МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Радиофизический факультет

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК
Руководитель ООП
к.ф.-м.н., доцент
_____ В.А. Мещеряков
«20» января 2022 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА СПЕЦИАЛИСТА
(ДИПЛОМНАЯ РАБОТА)

РАЗРАБОТКА СИСТЕМЫ СБОРА КЛИМАТИЧЕСКИХ ПАРАМЕТРОВ В УЧЕБНЫХ АУДИТОРИЯХ

по основной образовательной программе подготовки специалиста
по специальности 11.05.01 «Радиоэлектронные системы и комплексы»

Богославский Артём Андреевич

Руководитель ВКР
зав. кафедры ИТИДиС
_____ С.Н. Торгаев
«20» января 2022 г.

Автор работы
студент группы № 769
_____ А.А. Богославский

Томск-2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Радиофизический факультет

УТВЕРЖДАЮ
Руководитель ООП
к.ф.-м.н., доцент
_____ В.А. Мещеряков
« 23 » декабря 2021 г.

ЗАДАНИЕ
на подготовку ВКР специалиста
студенту Богославскому Артёму Андреевичу группы № 769

1. Тема ВКР: Разработка системы сбора климатических параметров в учебных аудиториях.

2. Срок сдачи студентом выполненной ВКР:
а) на кафедре 20.01.2022,
б) в ГЭК 02.02.2022.

3. Краткое содержание работы:
Система менеджмента качества в Национальном исследовательском Томском государственном университете, опираясь на требования СанПиН регламентирует производить замеры основных климатических показателей в учебных и лабораторных помещениях минимум два раза в сутки. Для контроля за климатом в учебных и лабораторных помещениях необходима сеть специальных устройств, специализирующаяся на сборе, систематизации и хранении климатических данных. Данная работа направлена на разработку устройства сбора и хранения основных климатических данных в учебной аудитории.

4. Календарный график выполнения ВКР:

а) оформление литературного обзора	03.01.2022–05.01.2022
б) оформление результатов разработки программного обеспечения для микроконтроллера ESP8266	05.01.2022–17.01.2022
в) оформление результатов исследования работы системы	05.01.2022–17.01.2022
г) техническое оформление ВКР в соответствии с требованиями	17.01.2022–19.01-2022

5. Дата выдачи задания « 23 » декабря 2021 г.

Руководитель НИР –
зав. кафедры ИТИДиС _____ Торгаев С.Н.

Задание принял к исполнению _____ Богославский А.А.

АННОТАЦИЯ

Отчет 70 с., 4 гл., 70 рис., 1 таблица, 19 источников.

ARDUINO IDE, VISUAL STUDIO, FUSION 360, ESP8266, NODEMCUV3, ESP-NOW, DHT11, СБОР КЛИМАТИЧЕСКИЙ ДАННЫХ, ОСЬ РАДИОСВЯЗИ, WI-FI.

Цель работы: Разработка системы сбора климатических параметров в учебных аудиториях.

Методы: программирование микроконтроллеров ESP8266 в среде Arduino IDE, создание прототипов устройств.

В ходе выполнения ВКР были получены следующие результаты:

- освоены и реализованы алгоритмы опроса датчика температуры и влажности DHT11;
- освоены и реализованы алгоритмы нахождения и конвертации MAC-адресов с дальнейшим созданием таблицы адресов оси радиосвязи;
- освоены алгоритмы структурирования данных, реализованы алгоритмы программирования протоколов передачи данных;
- практически реализована работа программы для ESP8266;
- практически реализована работа оси радиосвязи;
- практически реализовано взаимодействие с программным обеспечением н ПК;
- произведен расчет стоимости устройства;
- произведен расчет автономности устройств;
- разработан и распечатан корпус устройств;
- произведено исследование работы системы в разных климатических условиях;
- произведено исследование работы системы при выходе устройств из строя.

ОГЛАВЛЕНИЕ

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ, СОКРАЩЕНИЙ, ТЕРМИНОВ

– ПК – это прежде всего компьютер, предоставляющий возможность его использования пользователем в течение одной рабочей сессии. В англоязычных источниках ПК звучит как Personal Computer и имеет сокращение PC.

– МК (англ. Micro Controller Unit, MCU) — микроконтроллер, сочетающий на одном кристалле функции процессора и периферийных устройств, содержащий ОЗУ и (или) ПЗУ

– MAC (media access control address) – это уникальный идентификатор, который присваивается к сетевой карте/адаптеру любого устройства, способного подключаться к сети интернет.

– COM-порт (последовательный порт) – это двунаправленный последовательный интерфейс, который обменивается данными в байтах.

– СанПиН – Санитарные (санитарно-эпидемиологические) правила и нормы. Государственные подзаконные нормативные правовые акты с описаниями и требованиями безопасных и безвредных для человека, популяции людей и потомков факторов среды обитания и их оптимальных и безопасных количественных параметров с целью сохранения здоровья и нормальной жизнедеятельности.

ВВЕДЕНИЕ

Система менеджмента качества в Национальном исследовательском Томском государственном университете, опираясь на требования СанПиН регламентирует производить замеры основных климатических показателей в учебных и лабораторных помещениях минимум два раза в сутки. Основными климатическими показателями в учебных и лабораторных аудиториях являются температура и влажность, так как эти показатели могут достаточно сильно влиять на образовательный и исследовательский процесс. Для поддержания температуры и влажности на оптимальном уровне, зачастую, в учебных помещениях устанавливаются кондиционеры и подобные им устройства. Стоит отметить, что такие устройства наряду с огромной пользой могут принести и вред. Вышесказанное подтверждает, что для контроля за климатом в учебных и лабораторных помещениях необходима сеть специальных устройств, специализирующаяся на сборе, систематизации и хранении климатических данных.

В данной работе будет рассмотрена разработка устройства сбора и хранения основных климатических данных в учебной аудитории. В основе устройства лежит микроконтроллер ESP8266, совместно с датчиком температуры и влажности DHT11.

1 Обзор литературы

1.1 Техника безопасности при работе с электроникой

Программист перед стартом своей работы должен:

- Провести осмотр своего рабочего места; если нужно, то привести его в порядок;
- Провести регулировку освещения, чтобы экран был ему хорошо виден и не отражал свет;
- Проконтролировать корректное подключение электрических частей компьютера к сети;
- Проконтролировать отсутствие оголенных частей проводки на электрических проводах компьютера;
- Провести проверку целостности стола, стула, подставки для ног, выдвижной части стола для клавиатуры и т. д.; если необходимо, программист должен отрегулировать все эти моменты.

Программисту перед стартом работы запрещается:

- Работать за компьютером при обнаружении неисправности оборудования или электрических проводов компьютера;
- Начинать работу, если отсутствует заземление;
- Работать, если в рабочем помещении отсутствует огнетушитель и аптечка для первой медицинской помощи.

Во время своей работы программист должен:

- Выполнять только ту работу, за которую он несет ответственность;
- Следить за чистотой своего рабочего места;
- Не закрывать вентиляционные «окошки» компьютера;
- Корректно прекращать работу компьютера, когда это нужно;
- Следить за соблюдением своего графика работы и отдыха;
- Правильно и по назначению эксплуатировать компьютер и все его части;
- Вовремя выполнять физические упражнения для глаз, шеи, рук и туловища;

– Следить за своим расположением на рабочем месте: правильная осанка, расстояние до и экрана и т. д.

В процессе работы программисту запрещено:

– При включенном питании компьютера подключать или отключать от него устройства и вообще прикасаться к задней части системного блока;
– Заваливать свое рабочее место ненужными вещами: бумагами и посторонними предметами;
– Выключать питание компьютера «из розетки», не выполнив правильное завершение работы устройства;
– Самостоятельно ремонтировать или вскрывать системный блок.

После окончания рабочего дня или своего рабочего времени программист должен:

– Правильно завершить работу всех запущенных программ и устройств;
– Проверить отсутствие в дисководах дисков или дискет;
– Отключить системный блок от электросети;
– Отключить дополнительные устройства от электросети;
– Осмотреть свое рабочее место и привести его в порядок, если это необходимо.

В момент аварийных ситуаций программист должен:

– При обнаружении оголенных проводов, неисправного заземления, а также в случае появления запаха гари в этот же момент завершить работу компьютера и сообщить о подобном происшествии вышестоящему руководству или дежурному специалисту;
– Уметь оказать первую медицинскую помощь человеку, пострадавшему от электрического тока;
– Следить за наличием сбоя программного обеспечения компьютера и в случае их появления сообщить вышестоящему руководству или дежурному специалисту;

– В случае появления возгорания по возможности завершить работу компьютера, отключить компьютер от электрической сети, вызвать пожарную команду, сообщить о происшествии и предпринять самостоятельные попытки тушения пожара, если это возможно.

При работе с платами программирования:

– Избегать металлических предметов под платой или изолировать всю нижнюю сторону при помощи непроводящей пластины или изоляционной ленты;
– Расположить блоки питания, другие источники напряжения и провода с напряжением более 5В подальше от макетной платы;
– Присоединять плату по возможности не непосредственно к персональному компьютеру, а через разветвитель, который обычно снабжен дополнительной системой защиты;
– Использовать специальный заземляющий браслет, соединенный с ковриком;
– Стараться не касаться токопроводящих элементов печатной платы;
– Использовать специальную esd-одежду (халат и обувь);
– Учитывать дополнительные требования ТБ, такие как использование специального покрытия столешницы либо специального esd-коврика, заземление всего оборудования.

1.2 Датчик температуры и влажности DHT11

Для простоты использования в единый корпус устройства помещают и датчик влажности, и датчик температуры, к которым может быть добавлен микроконтроллер для того, чтобы принимать данные с устройства было проще, а его выходные данные представлялись цифровым сигналом.

DHT11 (рисунок 1) — широко распространённый датчик для определения относительной влажности и температуры, состоит из емкостного датчика влажности и термистора. Также, датчик содержит в себе восьми-битный микроконтроллер с АЦП для преобразования аналоговых значений

влажности и температуры в цифровые. Датчик позволяет измерять влажность в диапазоне от 20 % до 90 % и температуру от 0 °С до 50 °С с погрешностями до ± 5 %.

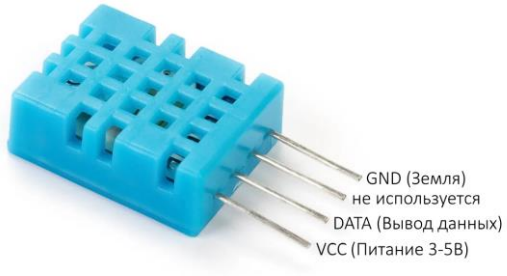


Рисунок 1 – Внешний вид и назначение контактов DHT11

Датчик выполнен в корпусе с четырьмя выводами и осуществляет передачу данных с помощью интерфейса «1-wire», который обеспечивает хорошее качество выходного сигнала, помехозащищенность и малое энергопотребление. Для связи используется один провод/шина с открытым коллектором, поэтому обязательна подтяжка резистором 5-10 кОм к плюсу питания. [1]

Датчик подключается к питанию и к земле, а шина данных подключается дополнительно к питанию через токоограничивающий резистор. Схема подключения датчика DHT11 к управляющему МК изображена на рисунке 2.

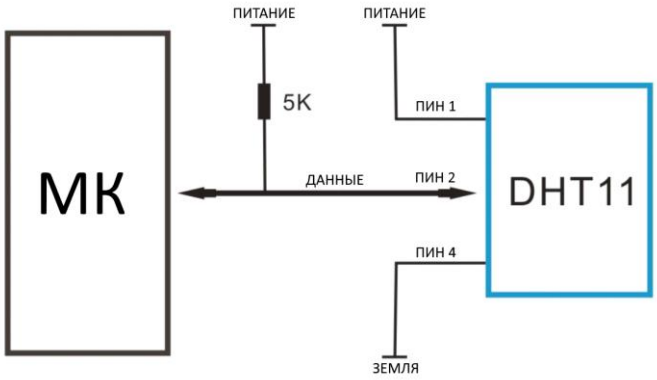


Рисунок 2 – Схема включения DHT11

На рисунке 3 представлена схематичная осциллограмма сигнала. Один процесс передачи данных с датчика DHT11 на управляющий МК занимает 4 мс. За это время датчик передаёт пакет данных в 5 байт: первый байт содержит целое значение влажности, второй байт – десятичное значение

влажности, третий байт – целое значение температуры, четвёртый байт – десятичное значение температуры, а пятый байт является контрольной суммой. Контрольная сумма представляет собой сумму первых четырех байт и служит для проверки принятого пакета данных.

Рисунок 3 – Схематичная осциллограмма сигнала

Передача данных представляет собой последовательность импульсов в цепи питания датчика. В самом начале шиной управляет МК: подает импульс инициализации – низкий уровень питания на 18 мкс, а затем отпускает шину и переходит в режим чтения. Через 20-40 мкс DHT11 формирует ответ с периодом 160 мкс, тем самым говоря, что сейчас начнётся передача данных. Передача происходит по битно, каждый бит представляет собой импульс, который составляет 80 мкс и отпускание шины питания либо на 30 мкс, что обозначает передачу нуля, либо на 70 мкс, что обозначает передачу единицы. Для понимания на рисунке 3 приведена схематичная осциллограмма сигнала во время инициализации и передачи данных.

1.3 Микроконтроллеры ESP8266

ESP8266 — микроконтроллер китайского производителя Espressif Systems с интерфейсом Wi-Fi. Помимо Wi-Fi, микроконтроллер отличается отсутствием флеш-памяти в SoC, программы пользователя исполняются из внешней флеш-памяти с интерфейсом SPI. Микроконтроллер привлек внимание в 2014 году в связи с выходом первых продуктов на его базе по необыкновенно низкой цене. Весной 2016 года началось производство ESP8285, совмещающей ESP8266 и флеш-память на 1 МБайт. Осенью 2015

года Espressif представила развитие линейки — микросхему ESP32 и модули на её основе. [2]

Существует около полутора десятка версий МК серии ESP и огромное количество плат с ними. Рассмотрим самые популярные из них.

ESP-01 (рисунок 4) имеет 8 разведённых контактов (VCC, GND, TXD – ножка передачи информации, RXD – ножка приёма информации, CH_PD – ножка выбора работы на приём или передачу, GPIO0, GPIO2, RST – ножка внешнего сброса) и PCB-антенну (печатный проводник на самой плате). Из разведённых выводов тут присутствуют только 2 GPIO, но не стоит видеть в этом одни минусы. Если нужно будет управлять одним реле или получать данные с датчика температуры, не понадобятся все выводы МК, достаточно будет лишь пары. К тому же, существуют платы расширения с возможностью простой коммутации именно к этой версии МК.

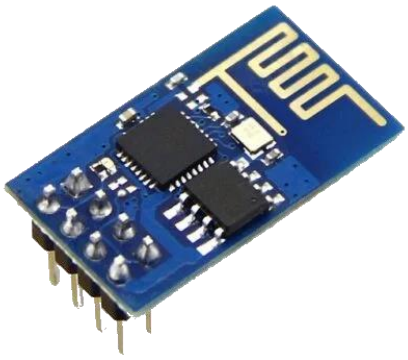


Рисунок 4 – ESP-01

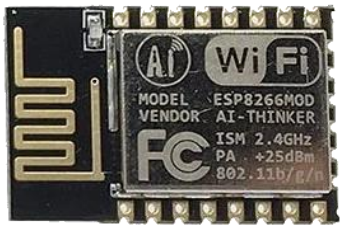
В ESP-03 (рисунок 5) появляется керамическая антенна. Она считается немного эффективней своего печатного собрата. Также на плате разведены все доступные выводы GPIO.

Рисунок 5 – ESP-03

В ESP-07 (рисунок 6) в глаза сразу бросается металлический экран (который перед этим появляется на ESP-06). Оснащен керамической антенной и разъёмом для внешней антенны.

Рисунок 6 – ESP-07

ESP-12 (рисунок 7) обладает наибольшем разнообразием, существует несколько вариантов этой версии: ESP-12S, ESP-12F, ESP-12E. Вторая и третья версии имеют на торце дополнительно 6 разведённых контактов.



а б в

Рисунок 7 – Виды формфакторов плат ESP-8266: а) ESP-12S;б) ESP-12F; в) ESP-12E

1.4 Отладочные макеты на базе микроконтроллера ESP8266

На базе МК ESP8266 создано несколько макетов. WeMos D1 mini (рисунок 8) имеет распайку девяти GPIO-контактов. На плате имеется небезызвестный мост CH34x (такие часто ставят на клоны Arduino). Установлен МК с 4 Мбайт flash-памяти. Макетная плата конструктивно совместима с различными выпускаемыми платами расширения реле/датчиками.

Рисунок 8 – WeMos D1 mini

Первое поколение плат серии NodeMCU v0.9/v1 (рисунок 9). На ней распаяны все 11 GPIO-портов. Некоторые из них обладают дополнительными функциями (UART, I2C – последовательная асимметричная шина, SPI, PWM – широтно-импульсная модуляция, ADC – АЦП). Хотя на плате впаяны контакты, она занимает всю ширину беспаечной макетной платы, что затрудняет работу на ней. МК имеет 4 Мбайт flash-памяти. Также имеется мост CH340.

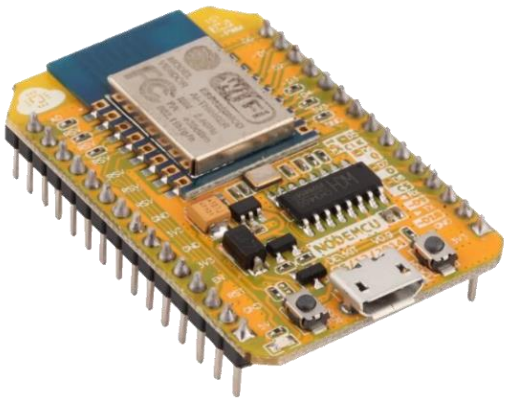


Рисунок 9 – NodeMCU v0.9/v1

NodeMCU v3 (рисунок 10). Финальная версия платы этой серии. Существует и v2 «Amica», которая меньше по габаритам. v3 носит название «LoLin» и отличается от предыдущей версии только размерами и незначительными деталями (например, дополнительной распайкой шины питания). Кроме традиционного моста CH340/CH341 на платы ставят чип CP2102, следует обращать внимание на выбор драйвера для них.

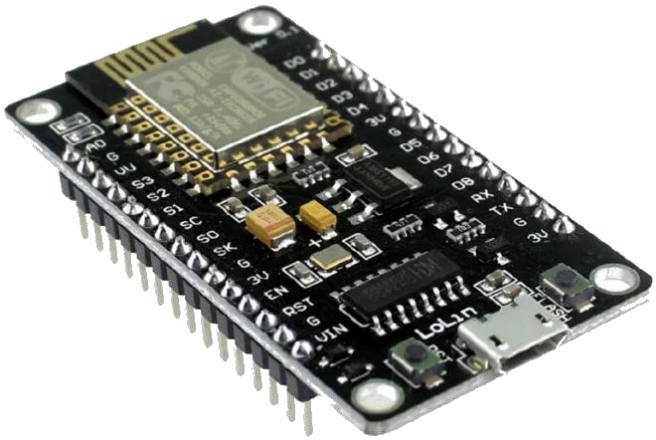


Рисунок 10 – NodeMCU v3

Все эти (и не только эти) платы выполнены на чипсете микроконтроллера ESP8266EX, а следовательно, имеют идентичные характеристики:

- Протоколы: 802.11 b/g/n/e/i.
- Диапазон частот: 2.4 ГГц – 2.5 ГГц.
- Процессорное ядро: Tensilica L106 32 разряда.
- Диапазон напряжений питания: 2.5 В – 3.6 В.
- Среднее потребление тока: 80 мА.
- Режимы WiFi: Station/SoftAP/SoftAP+Station.
- Безопасность: WPA/WPA2.
- Шифрование: WEP/TKIP/AES.
- Обновление прошивки: через UART, по радиоканалу (OTA — Other The Air).
- Сетевые протоколы: IPv4, TCP/UDP/HTTP/FTP.

