

Институт машиностроения
(наименование института полностью)

Кафедра Промышленная электроника
(наименование)

ЗАДАНИЕ **на выполнение бакалаврской работы**

Студент Ивановский Илья Федорович, Элб-1601а

1. Тема Диспетчерский пульт нагревателя

2. Срок сдачи студентом законченной выпускной квалификационной работы _____

3. Исходные данные к выпускной квалификационной работе:

Связь с компьютером по сети Wi-Fi

Наличие графического интерфейса: да

4. Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов, разделов)

Аннотация

Введение

1. Обзор рынка и актуальность

2. Разработка структурной схемы

3. Выбор способа отрисовки

4. Разработка принципиальной схемы

5. Написание программы

6. Описание алгоритма устройства

7. Интерфейс пользователя

8. Экономический расчет

Заключение

Список используемой литературы и используемых источников

5. Ориентировочный перечень графического и иллюстративного материала

1. Блок-схема алгоритма

2. Схема структурная устройства

3. Схема электрическая принципиальная устройства

4. Результат работы устройства

5. Схема подключения чипа NodeMCu v3

6. Схема подключения модуля ADS1015

6. Консультант по английскому языку Усатова И.Ю.

7. Дата выдачи задания «26» января 2020 г.

Руководитель выпускной
квалификационной работы

_____ (подпись)

Е.С. Глибин
(И.О. Фамилия)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт машиностроения
(наименование института полностью)

Кафедра Промышленная электроника
(наименование)

КАЛЕНДАРНЫЙ ПЛАН
выполнения бакалаврской работы

Студент И.Ф. Ивановский
по теме Диспетчерский пульт нагревателя

Наименование раздела (главы) ¹	Плановый срок выполнения раздела (главы) ¹	Фактический срок выполнения раздела (главы) ¹	Отметка о выполнении раздела (главы) ¹
Состояние вопроса	10.03.2020	10.03.2020	Выполнено
Обзор современных решений	17.03.2020	17.03.2020	Выполнено
Аппаратная часть	28.03.2020	28.03.2020	Выполнено
Программная часть	08.04.2020	08.04.2020	Выполнено
Выбор остальных элементов устройства	12.04.2020	12.04.2020	Выполнено
Выбор способа рисования графика	25.04.2020	25.04.2020	Выполнено
Разработка ПО	10.05.2020	10.05.2020	Выполнено
Заключение	17.05.2020	17.05.2020	Выполнено
Список литературы	24.05.2020	24.05.2020	Выполнено

Руководитель бакалаврской
работы

(подпись)

Е.С. Глибин
(И.О. Фамилия)

Аннотация

Название дипломной работы: «Диспетчерский пульт нагревателя».

Выпускная работа состоит из введения, восьми глав, заключения, двух таблиц, 40 рисунков, списка литературы, включая зарубежные источники, и графической части на 6 листах формата А1.

Ключевым вопросом дипломной работы является измерение температуры для обеспечения условий производства продукции в промышленных сферах.

Целью работы является разработка устройства для обеспечения постоянного слежения за температурными изменениями и отображением графика для легкого понимания со стороны пользователя.

Дипломная работа может быть разделена на следующие логически взаимосвязанные части: анализ существующих устройств с подобным функционалом; выбор подходящего оборудования и процесс разработки структурной схемы устройства; анализ способов рисования графиков; разработка принципиальной схемы устройства; написание кода программы для каждого элемента компонентной базы; описание алгоритма работы устройства; экономический расчет.

В конце разработки мы представляем успешный прототип устройства, которое способно выполнять все ранее сказанные функции.

Подводя итоги, мы бы хотели подчеркнуть, что данное устройство актуально не только в промышленной и сельскохозяйственной сферах, но и в медицине и домашнем использовании.

Abstract

The title of the graduation work is «Control room heater».

The senior paper consists of an introduction, eight parts, a conclusion, two tables, 40 figures, list of references including foreign sources, and a graphic part on 6 A1 sheets.

The key issue of the thesis is the measurement of temperature to ensure the conditions of production in industrial fields.

The aim of the work is to develop a device to ensure constant monitoring of temperature changes and graph display for easy understanding by the user.

