микроконтроллер была загружена прошивка. Работоспособность прошивки была подтверждена проверкой отправляемых данных, они соответствуют состоянию устройства.

Точность измерения температуры была подтверждена с помощью термометра, прошедшего поверку.

Была произведен замер дальности работы устройства. На открытой местности расстояние составило более ста метров. В грузовом прицепе с холодильной установкой, при размещении устройства внутри прицепа, сигнал оказался достаточным, чтобы снимать показания из кабины водителя.

По результатам тестирования устройства, прошивки и приложения, неполадок не выявлено.

ЭКОНОМИЧЕСКАЯ ЧАСТЬ

## Обоснование необходимости выведения продукта на рынок

# ЗАКЛЮЧЕНИЕ

В результате выполнения дипломного проекта

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

ПРИЛОЖЕНИЯ