- Поддержка WiFi Direct (P2P), P2P Discovery, P2P GO (Group Owner) mode, GC (Group Client) mode, P2P Power Management.
- Поддержка LUA-скриптов.

«Из коробки» МК поставляется с прошивкой для работы через AT-команды. Для этого ESP8266 подключается к любому другому МК по UART-интерфейсу. Для демонстрации работы AT-команд ESP8266 можно подключить к компьютеру через USB-UART переходник и запустить монитор последовательного порта (например, из Arduino IDE).

В большинстве случаев намного удобней прошивать МК и работать с ним со своей прошивкой. Однако тут тоже есть свои нюансы. При наличии микроконтроллера без отладочной платы, например ESP-01, потребуется USB-UART переходник, который нужно подключить к МК. Этот переходник обязательно должен быть на 3-вольтовой логике. На рисунке 11 представлен пример подключения USB-UART переходника к ESP-01

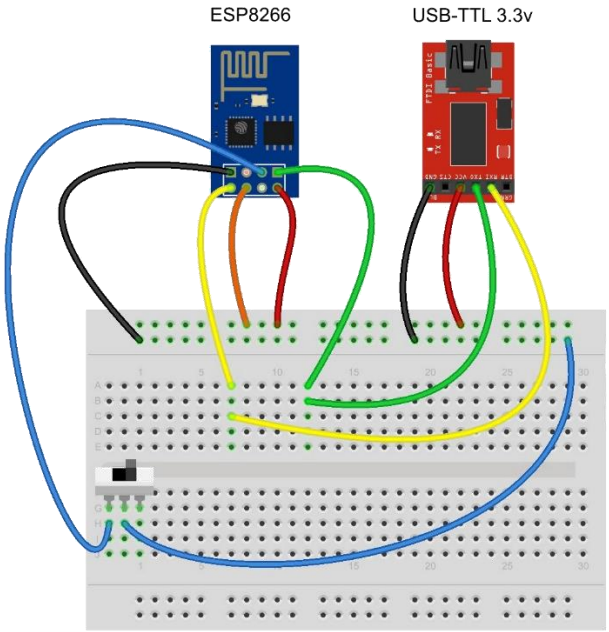


Рисунок 11 – Пример подключения USB-UART переходника к ESP-01

Также для прошивки МК можно использовать любую плату Arduino. Достаточно специальным образом подключить ESP8266 к UART-контактам Arduino, а её саму «отключить», замкнув контакт аппаратного сброса (RESET) на землю. Естественно, питать ESP8266 нужно будет от шины питания 3.3 В.

В этом случае в качестве переходника USB-UART будет выступать мост (чаще всего CH340) на самой плате Arduino. На рисунке 12 представлен пример программирования ESP-01 через Arduino UNO.

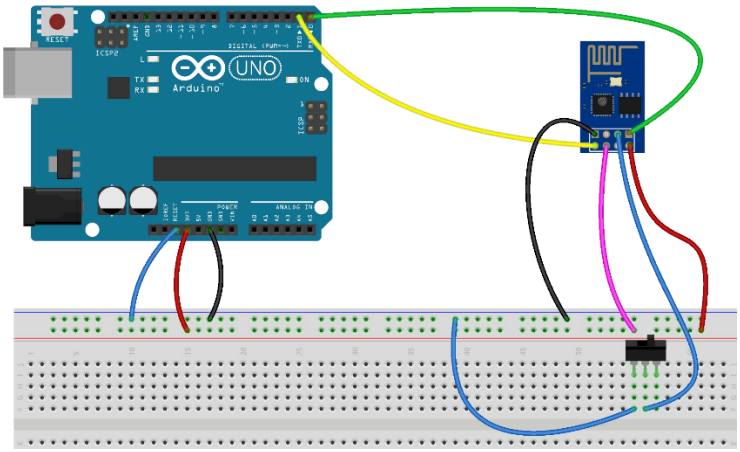


Рисунок 12 – Пример программирования ESP-01 через Arduino UNO

Однако, наилучшим вариантом для комфортной работы с МК будет плата с USB-UART мостом на борту, как NodeMCU, WeMos и прочие. В этом случае ничего дополнительного делать не нужно, достаточно подключить плату через USB и установить необходимые драйверы для работы с USB-UART мостом, которым укомплектована плата.

1.5 Программирование микроконтроллера ESP8266 с помощью Arduino IDE

Программировать МК ESP8266 можно в таких средах, как NodeMCU Flasher или ESPTool, однако нам интересен способ прошивки МК через Arduino IDE. Изначально среда Arduino IDE не предназначена для работы с МК серии ESP. Чтобы это исправить, необходимо сделать следующее:

- Открыть «Файл» → «Настройки» и в поле «Дополнительные ссылки для Менеджера плат» вставить ссылку: http://arduino.esp8266.com/stable/package_esp8266com_index.json
- Открыть «Инструменты» → «Плата» → «Менеджер плат» и в открывшемся списке найти и установить плату «esp8266 by ESP8266 Community»

На рисунке 13 представлен выбор платы на базе МК ESP8266.

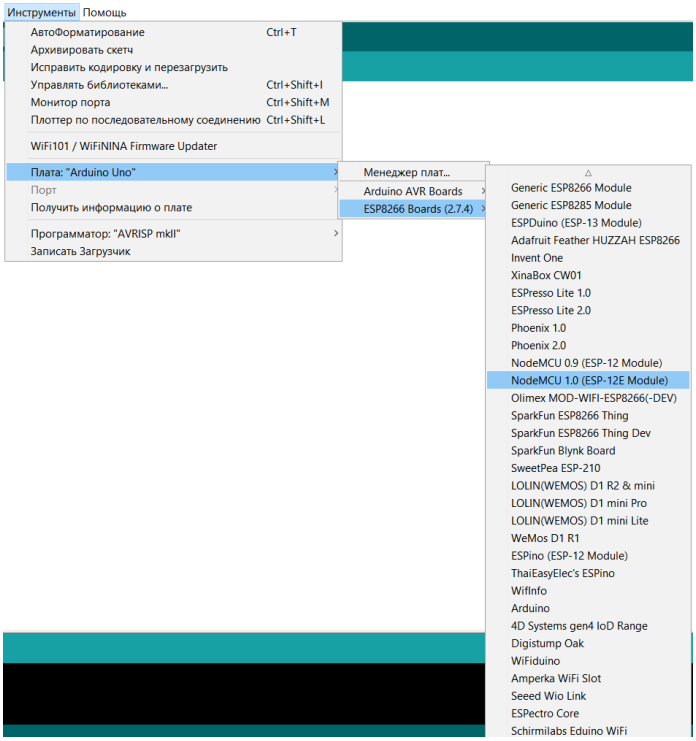


Рисунок 13 – Выбор платы на базе МК ESP8266

У ESP8266 две скорости передачи: основная — её вы указываете при инициализации последовательного порта, и скорость, на которой передаётся

отладочная информация. Она передаётся сразу после подачи питания на МК. Обычно это скорости 115200 бод и 74800 бод соответственно.[3]

1.6 Среда разработки Processing

Processing – язык программирования, основанный на Java, однако в нем нет практически никаких отличий от C++. Язык появился в 2001 году как упрощенный язык и среда разработки для людей, далеких от программирования. Изначально Processing был средством визуализации кода, для программирования анимаций, симуляций, процессов, работы с изображениями, создания простых программ с интерфейсами, игр и так далее. Возможности Processing приличные прямо из коробки и легко расширяются при помощи библиотек, которые устанавливаются в пару кликов мыши. Самая большая прелесть Processing для новичка – документация на официальном сайте и примеры готовых программ. На странице Reference расписаны все стандартные функции и прочие инструменты языка, где у каждой есть своё описание и примеры.

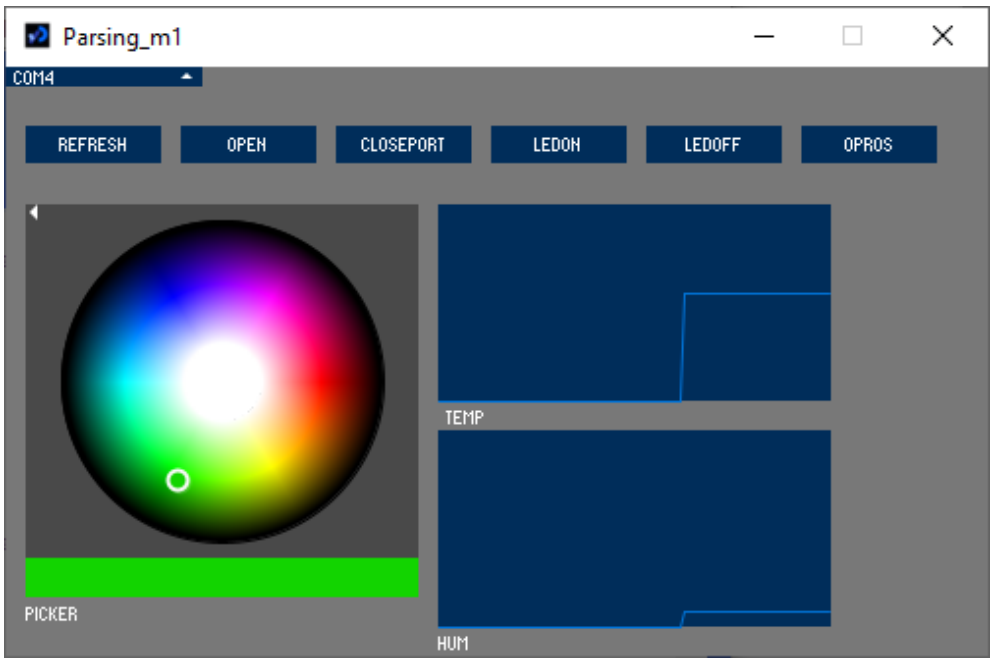


Рисунок 14 – Интерфейс программы в среде Processing

В данной работе среда разработки Processing использовалась для создания простого интерфейса, с помощью которого изучались основы

разработки протокола для связи с МК. Интерфейс созданной программы изображен на рисунке 14.

В данной программе реализованы процессы работы с COM-портом. При запуске программы происходит автоматический поиск доступных COM-портов, а дальнейшее сканирование новых COM-портов доступно по кнопке REFRESH. По нажатию кнопок OPEN и CLOSEPORT выполняются инструкции по открытию и закрытию соответственно COM-порта, выбранного в выпадающем списке COMLIST. При успешном открытии COM-порта становятся доступны остальные элементы интерфейса.

По нажатию кнопки LEDON выполняются операции, последствием которых является зажигание зеленой составляющей RGB-светодиода, подкаченного к ESP8266. Нажатие кнопки LEDOFF выключает светодиод – подает на все компоненты RGB-светодиода нуль. Также для плавного изменения результирующего цвета RGB-светодиода на интерфейсе присутствует инструмент PICKER. Вышеуказанные инструменты используют единую структуру пакета, изображенную на рисунке 15.

1,RED,GREEN,BLUE;

Рисунок 15 – Структура пакета, передаваемого от ПК к МК

В данном пакете: 1 – ключ операции, RED, GREEN, BLUE – соответственно составляющие цветов RGB-светодиода, изменяемые в диапазоне [0;1023].

```
switch (ints[0]){
  case 1:
    analogWrite (R_PIN, ints[1]);
    analogWrite (G_PIN, ints[2]);
    analogWrite (B_PIN, ints[3]);
    break;

   case 3:
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    Serial.print(0);
    Serial.print(',');
    Serial.print(h);
    Serial.print(',');
    Serial.println(t);
    break;

}
```

Рисунок 16 – Реализация работы протокола общения ПК и МК со стороны МК ESP8266.

Также в интерфейсе реализован механизм опроса датчика температуры и влажности DHT11, подключенного к ESP8266. Механизм запускается нажатием кнопки OPROS. Для запроса данных с датчика с ПК отправляется лишь код операции и терминатор «3;». МК, приняв код операции, переходит в часть программного кода, содержащий инструкции по опросу датчика DHT11 и формированию обратного пакета (Рисунок 16), содержащего климатические параметры, которые в последующем выводятся на графики температуры и влажности на интерфейсе ПК, продемонстрированном на рисунке 14.

1.7 Среда разработки Microsoft Visual Studio

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также

веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework и т.д.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии автодополнения, которая дописывает название функции при вводе начальных букв, и возможностью простейшего рефакторинга кода, где под рефакторингом понимается процесс изменения внутренней структуры программы, не затрагивающий её внешнего поведения и имеющий целью облегчить понимание её работы. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного, добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения.

Для данной работы используется версия Visual Studio Community, которая представляет из себя полнофункциональную, расширяемую и бесплатную интегрированную среду разработки для создания современных приложений Android, iOS и Windows, а также веб-приложений и облачных служб.

В Visual Studio Community для написания приложения с графическим интерфейсом будем использовать интерфейс программирования приложения Windows Forms, отвечающий за графический интерфейс пользователя. Данный интерфейс позволяет упростить создание графической оболочки, за счёт того, что большинство необходимых функции таких как кнопки, поля для

ввода информации, выпадающие списки и т.д. добавляются из панели элементов, при этом в коде автоматически описываются добавленные элементы [4].

1.8 Среда разработки Autodesk Fusion 360

Программный комплекс Fusion 360 стал эталонным CAD и CAM приложением благодаря своим безграничным возможностям, удобному интерфейсу, мощной поддержке и малым требованиям. Компания Autodesk с его помощью открыла новую эру в 3D-моделировании. Программа Fusion 360 с успехом применяется как в качестве среды для разработки учебного проекта, так и для запуска крупного промышленного производства. Программный комплекс сумел впитать в себя лучшие черты от других разработок Autodesk, став законченным и самодостаточным продуктом. Одна из особенностей – облачный сервис. Пользователи совместно работают над проектом, вносят изменения, делятся разработками и собирают из него одно целое. Сама программа Autodesk Fusion 360 не требовательна к оснащению компьютера, исправно работает на операционных системах Windows и Linux. Любой проект начинается с эскиза, трехмерной модели. В описании программы Fusion 360 указано большое количество возможностей, предоставляемых пользователю. Среди них особое внимание заслуживают:

- Сплайновое моделирование. Технология Т-сплайнов не теряет актуальности. Программа позволяет моделировать, задавая точную кривизну или редактируя вершины «вручную».
- Твердотельное моделирование. Autodesk Fusion 360 позволяет пользоваться привычными инструментами твердотельного моделирования вкупе с совершенно новыми функциями (например, временной шкалой).
- Сеточные модели. Теперь гораздо проще работать с данными, отсканированными с материальных объектов.

- Параметрическое моделирование. Задавайте размеры при создании эскизов. Любое изменение параметров ведет к волнообразному преобразованию связанных элементов.
- Библиотека готовых решений. Стандартные компоненты существенно экономят время.

Для инженерного анализа в программе Fusion 360 уже реализован целый ряд технологий:

- термальный анализ и термо-напряженный анализ;
- статичные напряжения и собственные частоты;
- нелинейные напряжения;
- моделирование ударных нагрузок;
- оптимизация формы;
- анализ болтовых соединений.

Благодаря виртуальной среде Autodesk удается проверить, как ведет себя сборка в реальных условиях, какие нагрузки сможет выдержать, как будет выглядеть и т.д. Комплекс позволяет редактировать множество исходных данных, задавать характер взаимодействия между деталями, создавать анимацию сборки.

Кроме этого, возможно создавать фотореалистичные изображения при помощи Fusion 360, воспользовавшись всей мощью рендера на облачных серверах Autodesk.

2 Разработка системы сбора климатических параметров

2.1 Разработка правил организации связи

Работа системы подразумевает пять состояний работы МК. В один момент времени на конкретном модуле может быть реализовано одно состояние, номер которого хранится в переменной состояния, однако на разных модулях состояние может быть различным. Рабочий цикл системы продемонстрирован на рисунке 15.

Состояние 0 - ожидание сигнала начала работы.

Состояние 1 - начало работы. В данном состоянии:

– обнуляются значения двухмерного массива климатических параметров;

– поиск «своего» места в таблице адресов;

– запись MAC-адресов «соседей» в соответствующие переменные;

– запись значений переменных передаваемой структуры;

– вызов функции передачи данных на следующий модуль.

Состояние 2 - ожидание пакета данных от следующего модуля. Сопровождается индикацией в виде мигающего светодиода.

Состояние 3 – сбор климатических параметров, запись их в двумерный массив, содержащий климатические параметры и вызов функции передачи пакета на прошлый модуль. МК переходит в данное состояние при возникновении ошибки передачи на следующий модуль – в случаях если следующий модуль вышел из строя или не существует.

Состояние 4 – вывод в COM-порт двумерного массива климатических данных. При выполнении операций, предусмотренных в состоянии 4, МК переходит в состояние 0, тем самым завершает рабочий цикл системы.

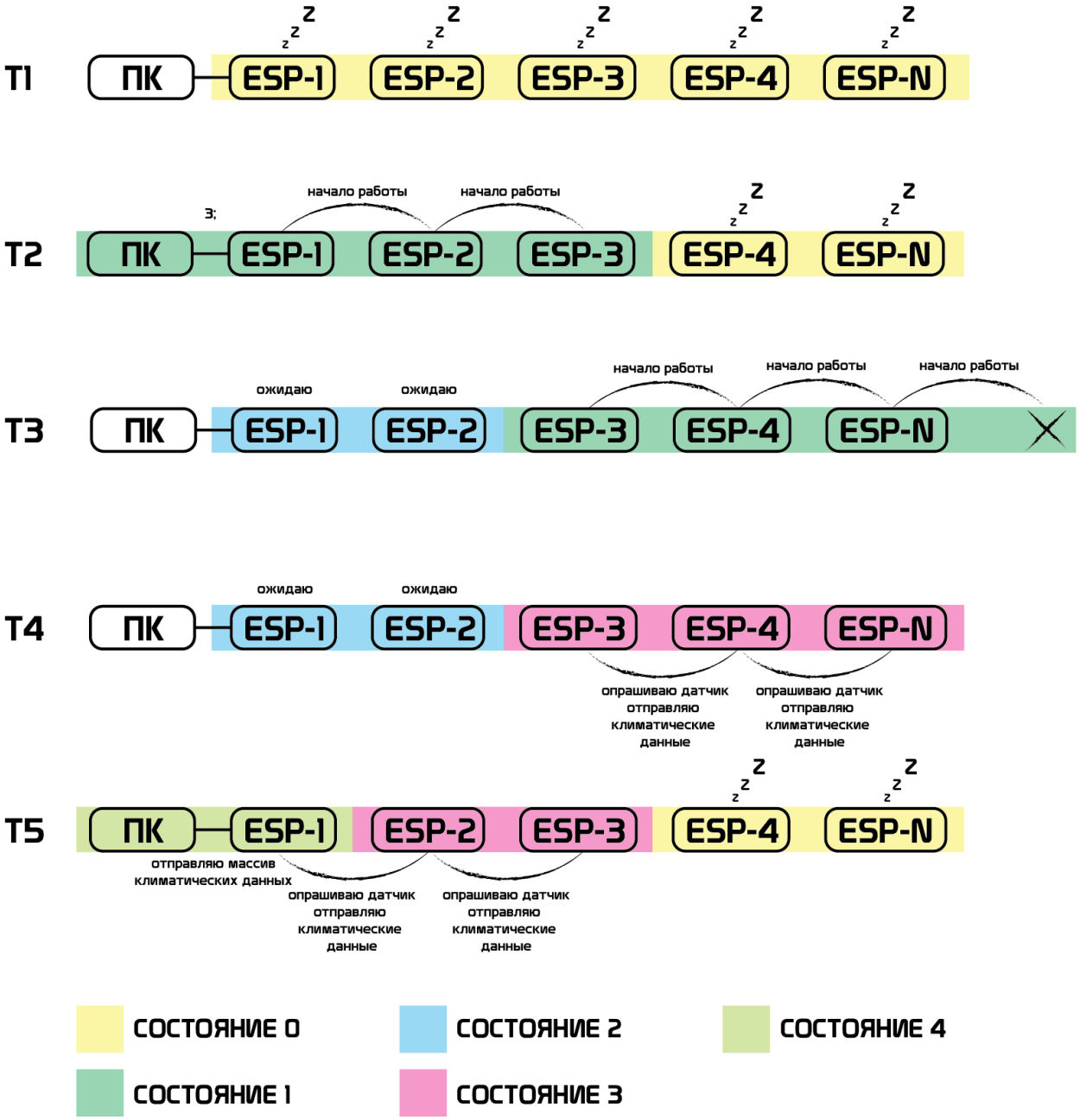


Рисунок 15 – Рабочий цикл системы

На рисунке 15 изображена система в пяти временных участках T1-T5:

– Во временном участке T1 все модули находятся в состоянии ожидания сигнала работы.