The graduation work may be divided into several logically connected parts which are: analysis of existing devices with similar functionality; the selection of suitable equipment and the process of developing a block diagram of the device; analysis of graph drawing methods; development of a schematic diagram of a device; writing program code for each element of the component base; description of the algorithm of the device; economic calculation.

Finally, we present the successful prototype of the device that is capable of performing all the previously said functions.

In conclusion we would like to emphasize that this device is relevant not only in industrial and agricultural fields, but also in medicine and home use.

Содержание

ВВЕДЕНИЕ	5
1. Обзор рынка и актуальность	8
2. Разработка структурной схемы	19
3. Выбор способа отрисовки	27
4. Разработка принципиальной схемы	31
5. Написание программы	36
5.1. Работа с Google Charts	36
5.2. Работа с ADS1015	40
5.3. Работа с DHT11	40
5.4. Работа с TMP35	41
5.5. Работа с термистором	41
6. Описание алгоритма устройства	43
7. Интерфейс пользователя	44
8. Экономический расчет	45
Заключение	46
Список используемой литературы и используемых источников	47
Приложение А Текст программы	50

Введение

Сегодня контролированию факторов, которые могут помешать процессам создания и выпуска качественной продукции отдается большое значение. Современные приборы в полной мере способны следить за соблюдением условий производства. В сферу их контроля входит множество параметров, в том числе и температура. Приборами для измерения температуры обеспечивают технологическое оборудование и системы нефтегазового сектора, металлургических и машиностроительных предприятий, химических и нефтеперерабатывающих заводов, производственных объектов пищевой, фармацевтической и энергетической отрасли.

Жидкостные и газовые термометры, представленные на рисунке 1, преимущественно используются для визуального контроля температуры нагревательного и холодильного оборудования.



Рисунок 1 – Газовый термометр

Термоэлектрические датчики (термопары), показанные на рисунке 2, широко применяются для автоматизации производственных процессов.



Рисунок 2 – Термопара

Пирометры, изображенные на рисунки 3, устанавливаются на теплоэнергетических объектах, в сфере пожарной безопасности и охранной сигнализации.



Рисунок 3 – Пирометр

Биметаллические датчики, представленные на рисунке 4, используются в автомобильной промышленности, отопительных и нагревательных системах.



Рисунок 4 – Биметаллический датчик

Кремниевые датчики, показанные на рисунке 5, присутствуют в различных электронных устройствах и оборудовании, и выполняют функцию контроля температуры их внутренних поверхностей.



Рисунок 5 – Кремниевый датчик температуры

Термоиндикаторы, изображенные на рисунке 6, устанавливают в холодильных установках для отслеживания превышений допустимых температур, а также в качестве одноразовых температурных датчиков.



Рисунок 6 – Термоиндикатор

В сфере цифровой медицины на крупнейшей в мире выставке MEDICA 2019 был представлен миниатюрный беспроводной термометр Thermosafer XST200, показанный на рисунке 7, который позволяет в непрерывном режиме осуществлять мониторинг температуры тела человека, в первую очередь ребенка, в течение длительного времени.



Рисунок 7 - термометр Thermosafer XST200

Неожиданное применение датчиков для слежения за ростом температуры нашлось в офисной сфере, а именно в серверной комнате. Слежение происходит за степенью нагрева серверов (рисунок 8), и при достижении пороговой температуры шлет уведомление по СМС.



Рисунок 8 – Серверная комната с установленными в ней устройствами для слежения за температурой

1. Обзор рынка и актуальность.

Наибольшую актуальность мониторинг темпа роста температуры получил в производственной сфере. Современные приборы способны в достаточной мере передавать точные данные, благодаря чему люди могут полноценно управлять производственными процессами, корректно выполнять требования рабочих режимов и в конечном итоге обеспечивать выпуск качественной промышленной продукции.

Компания Ashcroft Inc производит преобразователи/передатчики, переключатели давления и температуры, термометры, манометры, контрольно-измерительную аппаратуру. Область применения их аппаратуры распространяется практически на все виды деятельности: сточные воды очистительных сооружений, биотехнологические и фармацевтические лаборатории, медицинские аппараты, полупроводниковые установки, нефтеперерабатывающие заводы, электростанции, предприятия пищевой промышленности, целлюлозно-бумажные комбинаты, заводы химического производства.