– Во временном участке T2 ось начинает работу в момент приема от ПК пакета, содержащего код операции начала, после чего начинается отправка передаваемой структуры последовательно от модуля к модулю.

– Во временном участке Т3 модули, передавшие пакет, переходят в состояние 2, а модуль ESP-N стакивается с ошибкой отправки на следующий модуль, что является командой к переходу в состояние 3.

– Во временном участке Т4 модули в обратном порядке начиная с ESP-N передают пакет, уже содержащий собственные климатические данные, а также данные из вышестоящих модулей.

– Во временном участке Т5 завершается передача обратного пакета, содержащего все собранные климатические данные, которые передаются на ПК через COM-порт. Модули, передавшие пакет, переходят в состояние 0, тем самым завершая рабочий цикл системы.

2.2 Разработка механизмов записи MAC-адресов

В системе должно быть предусмотрено добавление новых устройств. Для этого в среде Arduino IDE предусмотрена операция возврата MAC-адреса Wi-Fi модуля ESP8266. В библиотеке «ESP8266WiFi.h» существует функция «WiFi.macAddress();», которая возвращает MAC-адрес данного устройства. MAC-адрес вида «E8:DB:84:A4:4C:A7» записывается в переменную типа String. В конце операции значение переменной записывается в COM-порт для передачи этой информации в программное обеспечение на ПК, где адрес добавляется в таблицу адресов оси радиосвязи.

После внесения изменений в таблицу MAC-адресов данные об адресах необходимо передать на головное устройство. Для этого необходимо сформировать строку типа «код операции, первый MAC-адрес, второй MAC-адрес,.., n-MAC-адрес;». При формировании строки необходимо соблюдать правила:

– Последовательность MAC-адресов должна в точности повторять последовательность устройств в оси радиосвязи от головного устройства к последнему.

– Код операции и MAC-адреса должны быть разделены запятыми, а строка должна оканчиваться на точку с запятой.

– Внутри кода операции и MAC-адресов не должно встречаться символов «,» и «;».

– Строка в COM-порт может записываться частями, однако время записи не должно превышать 30 секунд. Это правило диктуется функцией ожидания данных из COM-порта «Serial.setTimeout(30000);», меняя аргумент которой можно изменить время ожидания окончания строки.

– Головное устройство, приняв строку, должно распарсить принятую строку, перевести MAC-адреса в числовой тип данных и записать их в двухмерный массив MAC-адресов.

2.3 Разработка принципов структурирования данных

Умение свободно работать с потоком данных необходимо для оптимальной организации взаимодействия МК с приложением на ПК. Общение через COM-порт подразумевает передачу символьных данных, так как текст позволяет передавать любые данные, эти данные легко подготовить к отправке и так же легко принять. В данной работе мы ходим управлять головным устройством с приложения на ПК, а также принимать с него собранные климатические данные. Если бы это было простым действием, управление свелось бы к передаче в COM-порт одной цифры, однако в данной работе необходимо передавать большие объёмы разнородных данных, что требует организации протокола связи. Протокол связи однозначно определяет такую последовательность данных в пакете, чтобы оба устройства однозначно понимали, что они друг от руга хотят. Структура пакета данных, удовлетворяющих требованиям протокола изображена на рисунке 16.

<КЛЮЧ><РАЗДЕЛИТЕЛЬ><ЗНАЧЕНИЕ>
<РАЗДЕЛИТЕЛЬ><ЗНАЧЕНИЕ>...<ТЕРМИНАТОР>

Рисунок 16 – Структура пакета данных

Данная структура пакета подразумевает приём сразу N значений. Первостепенная задача – принять данные в буфер. Для этого удобно использовать функцию Serial.readBytesUntil(‘;’, data, 30), где ‘;’ – терминатор, data – массив символов типа char, 30 – размер массива data, указывать его необходимо, чтобы функция не записывала данные в массив, при его переполнении.

Получив пакет данных, его нужно разделить на строки, для работы с каждой по-отдельности. Для этого воспользуемся функцией GParser data(str, ‘,’) из библиотеки GParser.h, которую следует установить в среду разработки Arduino IDE. Аргументами данной функции выступают: data – массив строк, в которые будут записаны разделенные данные, str – строка содержащая весть принятой пакет, ‘,’ – символ разделитель. Далее необходимо вызвать метод data.split(), который разделит строку str на подстроки и вернёт количество этих подстрок. Теперь к подстрокам можно обращаться как к объектам data[n]. Это очень удобно и решает все проблемы. Далее с подстроками можно работать индивидуально, а именно в данной работе как один из вариантов – преобразовывать строки в числовые переменные для использования в качестве MAC-адресов для функции отправки данных по Wi-Fi.

2.4 Разработка механизмов перевода строковых данных в числовые

Каждый микроконтроллер ESP8266 имеет свой MAC-адрес, и чтобы обращаться к данному МК необходимо звать его MAC-адрес. Функция WiFi.macAddress() возвращает MAC-адрес устройства в виде строкового типа данных. Эти данные возможно использовать в качестве аргумента функции отправки данных на это устройство по Wi-Fi, однако адрес, в таком случае, необходимо будет вводить вручную, присваивая её значения численной переменной, например типа данных int или uint8_t.

Для автоматизированного использования MAC-адресов необходимо приставить их в виде численной переменной. Проанализируем MAC-адрес типа «E8:DB:84:A4:4C:A7», видим, что числа представлены в

шестнадцатеричном виде, и числа разделены символом «:», который нужен лишь для визуального разделения последовательности чисел. По итогу нам необходимо получить одномерный массив типа int вида «0xE8,0xDB,0x84,0xA4,0x4C,0xA7», где «0x» является указателем на шестнадцатеричный вид представления числа.

Заведем цикл, тело которого будет выполняться 6 раз – в соответствии с количеством цифр в MAC-адресе. В начале каждой итерации будем объявлять строковую переменную типа char, а также присваивать её нулевому и первому элементу значения «0» и «x» соответственно. Во второй элемент переменной будет записан элемент строки, содержащий MAC-адрес с номером – номер итерации цикла, помноженный на три. Третий элемент записываем соответственно прошлому, но с номером итерации цикла, помноженном на три и с добавлением единицы. По итогу имеем переменную типа char, в которую записано «0xE8» для первой итерации цикла. Далее воспользуемся методом «atof» который преобразует полученную символьную информацию в числовую и присвоит полученное значение заранее объявленной численной переменной в элемент с номером, соответствующим номеру нынешней итерации цикла. На рисунке 17 представлен текст программы, рассмотренный выше.

```
MYMAC=WiFi.macAddress();
Serial.println(MYMAC);
   for (int i=0; i <= 5; i++){
   char stro[4];
   stro[0]='0';
   stro[1]='x';
   stro[2]=MYMAC[i*3];
   stro[3]=MYMAC[i*3+1];
   stro[4]=0;
   //Serial.println(stro);
   mymac[i]=atof(stro);
    }
```

Рисунок 17 – Пример преобразования строковой информации в числовую

При прохождении всех шести кругов цикла имеем MAC-адрес, пригодный для дальнейшего использования в качестве аргумента функции передачи данных по Wi-Fi.

2.5 Разработка принципов обмена данными между модулями системы

В основе обмена данными между МК используется протокол беспроводной передачи данных ESP-NOW из библиотеки «espnow.h». Для успешного установления связи и передачи данных необходимо, чтобы на всех устройствах в оси радиосвязи была организована единая структура данных, включающая в себя:

- двухмерный массив, содержащий климатические данные;
- двухмерный массив, содержащий таблицу MAC-адресов;
- целочисленная переменная состояния оси радиосвязи.

Передача пакета данных сводится к вызову функции передачи esp_now_send(MAC_UP, (uint8_t *) &PRD, sizeof(PRD)), где MAC_UP – адрес принимаемого, PRD – передаваемая структура. Также на каждом МК необходимо организовать работу функций подтверждения приёма и отправки. Синтаксис объявления функций изображен на рисунках 18 и 19.

```
// Callback отправка
void OnDataSent(uint8_t *mac_addr, uint8_t sendStatus) {
Serial.print("Статус: ");
if (sendStatus == 0){
Serial.println("отправлено");

}
else{
Serial.println("ошибка отправки");

PRM.STAT=3;
PRD.STAT=3;
}
}

// Callback приём
void OnDataRecv(uint8_t * mac, uint8_t *PRM_bytes, uint8_t len) {
memcpy(&PRM, PRM_bytes, sizeof(PRM));
Serial.print("байт принято: ");
Serial.println(len);
//PRM.STAT=3;
}
```

Рисунок 18 – Функции подтверждения приёма и отправки

```
esp_now_register_send_cb(OnDataSent);
esp_now_add_peer(MAC_UP, ESP_NOW_ROLE_COMBO, 1, NULL, 0);
esp_now_add_peer(MAC_DOWN, ESP_NOW_ROLE_COMBO, 1, NULL, 0);
esp_now_register_recv_cb(OnDataRecv);
```

Рисунок 19 – Регистрация callback-функций и пиров

Включение callback-функций в программу необходимо для корректной работы оси радиосвязи при выходе какого-либо модуля из строя. Это позволяет ограничить ось до последнего корректно работающего модуля и успешно завершить рабочий цикл системы.

2.6 Разработка принципов обмена данными с компьютером

Для общения с МК в процессе работы системы предусмотрен ряд кодов операций:

– Код «1» предназначен для записи в оперативную память МК таблицы MAC-адресов.

– Код «2» предназначен для вывода в COM-порт таблицы адресов

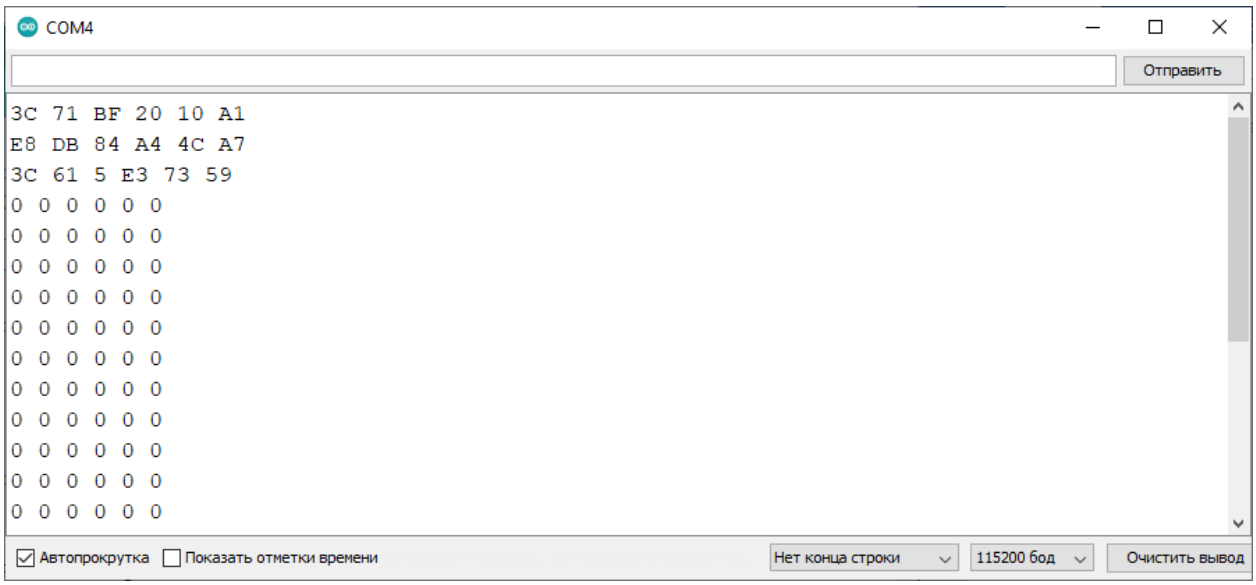


Рисунок 20 – Информация, выводимая в COM-порт по коду операции 2

– Код «3» предназначен для инициализации начала работы оси радиосвязи. Информация, возвращаемая МК по коду операции 3, продемонстрирована на рисунке 21.

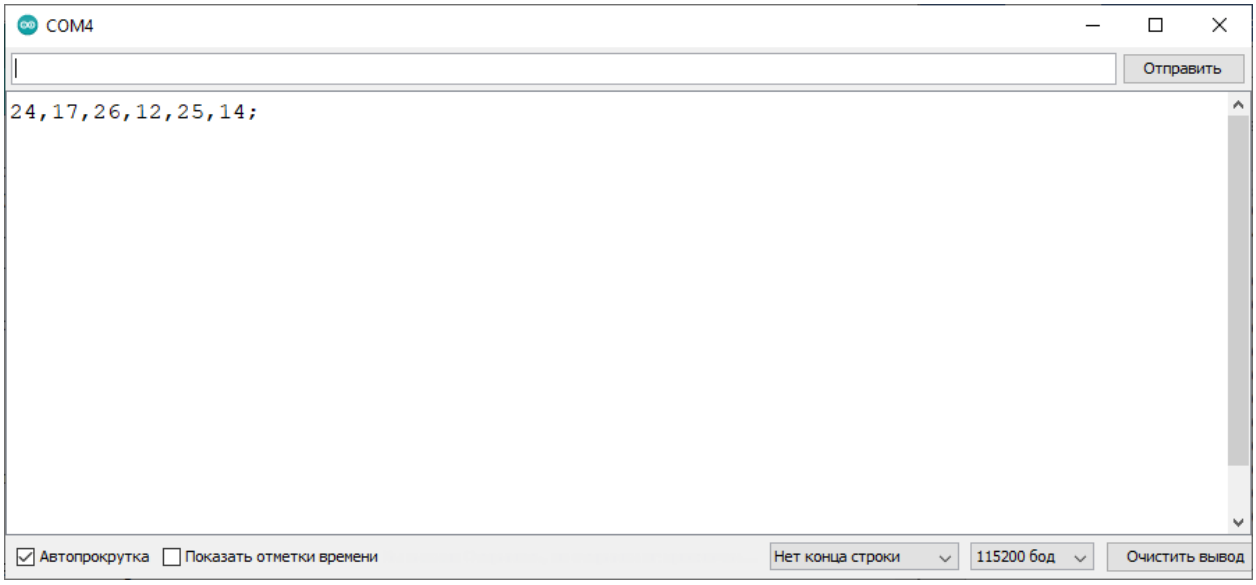


Рисунок 21 – Информация, выводимая в COM-порт по коду операции 3

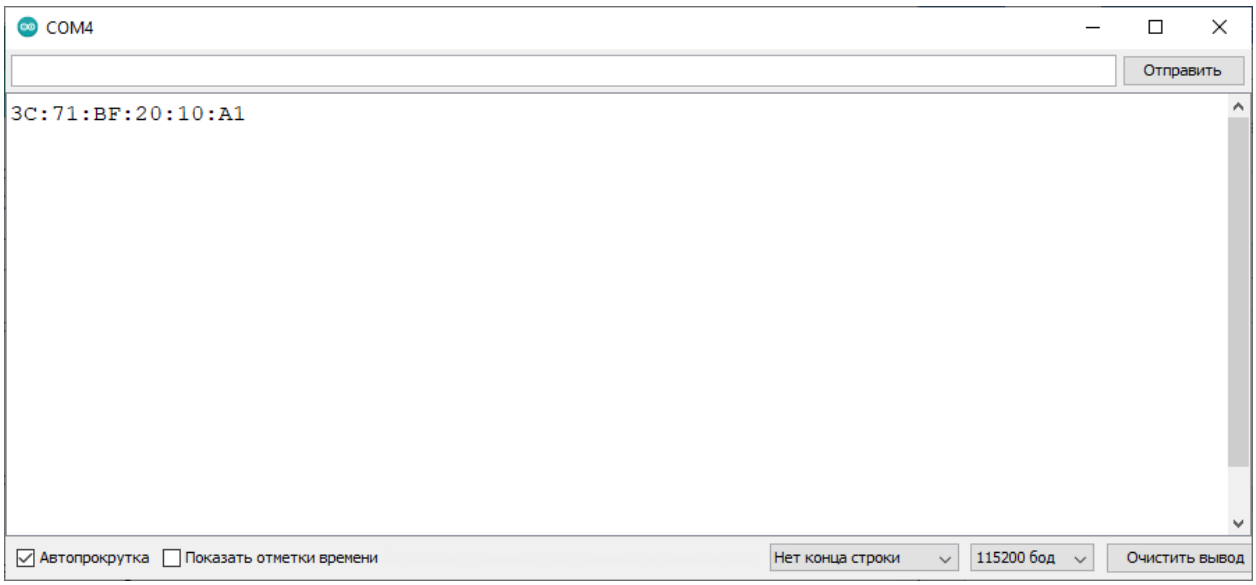


Рисунок 22 – вывод MAC-адреса в COM-порт

– Код «4» предназначен для вывода в COM-порт строки, содержащей MAC-адрес устройства. Операция предназначена для добавления MAC-адресов в базу данных на ПК. Отклик МК на код операции 4 продемонстрирован на рисунке 22.

2.7 Разработка блок-схем программы для ESP8266

Блок-схема объединяет все вышеописанные принципы и решения и является одним из завершающих шагов в разработке программного обеспечения для МК ESP8266 на плате NodeMCUv3 с периферией в виде датчика температуры и влажности DHT11. На рисунке 23 изображена обобщенная блок-схема, содержащая описание участков кода, ответственных за подключение библиотек, объявление переменных, объявление типа датчика и пина общения с ним, а также объявление структуры и переменные, входящие в неё.

Рисунок 23 – Обобщенная блок-схема программы для МК

На рисунках 24 и 25 изображены подробные блок-схемы callback функций, соответствующие callback функциям на блок-схеме, изображенной на рисунке 23.

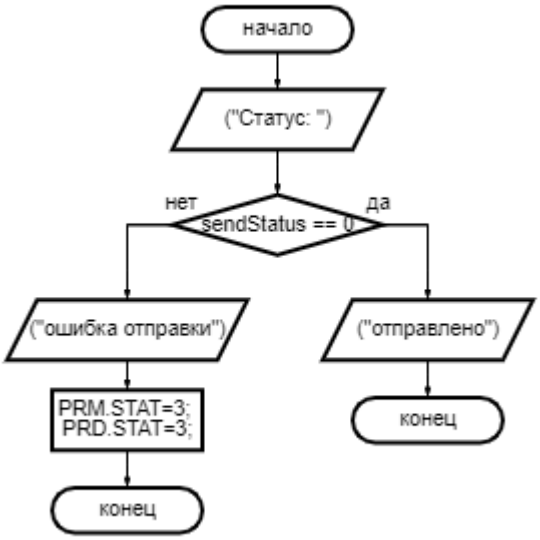


Рисунок 24 – Блок-схема функции отправки callback

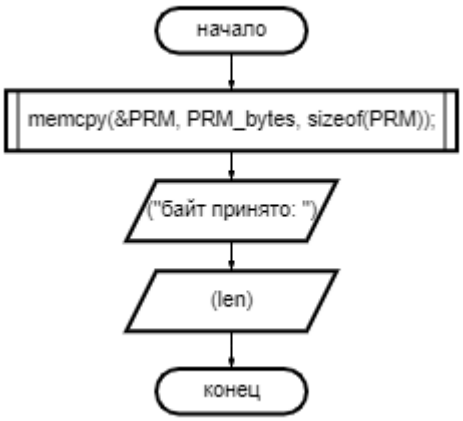


Рисунок 25 – Блок-схема функции приёма callback

В функции setup(), изображенном на рисунке 26, содержится блок инструкций, выполняющихся один раз при каждом запуске МК. В данном блоке настраивается время ожидания информации из COM-порта, инициализируется работа COM-порта на скорости 115200 бод, инициализируется работа датчика DHT11, настраивается режим работы W-Fi модуля, инициализируются callback функций и создание пиров (Рисунок 19), определяется собственный MAC-адрес с преобразованием его в числовое представление.

Рисунок 26 – Блок-схема функции setup()

В функции loop() реализован механизм общения с ПК через COM-порт с использованием парсинга данных. В зависимости от принятого ключа операции, записанного в объект data[0], программа переходит в один из блоков: 1 – запись таблицы адресов, определение своего места в таблице и запись MAC-адресов «соседей». 2 – вывод в COM-порт таблицы адресов, используется для проверки правильности работы алгоритмов. 3 – инициализация работы оси радиосвязи. 4 – вывод в COM-порт своего MAC-адреса, используется для добавления нового устройства в базу данных на ПК.

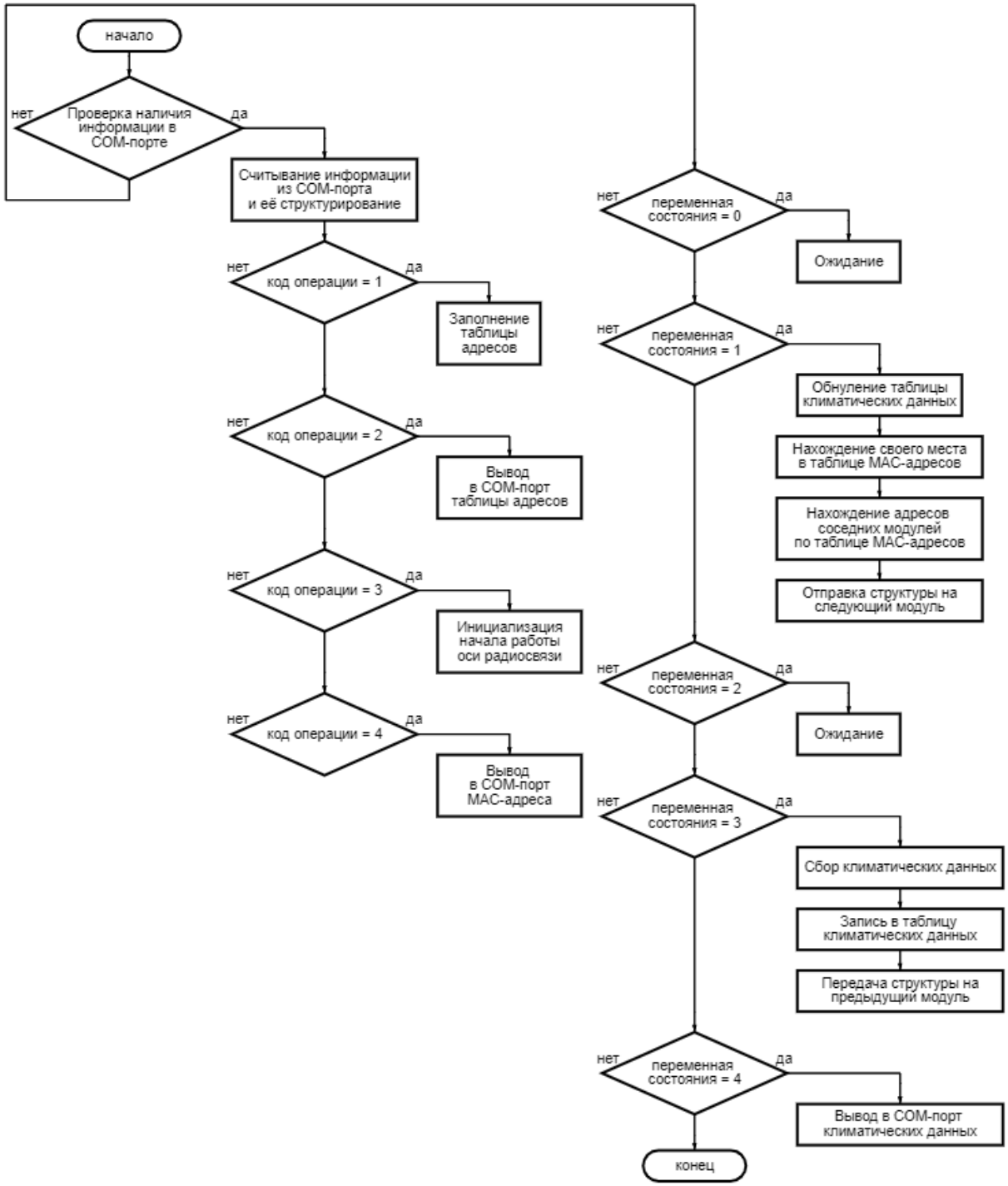


Рисунок 27 – Блок-схема функции loop()

Далее в Блок-схеме функции loop показаны инструкции, выполняемые во время конкретного состояния оси радиосвязи. Состояние оси хранится в переменной состояния.

2.8 Разработка 3D-модели корпуса устройства

Корпус устройства планируется напечатать на 3D-принтере, для чего необходимо разработать и смоделировать виртуальную модель. Для моделирования корпуса была использована среда моделирования Autodesk Fusion 360 (Рисунок 28). Внутри корпуса необходимо выделить места крепления для модуля МК Lolin и модуля питания Battery Shield. Для управления питанием и зарядки аккумуляторной батареи необходимо предусмотреть технологические отверстия.

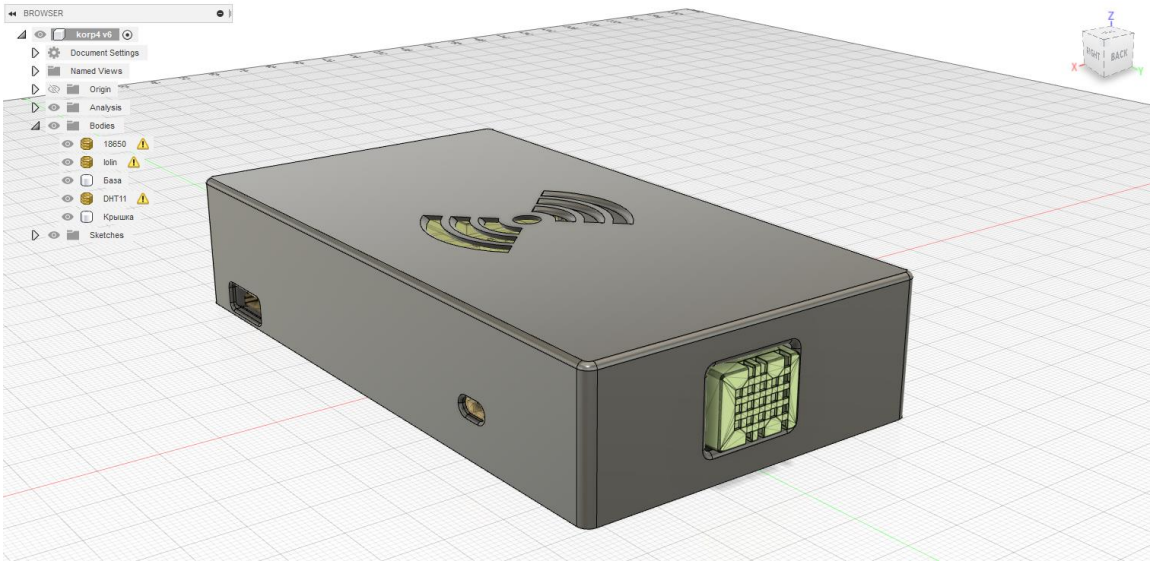


Рисунок 28 – 3D-модель корпуса устройства

Так же технологическое отверстие необходимо для выноса датчика температуры и влажности DHT11 лицевой стороной за пределы корпуса для его корректного считывания климатических параметров микроклимата помещения с внесением минимальной погрешности от теплового выделения внутренний электроники устройства.

Корпус должен состоять из двух частей – нижней части корпуса, на которой будут закреплены электронные модули (Рисунок 29), и верхней части – защитной крышки, с технологическими отверстиями, служащими как для рассеивания теплового излучения внутренней электроники, так и в качестве декоративного элемента.

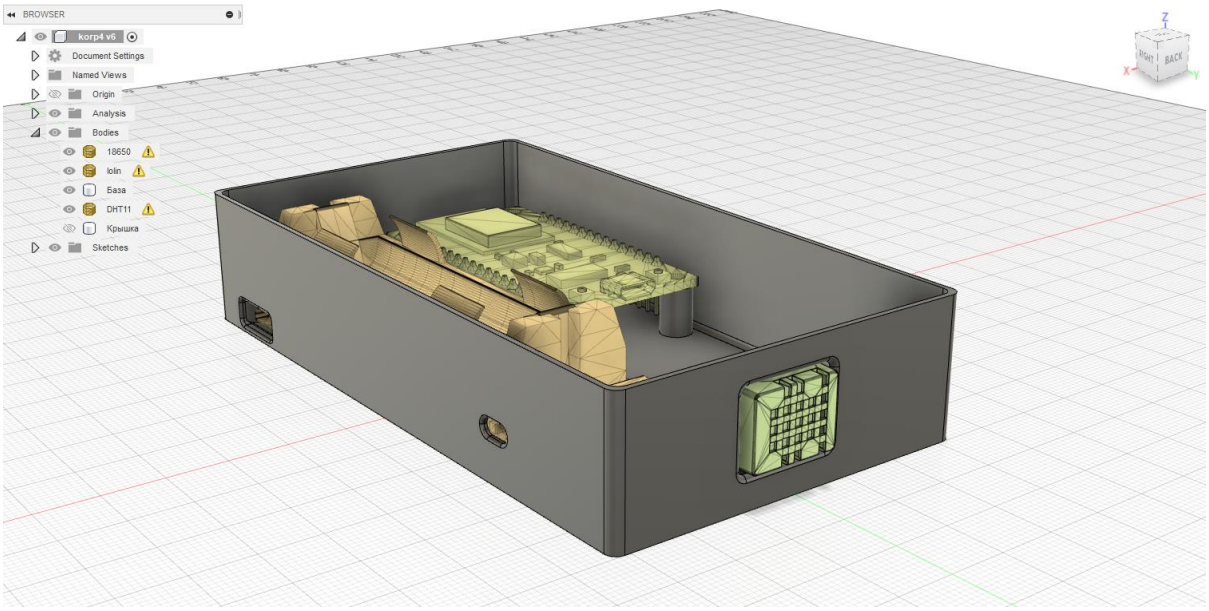


Рисунок 29 – 3D-модель корпуса устройства без верхней крышки

Рисунок 30 – Визуализация корпуса устройства

Так же среда проектирования Fusion 360 позволяет выполнить высококачественную визуализацию корпуса устройства, представленную на рисунке 30.

2.9 Расчет стоимости устройства

Для расчета стоимости обратились к наиболее популярным магазинам как в Томске, так и в России. В таблице 1 приведены собранные данные по ценам в выбранных магазинах на момент декабря 2021 года.

Таблица 1. Сравнение цен в наиболее популярных магазинах.

Магазин / Товар	Chipdip, руб	Амперка, руб	Озон, руб	AliExpress, руб	Эльград, руб
DHT11	690	310	369	60	185
NodeMCUv3	1950	___	1250/3	250	___
Battery Shield	___	___	353	154	448
Модуль RGB-светодиода	150	60*	300	103	15*

* - цена за светодиод

Анализируя вышеуказанную таблицу, получаем, что стоимость компонентов, необходимых сбора для одного устройства варьируется от 570 до 3390 рублей. Важно заметить, что эта стоимость не учитывает траты на дополнительное коммутационное оборудование (наборы проводников), печать корпуса устройства и транспортные издержки.

2.10 Расчет энергопотребления устройства

Автономность – это способность устройства работать от одного полного заряда батареи до её полной разрядки. Автономность измеряется в часах работы устройства от одного заряда. Для расчета автономности необходимо знать как потребление компонентов устройства, так и характеристики источника питания.

В данной работе планируется использовать АКБ ёмкостью 3400мАч. Максимальное потребление датчика температуры и влажности DHT11 во время сбора и отправки данных составляет 2.5мА при напряжении питания 3.3В. Потребление микроконтроллера в режиме ожидания 20мА, в режиме приёма данных по Wi-Fi – 60мА, а при передаче данных по Wi-Fi – до 200мА при напряжении питания 5В.

Для расчета времени автономной работы устройства воспользуемся формулой - Время работы аккумулятора = (10 * емкость батареи) / нагрузка, где ёмкость измеряется в Ач, а нагрузка в Вт.

Будем считать, что в сутки устройство будет производить 3 измерения. Время одного цикла считывания и отправки климатических данных занимает 400 мс из которых 100 мс затрачивается на приём данных, 100 мс на передачу данных, 100 мс на опрос датчика температуры и влажности и 100 мс обработку информации в МК и вывод её в COM-порт. Итого имеем, что в среднем в сутки потребление складывается из:

– 300 мс потребление устройства составляет 0.10825 Вт при опросе датчика;

– 300 мс потребление устройства составляет 0.3 Вт при приёме информации по Wi-Fi;

– 300 мс потребление устройства составляет 1 Вт при оправке информации по Wi-Fi;

– 300 мс потребление устройства составляет 0.1 Вт при обработке информации и выводе её в COM-порт.

Получаем среднее потребление в 0,38 Вт за время 1.2сек, соответственно за оставшиеся 86398,8 сек потребление составляет 0.1 Вт. Из всего вышерасчитанного можем вычислить среднее энергопотребление устройства за сутки, которое составляет 0.1024 Вт. Подставляя полученные и известные данные в формулу для расчета времени автономной работы, получаем что теоретическое время автономной работы составляет почти 14 суток.

2.11 Разработка схемы подключения устройства

Схема подключения устройства — это схема электрических соединений, выполненная в развернутом виде. Она является основной схемой проекта электрооборудования производственного механизма и дает общее представление об электрооборудовании данного механизма, отражает работу системы автоматического управления механизмом, служит источником для

составления схем соединений и подключений, разработки конструктивных узлов и оформления перечня элементов. Разработанная принципиальная схема устройства изображена на рисунке 31.

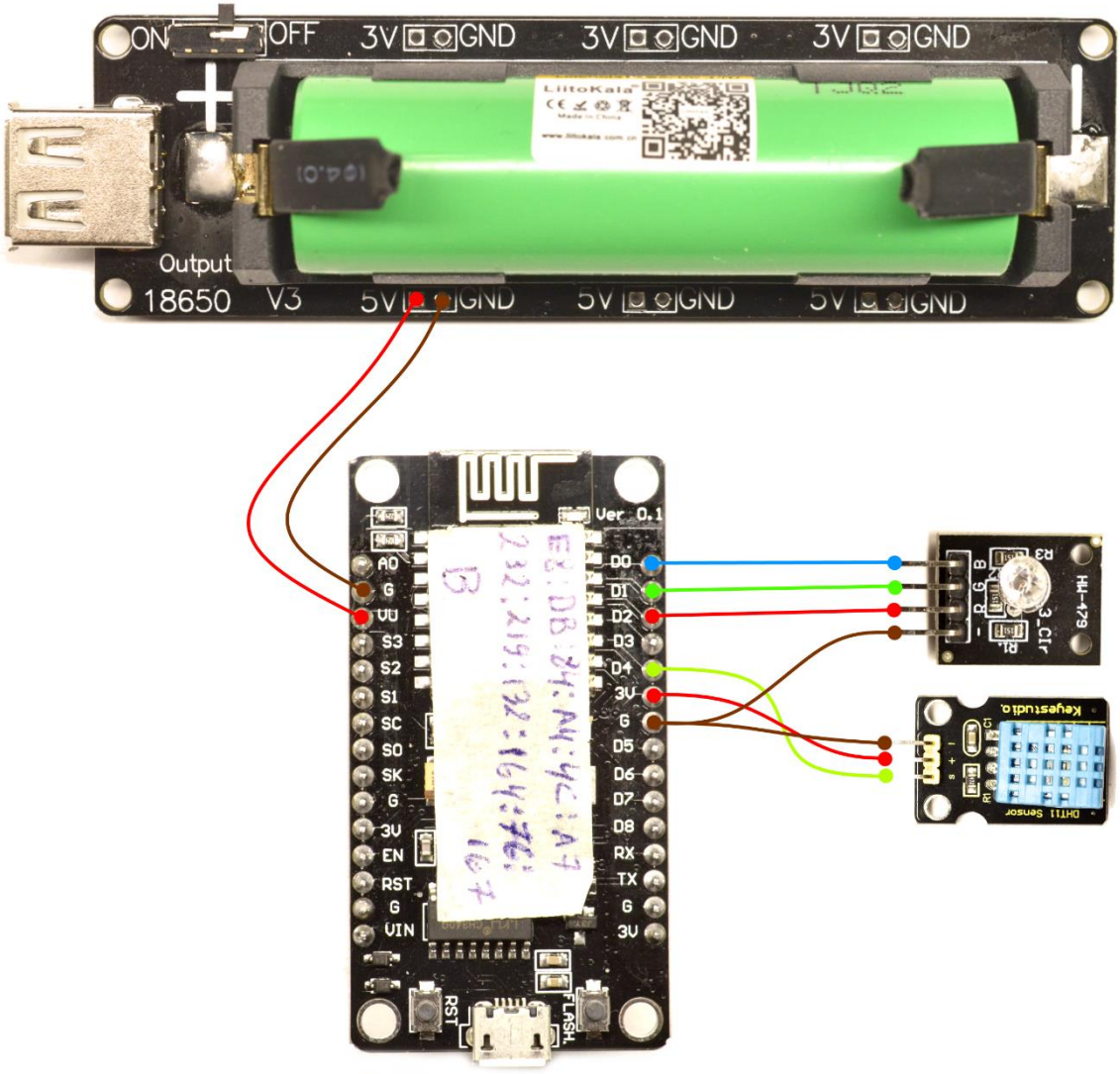


Рисунок 31 – Схема подключения устройства

3 Реализация устройства сбора климатических параметров

3.1 Реализация программы взаимодействия с DHT11

Основной задачей при создании устройства сбора и обработки климатических данных является написание программы для обмена данными между датчиком температуры и влажности DHT11 и микроконтроллером ESP8266. На рисунке 32 представлена блок-схема программы для работы с датчиком.