В частности Ashcroft Inc предлагает три типа термометров: биметаллические, с газопроводом Duratemp и цифровые.

Биметаллические термометры (рисунок 9) осуществляют замеры в пределах от -80°F до 1000°F . Принцип действия основан на упругой деформации, возникающей под воздействием температуры двух прочно соединенных металлических пластин, имеющих различные температурные коэффициенты линейного расширения, изгиб с помощью кинематического узла преобразуется во вращательное движение стрелки, показывающей измеряемое значение температуры по шкале термометра. Термометр состоит из круглого корпуса, в котором размещены циферблат и кинематический механизм со стрелкой, и биметаллического термочувствительного элемента в

защитной трубке (термобаллона). Корпус и термобаллон изготавливаются из нержавеющей стали.



Рисунок 9 – Биметаллический термометр компании Ashcroft Inc

Термометры серии Duratemp измеряют температуру от -320°F до 1200°F . Принцип работы основан на зависимости между температурой и давлением термометрического вещества – инертного газа, находящегося в герметично замкнутой манометрической термосистеме. Под воздействием температуры на термобаллон термометра изменяется давление внутри манометрической термосистемы и под действием давления происходит раскрутка манометрической пружины, связанной со стрелкой отсчетного устройства.

Компания ООО «Теплоприбор Экспресс Анализ» разрабатывает, создает и реализует продукцию «экспресс-анализа» для всех типов расплавов металла. Продукция ориентирована на предприятия металлургической промышленности, а также других отраслей, имеющих свои производства по выплавке металлов. Их термоэлектрические преобразователи одноразового применения (рисунок 10) серий ЗК, ПТП предназначены для измерения температуры жидких металлов путем кратковременного погружения термопарных вставок, закрепленных в многослойных картонных трубках. Принцип действия преобразователей основан на генерировании термоэлектродвижущей силы, возникающей из-за разности температур между двумя соединениями различных металлов или сплавов.



Рисунок 10 - Термоэлектрические преобразователи одноразового применения

Компании Fluke создала пирометр инфракрасный Fluke 62 MAX ESPR (рисунок 11), разработанный специально для эксплуатации в жестких условиях высокой запыленности, вибраций, повышенной влажности. Он предназначен для бесконтактного измерения температуры поверхностей твердых тел по их собственному тепловому излучению, при этом размеры отображаемой поверхности объекта определяются угловым полем зрения пирометра. Принцип действия основан на преобразовании потока инфракрасного излучения исследуемого объекта, переданного через оптическую систему и инфракрасный фильтр на фотоэлектрический приемник, в электрический сигнал, пропорциональный температуре, затем сигнал преобразуется внутренней микропроцессорной системой в цифровой сигнал. Пирометр представляет собой оптико-электронное устройство, состоящее из: объектива, фокусирующего излучение объекта на термоэлектрический приемник и электронного блока измерения, регистрации и индикации. Микропроцессорная система пирометров обеспечивает обработку полученного результата измерения и индикацию на жидкокристаллическом дисплее текущего, максимального, минимального значения измеряемой температуры объекта, а также разности температур и средней температуры объекта измерений.



Рисунок 11 - Пирометр инфракрасный Fluke 62 MAX ESPR

Также компанией Fluke был выпущен термометр Fluke 52 II (рисунок 12). Предназначен для измерений и преобразования сигналов термо-ЭДС от внешних термоэлектрических преобразователей в значение температуры в соответствии с номинальной статической характеристикой преобразования (НСХ) по ГОСТ Р 8.585-2001, с последующем отображением результатов измерений на дисплее. Представляет собой переносной микропроцессорный прибор, конструктивно выполненный в виде единого блока обработки сигналов с автономным питанием, расположенного в пластиковом корпусе в защитном резиновом чехле и состоящего из микропроцессора, цифрового жидкокристаллического индикатора и панели управления.