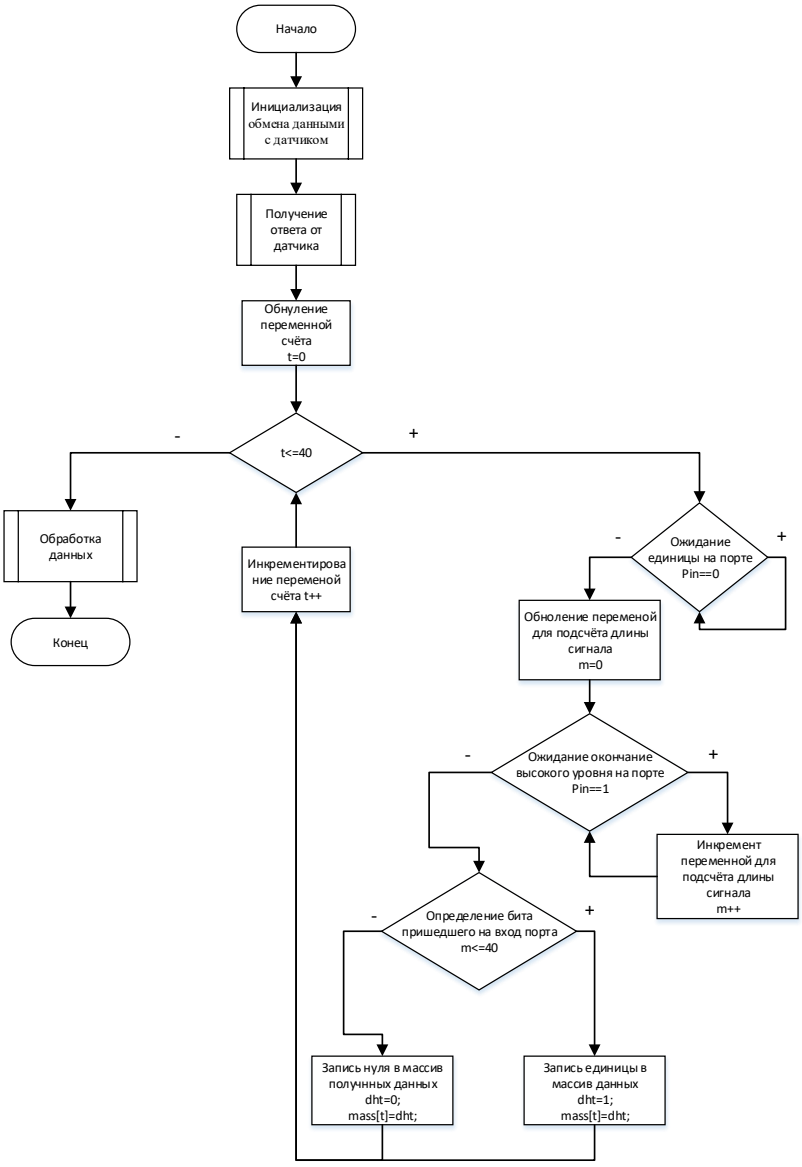


Рисунок 32 – Блок-схема программы обмена данными с датчиком

В начале передачи данных МК необходимо опустить шину данных в низкий уровень на 18 мс, а затем отпустить шину. Эта операция даёт понять датчику о готовности принятия данных и называется инициализацией. МК переходит в режим чтения и ожидает ответ от датчика. Ответ датчика представляет собой преамбулу с периодом 160 мкс, которая утверждает о готовности передачи данных. МК обнуляет переменную счёта битов t. Передача бита начинается с опускания шины данных, МК ожидает единичного значения на порте и обнуляет переменную счёта длинны сигнала m. При подаче на порт высокого уровня МК запускает цикл инкрементирования переменной m с задержкой времени 10 мкс. Инкрементирование продолжается до тех пор, пока на порт не поступит низкий уровень сигнала. При выходе из цикла условия происходит сравнение переменой m с числом 4. Если m < 4, то длительность импульса менее 40 мкс и был передан нуль. Если m > 4, то длительность импульса более 40 мкс и была передана единица. Далее происходит инкрементирование переменной счёта битов и начинается передача очередного бита. Достижение переменной счёта битов t = 40 говорит о приёме всего пакета данных от датчика и МК переходит к обработке данных.

3.2 Реализация работы с DHT11

В среде разработки Arduino IDE была написана программа, реализующая обмен данными между датчиком и МК используя интерфейс передачи данных «1-wire». Алгоритм работы программы изображен на рисунке 32, а сама программа на рисунке 33.

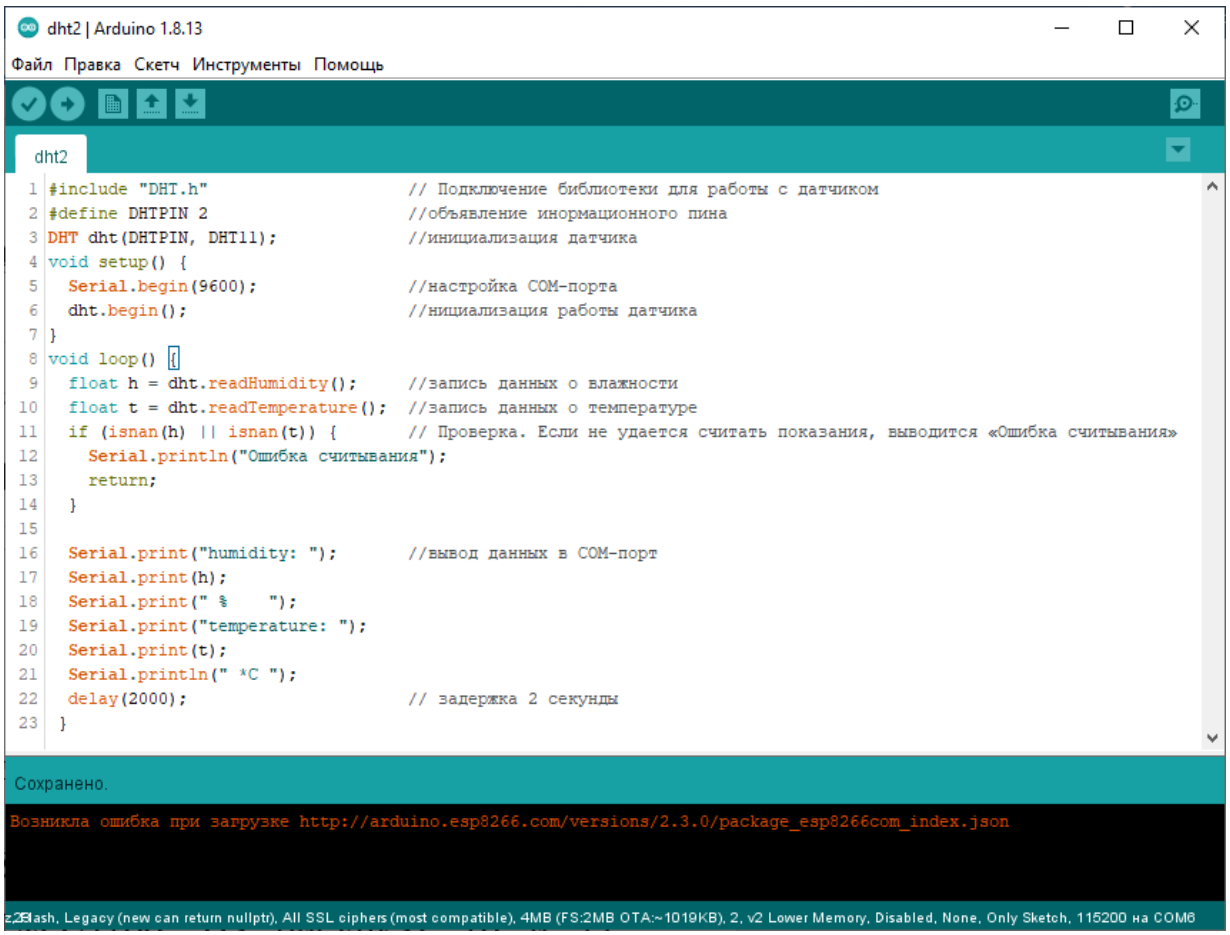


Рисунок 33 – Схема подключения датчика к микроконтроллеру

В результате эксперимента был запущен и опробован механизм опроса датчика DHT11 микроконтроллером ESP8266 на плате «lolin». В ходе проверки работоспособности системы были сняты показания как при комнатных (рисунок 34), так и в нестандартных условиях (рисунки 35, 36). Из рисунков 35 и 36 видно, что датчик температуры и влажности способен производить измерение вне рабочего диапазона, указанного производителем.

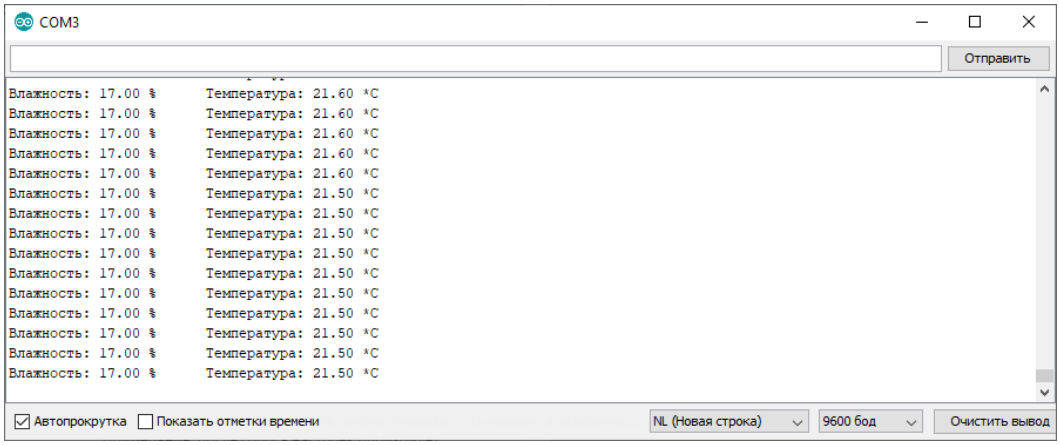


Рисунок 34 – Значения температуры и влажности при комнатных условиях

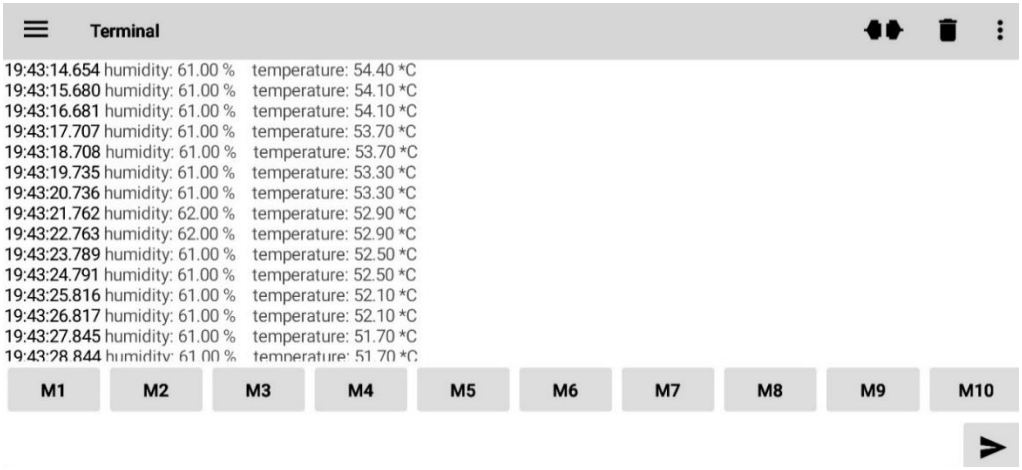


Рисунок 35 – Значения температуры и влажности в среде повышенной температуры и влажности

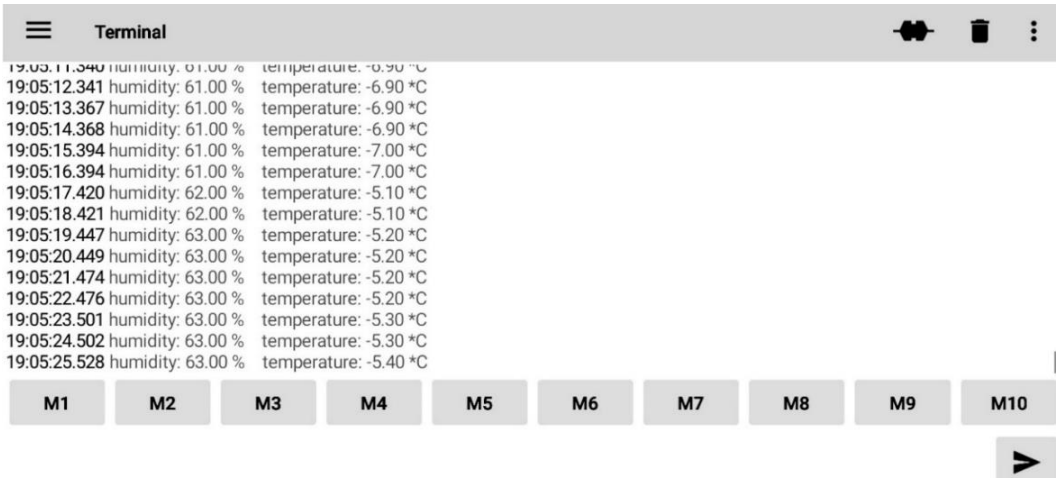


Рисунок 36 – Значения температуры и влажности в среде пониженной температуры

3.3 Реализация записи MAC-адресов в таблицу адресов на микроконтроллере

Запись MAC-адресов в оперативную память МК ESP8266 подразумевает выполнение ряда операций:

– приём пакета на МК;
– разделение пакета на ключ операции и подстроки;
– преобразование MAC-адресов из строкового вида в числовой;
– запись MAC-адресов в таблицу;
– определение своего положения в таблице адресов;

– определение адресов соседствующих модулей по таблице адресов.

Приём пакета адресов осуществляется вызовом функции Serial.readBytesUntil, которая будет вызвана при наличии в COM-порте полезной информации. Запись пакета осуществляется в переменную str типа char и остановится при приёме символа «;». Код, реализующий вышесказанное, представлен на рисунке 37.

```
if (Serial.available() > 0){
  char str[99];
  int amount = Serial.readBytesUntil(';', str, 99);
  str[amount] = NULL;
  GParser data(str, ',');
  int am=data.split();
  data.getInt(0);
```

Рисунок 37 – Приём пакета на МК

Заполненную строковую переменную необходимо разложить на подстроки и выделить из первой подстроки ключ операции. Разделение на подстроки реализуются вызовом функций GParser data(str, ',') и int am=data.split(), изображенных на рисунке 37, где переменная am показывает количество подстрок. Для выделения ключа операции необходимо первую подстроку преобразовать в числовой вид, используя функцию data.getInt(0).

Алгоритм преобразования MAC-адресов из строкового вида в числовой изображен на рисунке 17. Преобразованный адрес поэлементно записывается в таблицу адресов операцией PRD.MACs[i][g]=atof(stro).

Для нахождения своего места в таблице адресов необходимо поэлементно сравнивать адреса из таблицы адресов с собственным MAC-адресом, полученным по алгоритму, изображенному на рисунке 7. Определив своё место в таблице не составит труда и определение адресов соседей, реализуемый поэлементной записью адресов из вышестоящей и нижестоящей строки таблицы адресов в переменные для хранения адресов MAC_DOWN[] и MAC_UP[]. Код, реализующий этот алгоритм представлен на рисунке 39, а информация, передающаяся в COM-порт, на рисунке 38.

```
15:47:47.014 Connected to CH34x device
15:48:04.530 1,3C:71:BF:20:10:A1,E8:DB:84:A4:4C:A7,3C:
61:05:E3:73:59;
15:48:04.592 В таблице я под номером 1
15:48:04.592 Адрес прошлого 0 0 0 0 0 0
15:48:04.592 Адрес следующего E8 DB 84 A4 4C A7
```

Рисунок 38 – Демонстрация работы алгоритма записи MAC-адресов

```
for (int i=1; i <= 5; i++){
  if (PRM.MACs[i][0]==MYMAC[0]&&
      PRM.MACs[i][1]==MYMAC[1]&&
      PRM.MACs[i][2]==MYMAC[2]&&
      PRM.MACs[i][3]==MYMAC[3]&&
      PRM.MACs[i][4]==MYMAC[4]&&
      PRM.MACs[i][5]==MYMAC[5])
   {
    Serial.print("В таблице я под номером ");
    Serial.println(i);
    MyN=i;
    for (int g=0;g<=5;g++){
      MAC_DOWN[g]=PRM.MACs[i-1][g];
      MAC_UP[g]=PRM.MACs[i+1][g];
    }
    Serial.print("Адрес прошлого ");
    for (int k=0;k<=5;k++){
      Serial.print(MAC_DOWN[k],HEX);
      Serial.print(" ");
    }
    Serial.println();

    Serial.print("Адрес следующего ");
    for (int k=0;k<=5;k++){
      Serial.print(MAC_UP[k],HEX);
      Serial.print(" ");
    }
    Serial.println();
   }//if
}
```

Рисунок 39 – Определение MAC-адресов «соседей»

3.4 Реализация вывода MAC-адресов для записи в базу данных на компьютере

Запись MAC-адресов в базу данных производится по коду операции 4. МК, приняв пакет «4;» переходит в часть кода, соответствующую ключу 4, изображенную на рисунке 40.

```
case 4:
Serial.println(mymac);
break;
```

Рисунок 40 – Вывод адреса модуля в COM-порт

В переменной mymac хранится строковое представление MAC-адреса модуля, которое записывается операцией mymac=WiFi.macAddress(), находящейся в блоке кода void setup() при каждом запуске модуля.

Также в программе предусмотрен вывод в COM-порт всей таблицы адресов, реализуемый кодом, представленным на рисунке 41, и продемонстрированный на рисунке 42.

```
case 2: //вывод таблицы адресов
  for (int g = 1; g <= 23; g++) {
    for (int k = 0; k <= 5; k++) {
      Serial.print(PRD.MACs[g][k], HEX);
      Serial.print(" ");
    }
    Serial.println();
  }
  break;
```

Рисунок 41 – Вывод в COM-порт таблицы адресов

```
15:48:12.066 2;
15:48:12.093 3C 71 BF 20 10 A1
15:48:12.093 E8 DB 84 A4 4C A7
15:48:12.093 3C 61 5 E3 73 59
15:48:12.093 0 0 0 0 0 0
15:48:12.093 0 0 0 0 0 0
```

Рисунок 42 – Пример вывода таблицы адресов в COM-порт

3.5 Реализация корпуса устройства

В ходе выполнения работы корпус устройства был спроектирован в среде Autodesk Fusion 360 и распечатан на 3D принтере с использованием ABS-пластика. Распечатанный корпус изображен на рисунках 43 и 44.

Рисунок 43 – Изготовленный корпус устройства

Рисунок 44 – Изготовленный корпус устройства в раскрытом состоянии

3.6 Реализация кода программы для микроконтроллера

Код программы для МК включает в себя все вышеописанные механизмы. На рисунке 45 изображен участок кода программы для МК, отвечающий за подключение библиотек, объявление переменных и структуры для работы системы.

D4_Clean

```
#include <ESP8266WiFi.h>
#include <espnow.h>
#include <GParser.h>
#include "DHT.h"
DHT dht(2, DHT11);

int MyN;
String mymac;
uint8_t MYMAC[6];
uint8_t MAC_DOWN[6];
uint8_t MAC_UP[6];
uint32_t TM;
boolean Red;
boolean Green;
boolean Blue;

typedef struct struct_message {
  uint8_t DATA[24][2];
  uint8_t MACs[24][6];
  uint8_t STAT;
} struct_message;
struct_message PRD;
```

Рисунок 45 – выдержка из кода программы для МК

Подключенные библиотеки отвечают за работу с Wi-Fi модулем, структурирование данных и работу с датчиком DHT11. Переменная MyN хранит порядковый номер устройства системе. Переменные MYMAC, MAC_DOWN, MAC_UP хранят MAC-адреса, TM – переменная, хранящая время с момента запуска устройства, а переменные Red, Green и Blue – состояния компонент RGB-светодиода. Структура PRD содержит массив для хранения климатических данных, массив для хранения MAC-адресов и переменную состояния.

На рисунке 46 изображено объявление Callback функций с управлением RGB-светодиодом в зависимости от события.

```
// Callback отправка
void OnDataSent(uint8_t *mac_addr, uint8_t sendStatus) {
  //  Serial.print("Статус: ");
  if (sendStatus == 0) {
    //    Serial.println("отправлено");
    Blue = 1;
  }
  else {
    //    Serial.println("ошибка отправки");
    PRD.STAT = 3;
    Red = 1;
  }
}

// Callback приём
void OnDataRecv(uint8_t * mac, uint8_t *PRD_bytes, uint8_t len) {
  memcpy(&PRD, PRD_bytes, sizeof(PRD));
  //  Serial.print("байт принято: ");
  //  Serial.println(len);
  Green = 1;
}
```

Рисунок 46 – выдержка из кода программы для МК

На рисунке 47 показана настройка Com-порта, инициализация работы датчика и настройка Wi-Fi модуля. Также показан механизм определения MAC-адреса и его конвертация в численный вид.

```
void setup() {
  Serial.setTimeout(1000);
  Serial.begin(115200);
  dht.begin();
  float t = dht.readTemperature();
  float h = dht.readHumidity();
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();

  if (esp_now_init() != 0) {  // Инициализируем протокол ESP-NOW
    Serial.println("Error initializing ESP-NOW");
    return;
  }
  esp_now_set_self_role(ESP_NOW_ROLE_COMBO);  // Указываем роль платы в сети
  esp_now_register_send_cb(OnDataSent);       // Регистрируем callback-функцию для получен
  esp_now_add_peer(MAC_UP, ESP_NOW_ROLE_COMBO, 1, NULL, 0); // Регистрируем пиры
  esp_now_add_peer(MAC_DOWN, ESP_NOW_ROLE_COMBO, 1, NULL, 0); // Регистрируем пиры
  esp_now_register_recv_cb(OnDataRecv);       // Регистрируем callback-функцию для получен

  PRD.STAT = 0;

  mymac = WiFi.macAddress();
  Serial.println();
  for (int i = 0; i <= 5; i++) {
    char stro[4];
    stro[0] = '0';
    stro[1] = 'x';
    stro[2] = mymac[i * 3];
    stro[3] = mymac[i * 3 + 1];
    stro[4] = 0;
    MYMAC[i] = atof(stro);
  }
}
```

Рисунок 47 – выдержка из кода программы для МК

На рисунке 48 изображен механизм работы световой индикации событий.

```
void loop() {

  if (Red == 1) {
    analogWrite(4, 0);
    analogWrite(16, 1000);
    delay(500);
    analogWrite(16, 0);
    Red = 0;
  }

  if (Green == 1) {
    analogWrite(4, 0);
    analogWrite(5, 1000);
    delay(200);
    analogWrite(5, 0);
    Green = 0;
  }

  if (Blue == 1) {
    analogWrite(4, 1000);
    delay(200);
    analogWrite(4, 0);
    Blue = 0;
  }
```

Рисунок 48 – выдержка из кода программы для МК

На рисунке 49 представлен код, реализующий приём и структурирование данных, полученных их COM-порта.

```
  if (Serial.available() > 0) {
    char str[410];
    int amount = Serial.readBytesUntil(';', str, 410);
    str[amount] = NULL;
    GParser data(str, ',');
    int am = data.split();
    data.getInt(0);
```

Рисунок 49 – выдержка из кода программы для МК

На рисунке 50 изображена первая часть структуры обработки полученной информации, отвечающая за получение MAC-адресов от ПК, их конвертация в числовой вид и запись в таблицу MAC-адресов структуры PRD. Также реализуется механизм поиска своего места в системе и нахождения MAC-адресов соседних модулей.

```
114     switch (data.getInt(0)) {
115       case 1: //заполенеие таблицы адресов
116         for (int i = 1; i < am; i++) {
117           String MAC = data[i];
118           int mac[6];
119           for (int g = 0; g <= 5; g++) {
120             char stro[4];
121             stro[0] = '0';
122             stro[1] = 'x';
123             stro[2] = MAC[g * 3];
124             stro[3] = MAC[g * 3 + 1];
125             stro[4] = 0;
126             PRD.MACs[i][g] = atof(stro);
127           }
128         }
129
130         for (int i = 1; i <= 23; i++) {
131           if (PRD.MACs[i][0] == MYMAC[0] &&
132               PRD.MACs[i][1] == MYMAC[1] &&
133               PRD.MACs[i][2] == MYMAC[2] &&
134               PRD.MACs[i][3] == MYMAC[3] &&
135               PRD.MACs[i][4] == MYMAC[4] &&
136               PRD.MACs[i][5] == MYMAC[5])
137           {
138             MyN = i;
139             for (int g = 0; g <= 5; g++) {
140               MAC_DOWN[g] = PRD.MACs[i - 1][g];
141               MAC_UP[g] = PRD.MACs[i + 1][g];
142             }
143           }//if
144         }
145         break;
```

Рисунок 50 – выдержка из кода программы для МК

Во второй части структуры обработки полученной информации (Рисунок 51) реализованы механизмы вывода в COM-порт таблицы MAC-адресов, вывода в COM-порт своего MAC-адреса для добавления его в базу данных на ПК, а так же инициализации начала работы рабочего цикла оси радиосвязи.

```
147       case 2: //вывод таблицы адресов
148         for (int g = 1; g <= 23; g++) {
149           for (int k = 0; k <= 5; k++) {
150             Serial.print(PRD.MACs[g][k], HEX);
151             Serial.print(" ");
152           }
153           Serial.println();
154         }
155         break;
156
157       case 3:
158         PRD.STAT = 1;
159         TM = millis();
160         break;
161
162       case 4:
163         Serial.println(mymac);
164         break;
165
166     }
167   }
```

Рисунок 51 – выдержка из кода программы для МК

На рисунке 52 изображен код, выполняемый при состоянии оси радиосвязи 0.

```
switch (PRD.STAT) {
  case 0:
    analogWrite(4, 100);
    delay(100);
    analogWrite(4, 0);
    break;
```

Рисунок 52 – выдержка из кода программы для МК

На рисунке 53 изображен код, выполняемый при состоянии оси радиосвязи 1. Производятся операции по определению своего места в системе, определению MAC-адресов соседних модулей, обнулению таблицы климатических данных и передачи структуры PRD на следующий модуль в случае, если в таблице MAC-адресов существует адрес следующего устройства.

```
  case 1:
    analogWrite(4, 1000);
    delay(200);
    PRD.STAT = 1;
    analogWrite(4, 0);

    for (int i = 0; i <= 23; i++) {
      for (int g = 0; g <= 1; g++) {
        PRD.DATA[i][g] = 0;
        PRD.DATA[i][g] = 0;
      }
    }

    for (int i = 1; i <= 23; i++) {
      if (PRD.MACs[i][0] == MYMAC[0] &&
          PRD.MACs[i][1] == MYMAC[1] &&
          PRD.MACs[i][2] == MYMAC[2] &&
          PRD.MACs[i][3] == MYMAC[3] &&
          PRD.MACs[i][4] == MYMAC[4] &&
          PRD.MACs[i][5] == MYMAC[5])
      {
        MyN = i;
        for (int g = 0; g <= 5; g++) {
          MAC_DOWN[g] = PRD.MACs[i - 1][g];
          MAC_UP[g] = PRD.MACs[i + 1][g];
        }
      }//if
    }//for
    if (MAC_UP[0] == 0) {
      Serial.println("Я последний");
      PRD.STAT = 3;
    }
    else {
      esp_now_send(MAC_UP, (uint8_t *) &PRD, sizeof(PRD));
      // Serial.println("Отправляю вперед");
      PRD.STAT = 2;
    }
    break;
```

Рисунок 53 – выдержка из кода программы для МК

На рисунке 54 изображены операции, выполняемые при состоянии оси радиосвязи 2 и 3. При состоянии 3 выполняются операции по опросу датчика

температуры и влажности, записи климатических данных в структуру PRD и отправке её на прошлый модуль.

```
215     case 2:
216       //  Serial.println("жду");
217       for (int i = 1; i <= 1001; i = i + 5) {
218         analogWrite(4, i);
219         delay(1);
220       }
221       break;
222 
223     case 3:
224       analogWrite(4, 0);
225       // Serial.println("Отправляю назад");
226       PRD.DATA [MyN][0] = dht.readTemperature();
227       PRD.DATA [MyN][1] = dht.readHumidity();
228       PRD.STAT = 3;
229       if (MyN > 1) {
230         esp_now_send(MAC_DOWN, (uint8_t *) &PRD, sizeof(PRD));
231       }
232       PRD.STAT = 4;
233       break;
```

Рисунок 54 – выдержка из кода программы для МК

На рисунке 55 изображены операции, выполняемые при состоянии 4 – вывод в COM-порт собранных климатических данных и переход устройства в состояние 0.

```
235     case 4:
236       // Serial.println("Вывожу на экран");
237       for (int i = 1; i <= 23; i++) {
238         if (PRD.MACs[i][0] > 0) {
239           Serial.print(PRD.DATA[i][0]);
240           Serial.print(",");
241           Serial.print(PRD.DATA[i][1]);
242           if (PRD.MACs[i + 1][0] > 0) {
243             Serial.print(",");
244           }
245           else {
246             Serial.print(";");
247           }
248         }
249       }
250       Serial.println();
251       PRD.STAT = 0;
252       break;
253   } //switch (PRD.STAT)
254 
255 }
```

Рисунок 55 – выдержка из кода программы для МК

Далее МК переходит в ожидание дальнейших указаний от ПК, что характеризуется слабым синим свечением RGB-светодиода.

3.7 Реализация устройств системы

В ходе практической реализации были закуплены необходимые компоненты (рисунок 56) для создания оси радиосвязи, состоящей из трех модулей. Каждый модуль включает в себя отладочную плату NodeMCUv3 под управлением МК ESP8266, датчик температуры и влажности DHT11, светодиод для индикации состояния работы МК, а также для всех модулей, кроме первого (подключенного непосредственно к ПК), плату контроля питания Wemos Battery Shield под АКБ 18650.

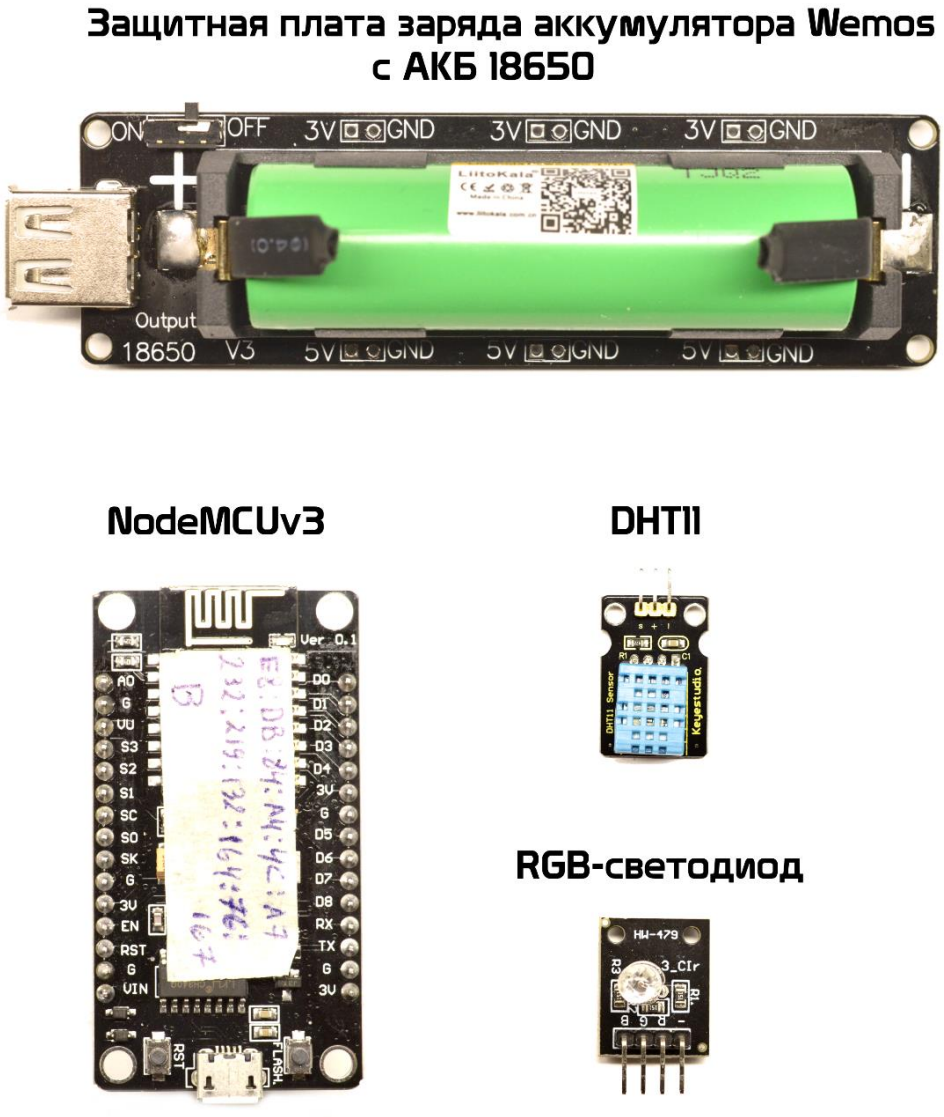


Рисунок 56 – Компоненты, необходимые для сборки устройства

Первое устройство, питающееся непосредственно от ПК, было собрано на беспаечной макетной плате. Устройство изображено на рисунке 57. Остальные устройства собраны с использованием комплекта проводов «мама-мама»

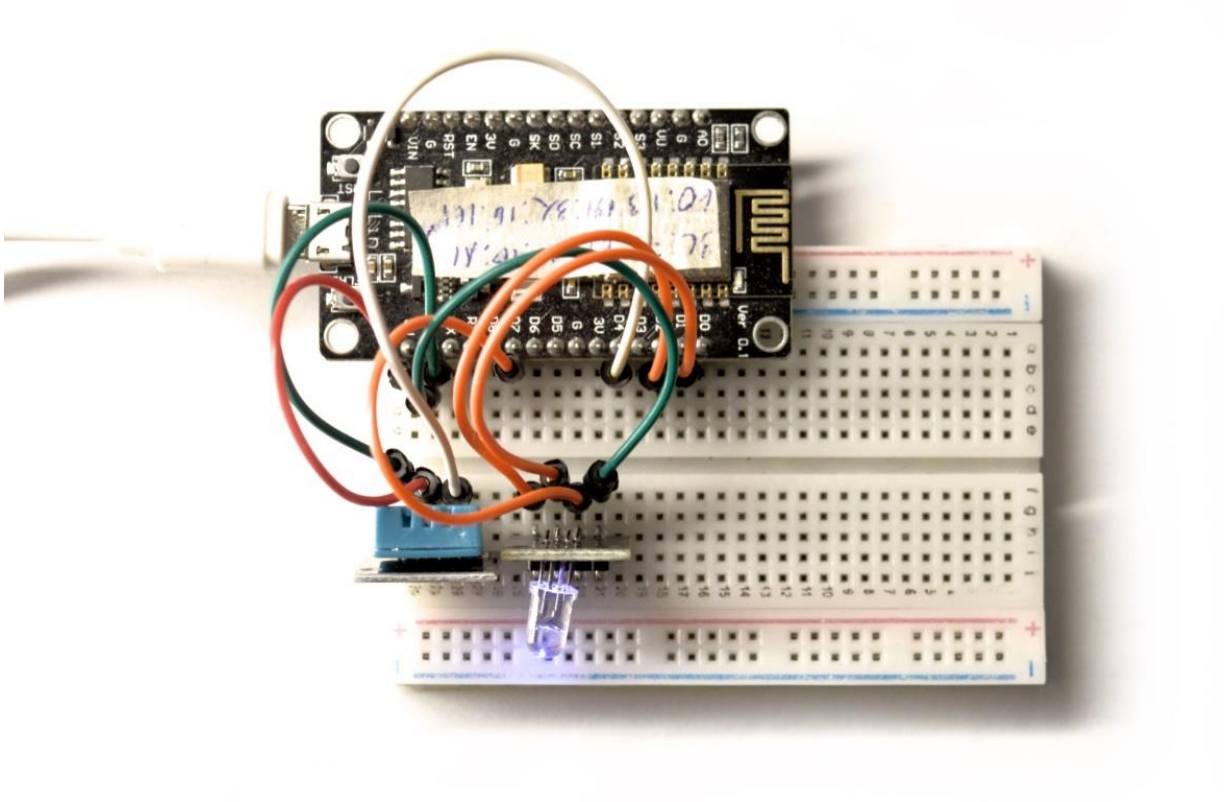


Рисунок 57 – Устройство, собранное на беспаечной макетной плате

4 Экспериментальные исследования системы

4.1 Реализация добавления новых устройств в базу данных на компьютере

Для добавления новых устройств в базу данных на ПК предусмотрена операция по коду 4. МК при получении пакета, содержащего «4;» возвращает в COM-порт свой MAC-адрес. Вывод адреса в COM-порт представлен на рисунке 58, а уже добавленный в базу данных адрес на рисунке 59.

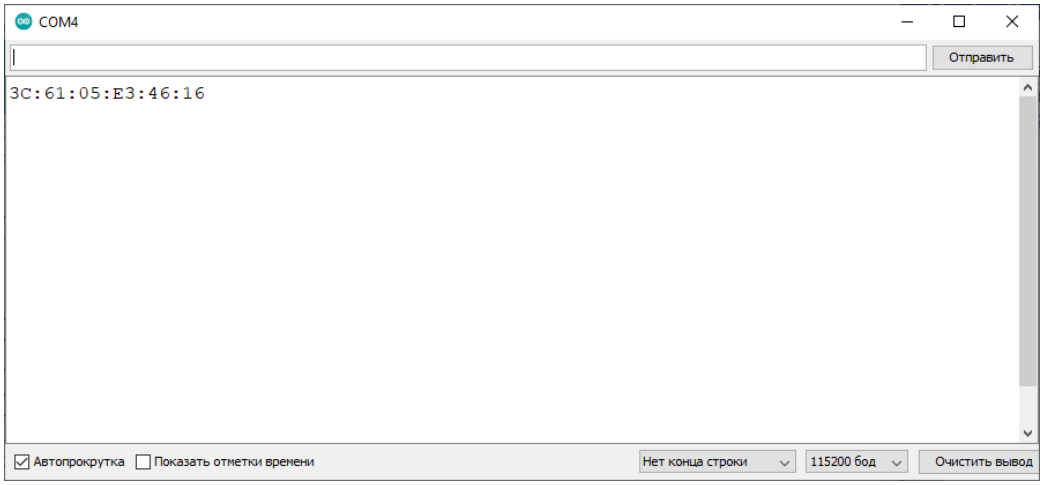


Рисунок 58 – Информация, выводимая в COM-порт

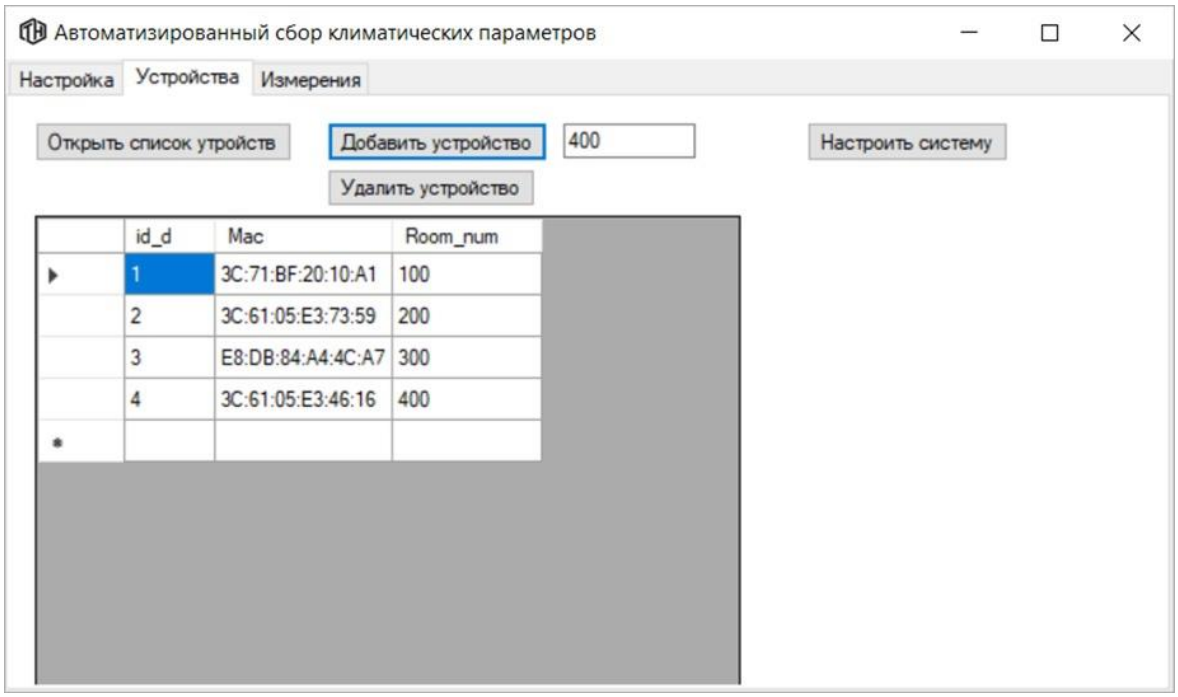


Рисунок 59 – Таблица MAC-адресов на ПК

4.2 Реализация работы системы

Для реализации работы была собрана система из трёх устройств. Собранная система продемонстрирована на рисунке 60.

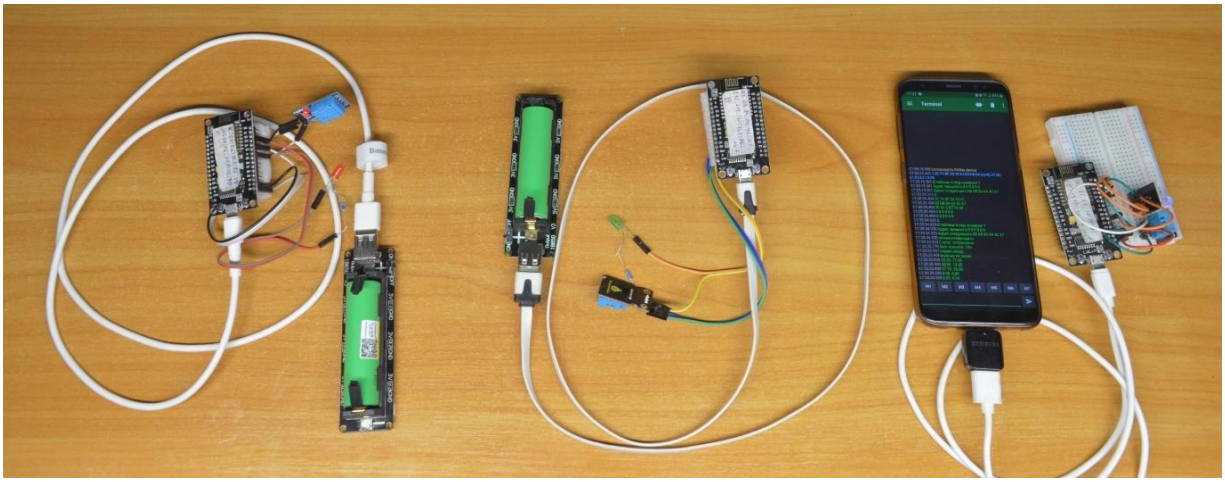


Рисунок 60 – Собранная ось радиосвязи в работе

После каждого запуска головного устройства необходимо загрузить в него таблицу MAC-адресов. Для этого с ПК необходимо отправить пакет, содержащий код операции «1» и MAC-адреса в той последовательности, в которой физически будут расставлены модули. Адреса необходимо разделять запятыми, а окончание пакета обозначать символом точка с запятой. Пример записи таблицы MAC-адресов в МК представлен на рисунке 61.

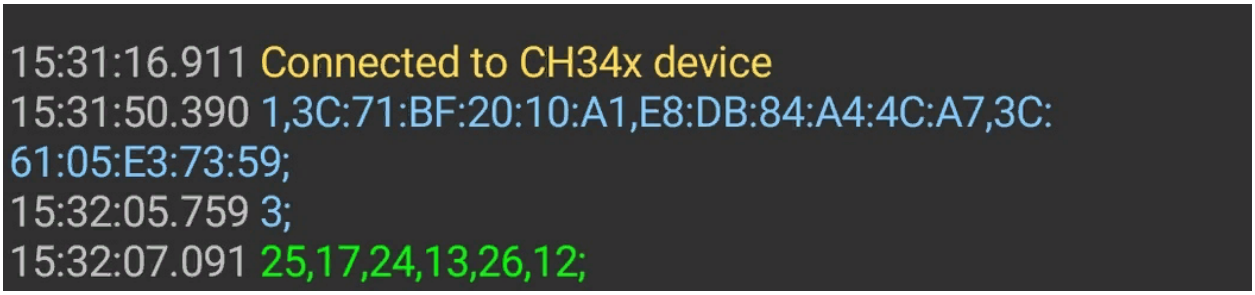


Рисунок 61 – Информация, передаваемая в COM-порте во время работы оси радиосвязи

Начало работы оси радиосвязи провоцируется командой «3;», при которой МК проводит:

– анализ таблицы MAC-адресов, нахождение своего места в таблице, нахождение, конвертация и запись в переменные MAC-адресов соседних модулей;

– обнуление массива климатических данных, отправка структуры PRD на следующий модуль;

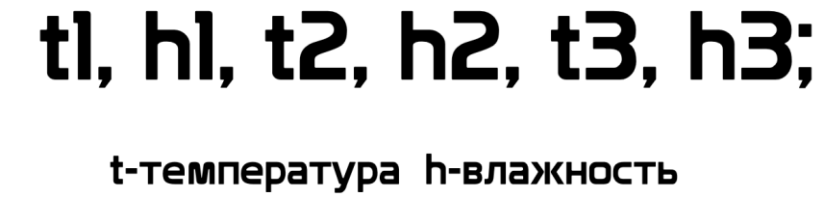
– ожидание ответа от следующего модуля;

– приём климатических данных от последующего модуля;

– добавление своих климатически данных в структуру PRD;

– вывод в COM-порт климатических данных.

Пакет, формируемый для вывода климатических данных в COM-порт, содержит температуру и влажность с каждого устройства в том порядке, в котором устройства расположены в таблице MAC-адресов. Вывод в COM-порт климатических данных изображен на рисунке 61, а структура пакета на рисунке 62.



t-температура h-влажность

Рисунок 62 – Структура пакета климатических данных

Так же в устройствах предусмотрена световая индикация работы с помощью RGB-светодиода.

– Тусклое синее свечение – ожидание команд от ПК.
– Красное свечение – прошивка МК.
– Синее мигание – МК ожидает ответа от следующего модуля.
– Синий импульс – успешная отправка по Wi-Fi.
– Зеленый импульс – по Wi-Fi принят пакет.
– Красный импульс – ошибка отправки по Wi-Fi.

4.3 Итог работы системы

По окончании одного рабочего цикла на ПК были отправлены климатические данные, которые группируются по принадлежности к конкретному устройству и записываются в базу данных. Климатические данные, добавленные в базу данных, изображены на рисунке 63.

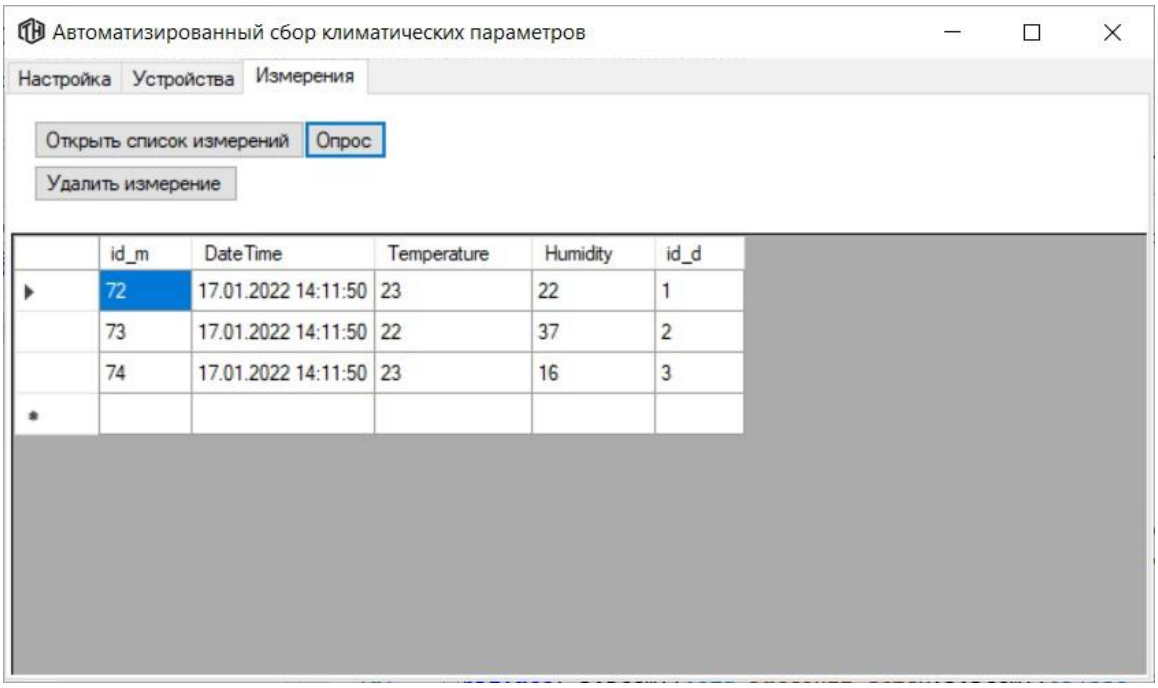


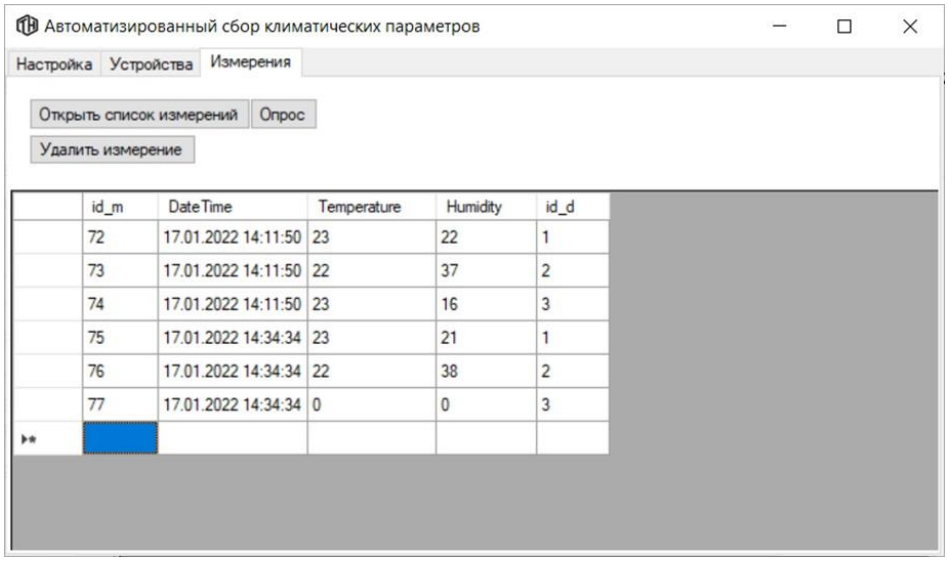
Рисунок 63 – База данных климатических данных

4.4 Исследование работы системы при выходе из строя модуля

Все устройства системы, кроме «головного» оснащены АКБ, которая имеет свойство разряжаться. Случаи, при которых один из модулей выходи из строя, предусмотрены в коде программы для МК.

В случае если выходит из строя последний модуль, предпоследний модуль сталкивается с ошибкой отправки пакета данных по Wi-Fi, что означает, что данный модуль с этого момента условно считается последним в соответствии с чем МК переходит в часть кода, соответствующую опросу датчика и запуску работы оси радиосвязи в обратном направлении. Ошибку данной природы можно распознать с помощью цветовой индикации в виде импульса красного цвета. На ПК будет отправлен пакет, содержащий климатические данные всех модулей, однако климатические параметры с

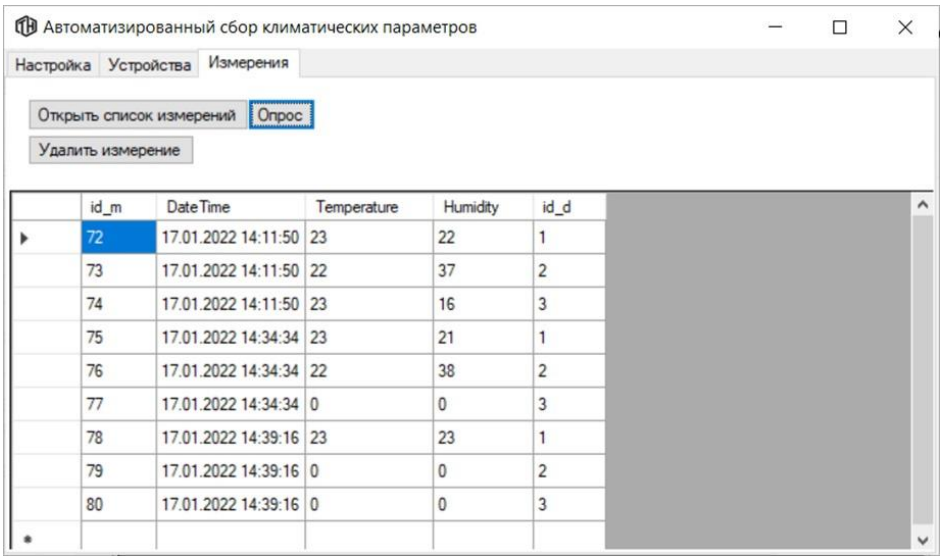
последнего модуля будут нулевыми. На рисунке 64 представлены климатические данные, полученные при неисправном последнем модуле.



id_m	DateTime	Temperature	Humidity	id_d
72	17.01.2022 14:11:50	23	22	1
73	17.01.2022 14:11:50	22	37	2
74	17.01.2022 14:11:50	23	16	3
75	17.01.2022 14:34:34	23	21	1
76	17.01.2022 14:34:34	22	38	2
77	17.01.2022 14:34:34	0	0	3

Рисунок 64 - Климатические данные, полученные при неисправном последнем модуле

Если же из стоя выходит модуль, находящийся внутри оси радиосвязи, то связь будет потеряна и со всеми последующими модулями. Итог работы оси радиосвязи при выключенном среднем модуле продемонстрирован на рисунке 65.



id_m	DateTime	Temperature	Humidity	id_d
72	17.01.2022 14:11:50	23	22	1
73	17.01.2022 14:11:50	22	37	2
74	17.01.2022 14:11:50	23	16	3
75	17.01.2022 14:34:34	23	21	1
76	17.01.2022 14:34:34	22	38	2
77	17.01.2022 14:34:34	0	0	3
78	17.01.2022 14:39:16	23	23	1
79	17.01.2022 14:39:16	0	0	2
80	17.01.2022 14:39:16	0	0	3

Рисунок 65 – Климатические данные, полученные при неисправном модуле внутри оси радиосвязи

4.5 Исследование предельной дальности связи

Для исследования предельной дальности связи систему из трёх устройств разметили в учебном корпусе. Расположение модулей проиллюстрировано на рисунке 66.

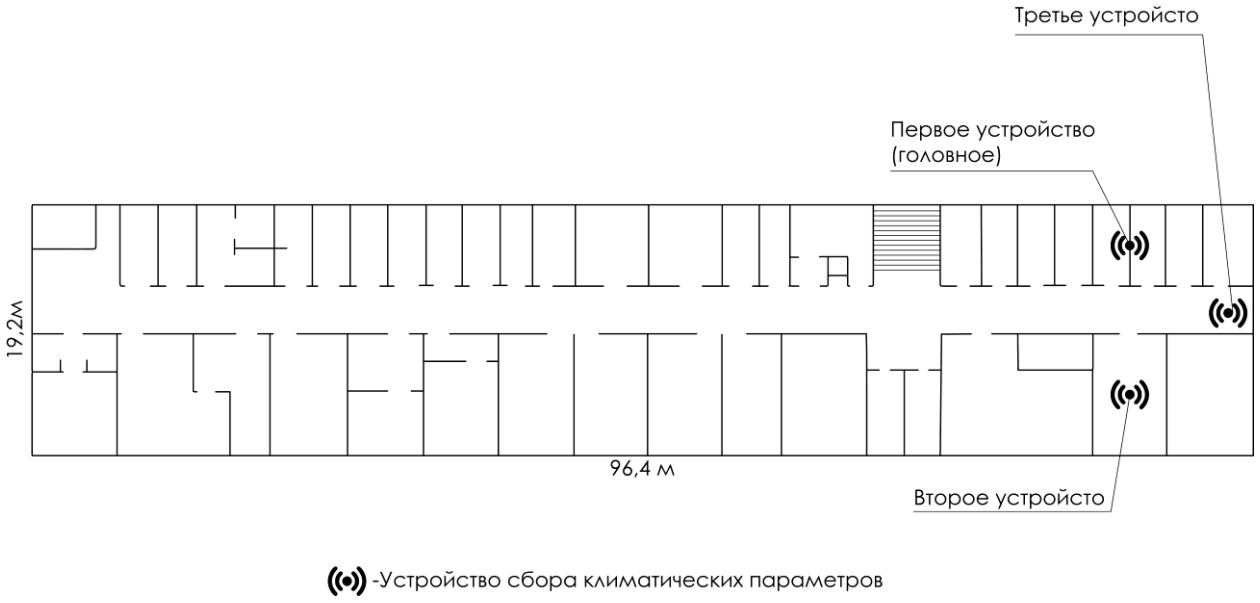


Рисунок 66 – Схема расположения устройств

В процессе исследования третье устройство перемещалось вдоль коридора, что проиллюстрировано на рисунке 67.

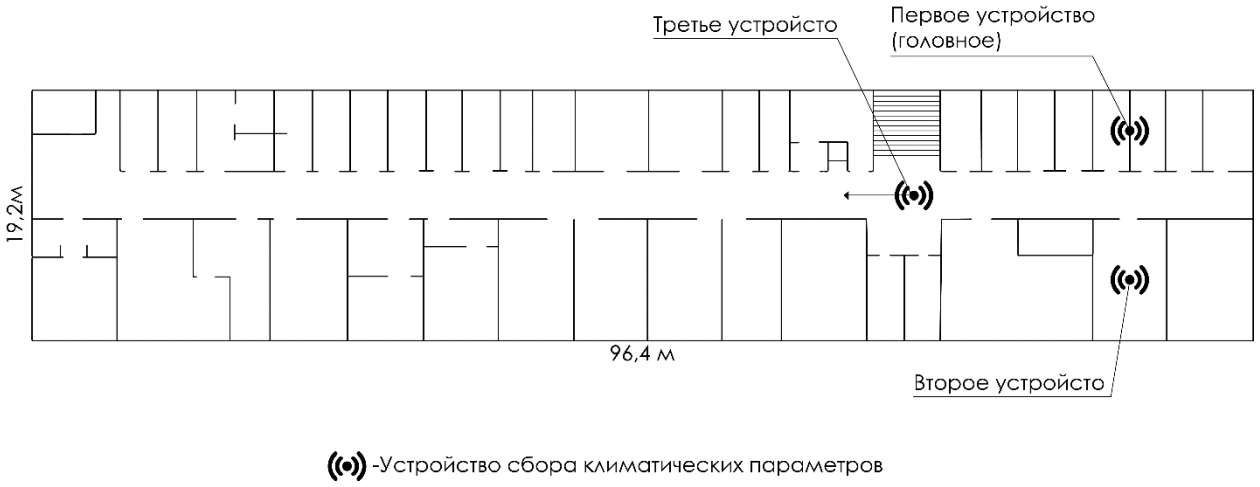


Рисунок 67 – Схема расположение и перемещения устройств

В ходе исследования установлено, что связь теряется, когда третье устройство находится в пространстве между аудиториями 200 и 230. Расположение модулей с обеспечением гарантированной связи изображено на

рисунке 68. Гарантированная дальность связи в условиях учебного корпуса порядка 50 м.

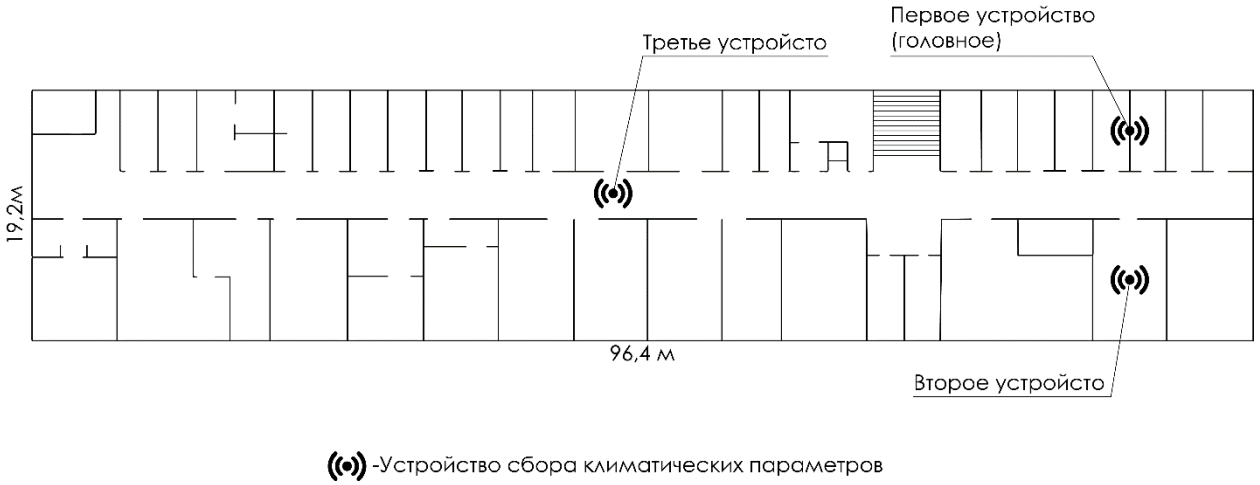
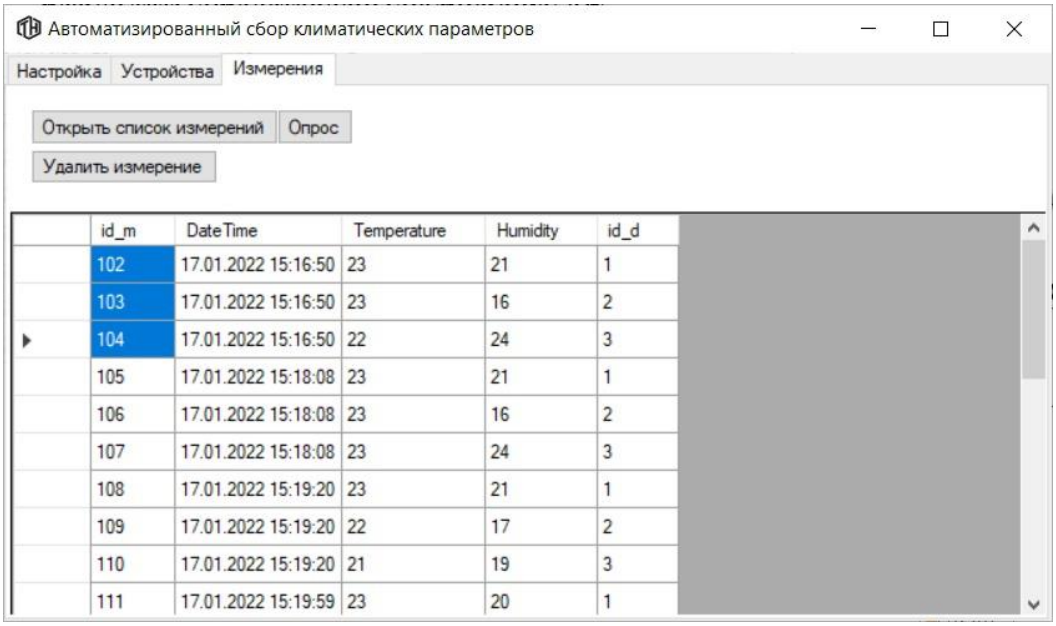


Рисунок 68 – Расположение модулей с обеспечением гарантированной связи

Данные, полученные во время исследования дальности, представлены на рисунке 69.



id_m	DateTime	Temperature	Humidity	id_d
102	17.01.2022 15:16:50	23	21	1
103	17.01.2022 15:16:50	23	16	2
104	17.01.2022 15:16:50	22	24	3
105	17.01.2022 15:18:08	23	21	1
106	17.01.2022 15:18:08	23	16	2
107	17.01.2022 15:18:08	23	24	3
108	17.01.2022 15:19:20	23	21	1
109	17.01.2022 15:19:20	22	17	2
110	17.01.2022 15:19:20	21	19	3
111	17.01.2022 15:19:59	23	20	1

Рисунок 69 – климатические данные, полученные в ходе эксперимента

Данные с номерами 102-104 соответствуют расположению модулей из рисунка 66, а данные с номерами 108-110 – рисунку 68.

4.6 Исследование работы системы при изменении микроклимата

Для исследования работоспособности системы модули были разнесены в места с различным микроклиматом. Первый модуль остался в аудитории, второй был помещен вблизи с горячей водой, а третий – вблизи открытого окна. Результаты измерений представлены на рисунке 70.

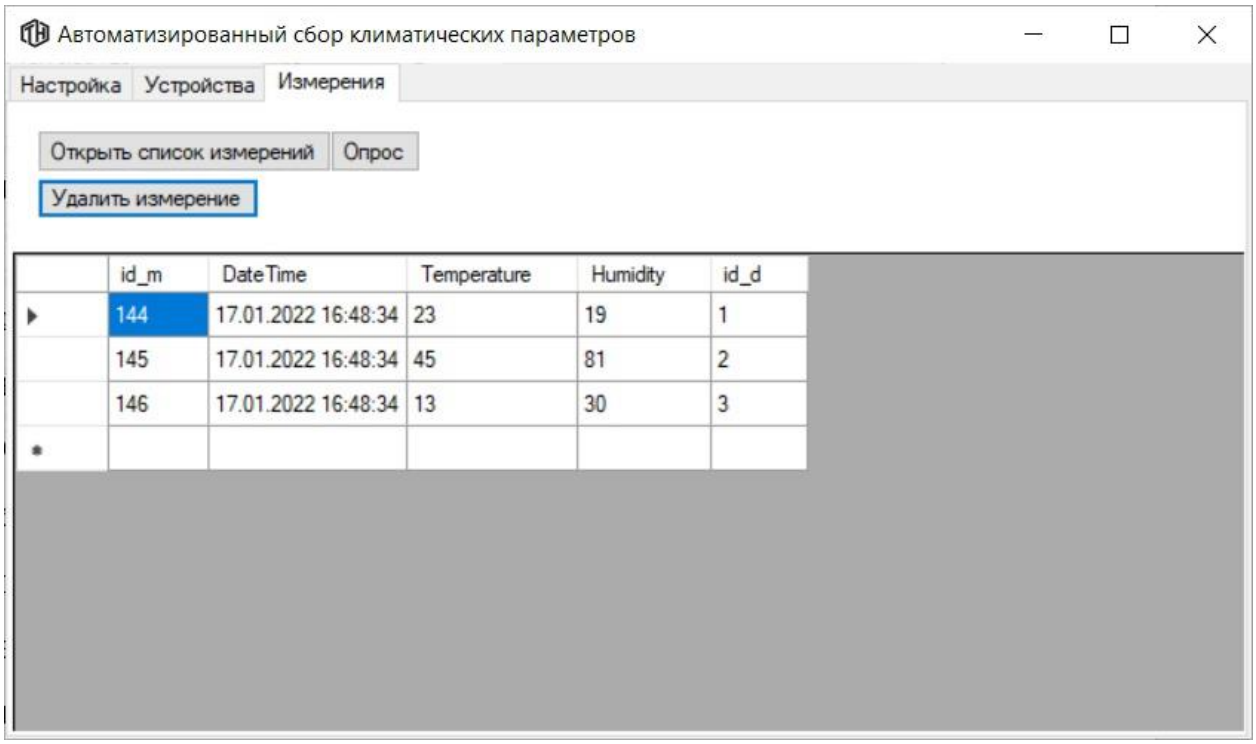


Рисунок 70 – Результаты измерений

Из базы данных видно, что параметры микроклимата второго и третьего модулей значительно отличаются от эталонных показаний первого модуля, находившегося в комнатных условиях.

ЗАКЛЮЧЕНИЕ

В ходе выполнения ВКР были получены следующие результаты:

1. Изучены и реализованы алгоритмы опроса датчика температуры и влажности DHT11.
2. Освоены среды проектирования и моделирования Arduino IDE, Processing, Autodesk Fusion 360.
3. Освоены и реализованы алгоритмы передачи и структурирования данных, передаваемых через COM-порт. Реализована работа с программным обеспечением на ПК.
4. Освоены и реализованы алгоритмы обработки MAC-адресов устройств:
   а) запись MAC-адресов в базу данных на ПК;
   б) запись таблицы MAC-адресов в память микроконтроллера;
   в) определение микроконтроллером своего места в системе и MAC-адресов соседних модулей;
   г) передача таблицы MAC-адресов на следующий микроконтроллер.
5. Изучены и реализованы алгоритмы беспроводной передачи данных между ESP8266 с помощью протокола ESP-NOW. Реализованы callback функции и индикация событий беспроводной передачи данных с помощью RGB-светодиода.
6. Разработаны и реализованы принципы установления связи системы микроконтроллеров.
7. Разработан и реализован прототип корпуса устройства.
8. Произведен расчет стоимости и энергопотребления устройства.
9. Реализована работа системы из трех устройств.
10. Произведено исследование работы
    а) при изменении климатических данных;
    б) при изменении расстояний между модулями;
    в) при выходе из строя устройств системы.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Датчики температуры и влажности DHT11 и DHT22 // MicroPi – Программирование микроконтроллеров, Banana Pi, Orange Pi, Rasperry Pi. – 2021. – URL: https://micro-pi.ru/dht11-и-dht22-датчики-температуры-и-влажности/, Режим доступа: свободный (дата обращения 21.06.2021).

2. ESP8266 // Википедия — свободная энциклопедия. – 2021. –URL: https://ru.wikipedia.org/wiki/ESP8266,
Режим доступа: свободный (дата обращения 21.06.2021).

3. Знакомство с недорогим и функциональным микроконтроллером ESP8266: прошивка и пример использования. // Tproger – сайт о программировании для программистов. – 2021. – URL: https://tproger.ru/articles/about-esp8266/,
Режим доступа: свободный (дата обращения 21.06.2021).

4. Документация по Visual Studio // Microsoft Docs. – 2021. – URL: https://docs.microsoft.com/ru-ru/visualstudio/windows/?view=vs-2019,
Режим доступа: свободный (дата обращения 21.06.2021).

5. Microsoft Visual Studio // Википедия — свободная энциклопедия. – 2021. –URL: https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio#Visual_Studio_Community
,
Режим доступа: свободный (дата обращения 21.06.2021).

6. SerialPort Класс // Microsoft Docs: Средства разработчика, техническая документация и учебные ресурсы. – 2021. – URL: https://docs.microsoft.com/ru-ru/dotnet/api/system.io.ports.serialport?view=dotnet-plat-ext-5.0,
Режим доступа: свободный (дата обращения 21.06.2021).

7. Двусторонняя передача данных между ESP8266 – 2021 URL: https://voltiq.ru/esp-now-two-way-communication-esp8266-nodemcu/,
Режим доступа: свободный (дата обращения 01.10.2021).

8. ESP-NOW на ESP8266: отправка данных по схеме «one-master-multi-slave» – 2021. URL: https://voltiq.ru/esp-now-one-to-many-esp8266-nodemcu/,
Режим доступа: свободный (дата обращения 01.10.2021).

9. Онлайн редактор блок-схем – 2021
URL: https://programforyou.ru/block-diagram-redactor,
Режим доступа: свободный (дата обращения 01.10.2021).

10. Управляем Ардуиной с компьютера через Serial. Gui на Processing – 2021
URL: https://www.youtube.com/watch?v=IfWxl5LhJE8&t=343s,
Режим доступа: свободный (дата обращения 01.10.2021).

11. Уроки Arduino. Общение по Serial, парсинг данных, протоколы связи – 2021
URL: https://www.youtube.com/watch?v=Ul03Vkg9A40&t=698s,
Режим доступа: свободный (дата обращения 01.10.2021).

12. Уроки Ардуино. Работа с текстом, String и char[] – 2021
URL: https://www.youtube.com/watch?v=1VgePUaF7R8,
Режим доступа: свободный (дата обращения 01.10.2021).

13. Делаем программу с интерфейсом на Processing – 2021
URL: https://www.youtube.com/watch?v=uidRiK1IjHo&t=17s,
Режим доступа: свободный (дата обращения 01.10.2021).

14. Большой урок по программированию на Processing – 2021
URL: https://www.youtube.com/watch?v=2fs1tuUUJRM&t=1411s,
Режим доступа: свободный (дата обращения 01.10.2021).

15. ЧИП и ДИП - розничная и мелкооптовая продажа измерительных приборов, электронных компонентов, паяльного оборудования, инструментов, средств разработки и отладки, домашней техники и электроники, сопутствующих товаров. – 2021
URL: https://www.chipdip.ru/,
Режим доступа: свободный (дата обращения 01.10.2021).

16. Амперка / Всё для Arduino и Raspberry Pi. – 2021 URL: https://amperka.ru/, Режим доступа: свободный (дата обращения 01.10.2021).

17. Интернет-магазин OZON.ru. Быстрая доставка по России – 2021 URL: https://www.ozon.ru/, Режим доступа: свободный (дата обращения 01.10.2021).

18. AliExpress | Официальный сайт – 2021 URL: https://aliexpress.ru/, Режим доступа: свободный (дата обращения 01.10.2021).

19. ООО "ЭЛГРАД Про" Производственные поставки – 2021 URL: http://elgrad.pro/, Режим доступа: свободный (дата обращения 01.10.2021).

АНТИПЛАГИАТ
ОБНАРУЖЕНИЕ ЗАИМСТВОВАНИЙ

Отчет о проверке на заимствования №1

Автор: Богославский Артём
Проверяющий: Богославский Артём (bogoslavskiy.tsu@yandex.ru / ID: 9605517)

Отчет предоставлен сервисом «Антиплагиат» - users.antiplagiat.ru

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 1
Начало загрузки: 19.01.2022 13:16:32
Длительность загрузки: 00:00:02
Имя исходного файла: BKP.pdf
Название документа: BKP
Размер текста: 66 кБ
Символов в тексте: 67232
Слов в тексте: 8276
Число предложений: 401

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Начало проверки: 19.01.2022 13:16:35
Длительность проверки: 00:00:03
Комментарии: не указано
Модули поиска: Интернет Free

ЗАИМСТВОВАНИЯ	САМОЦИТИРОВАНИЯ	ЦИТИРОВАНИЯ	ОРИГИНАЛЬНОСТЬ
2,97%	0%	0%	97,03%

Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.
Самоцитирования — доля фрагментов текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа, по отношению к общему объему документа.
Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общеупотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.
Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.
Заимствования, самоцитирования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.
Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Источник	Актуален на	Модуль поиска
[01]	1,93%	Программа графического отображения данных от мультиметра http://library.eltech.ru	19 Сен 2019	Интернет Free
[02]	0,34%	62_144_82_0_0.600_76751459 4184.07.01;РУ.01;1.doc http://sga-help.ru	06 Дек 2020	Интернет Free
[03]	0,56%	Microsoft Visual Studio http://ru.wikipedia.org	20 Авг 2020	Интернет Free

Еще источников: 7
Еще заимствований: 0,14%