Рисунок 12 - Термометр Fluke 52 II

Вспоминая про врачей и докторов, не стоит забывать, что у них также ведется технологическая гонка по разработке устройств для слежения за нашим здоровьем. Так корейская компания Chois Technology создала относительно простое медицинское устройство, получившее название Thermosafer XST200, которое с помощью Bluetooth (на расстоянии, не

превышающем 50 м) подключается к смартфону родителей для контроля температуры тела младенцев в режиме реального времени. Система поддерживает телефоны на базе iOS и Android. В случае, если температура младенца превышает заданный уровень, Thermosafer подает сигнал тревоги и в соответствующем приложении на смартфоне отображает графики изменения температуры (рисунок 13).



Рисунок 13 – Пример использования Thermosafer XST200

Термометр использует традиционный метод измерения температуры в подмышечной впадине, который известен всем родителям, благодаря чему может получить точную температуру. После определения нормального диапазона температур тела в зависимости от возраста ребенка пользователя, это устройство помогает своевременно отреагировать при первых признаках появления аномальной температуры и предпринять необходимые шаги. Измерения температуры в автоматическом режиме производятся каждые 3 секунды.

Устройство отличается высокой точностью ($\pm 0,1^{\circ}\text{C}$ в диапазоне температур 25°C - 65°C) и закрепляется на теле ребенка при помощи гипоаллергенного силиконового липкого патча. Размер термометра: 2,9 x 5,7 x 0,25 см, вес - около 60 г. Продолжительность непрерывной работы - 14 дней.

Разработанный компанией Chois Technology хаб CMS Gateway, показанный на рисунке 14, позволяет обслуживать сразу несколько

подобных устройств и контролировать температуру пациентов в больнице. При использовании этого устройства значения температуры пациентов могут в непрерывном режиме передаваться через облачную систему в рабочую станцию на сестринском посту или прямо в смартфон/планшет медсестры.



Рисунок 14 - Хаб CMS Gateway

Этот же хаб используется для передачи данных температуры, показанный на рисунке 15, родителям, если они, например, находятся на расстоянии от ребенка, а с ним находится приходящая няня. Для этого используется облачная система, к которой подключен хаб.



Рисунок 15 – Оповещение родителей о повышенной температуре ребенка

Для предотвращения чрезвычайных ситуаций в работе серверного оборудования компания ООО «Алентис Электроникс» разработала серию недорогих и функциональных устройств мониторинга температуры по сети Ethernet/Internet.

Устройства NetPing позволяют удалённо получать информацию о состоянии датчиков, а также мгновенно выполняют отправку e-mail, syslog, SNMP trap, SMS-уведомлений о срабатывании датчиков обслуживающему персоналу в любое время суток.

Устройства NetPing для мониторинга температуры в серверной комнате и реализации других возможностей используют различные технологии и протоколы. Самым простым способом наблюдения за показаниями температуры является использование встроенного web-интерфейса устройства. Так, например, на странице «Термодатчики» (рисунок 16) можно следить за показаниями подключенных датчиков температуры для серверной в режиме реального времени.

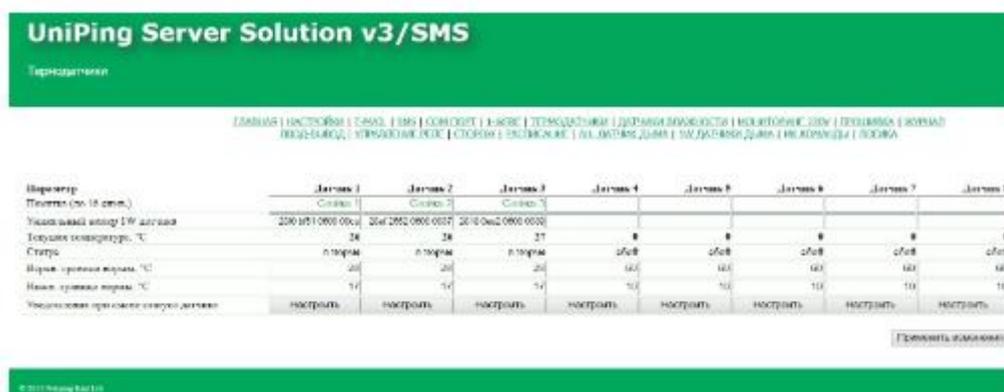


Рисунок 16 – Просмотр датчиков с помощью web-интерфейса

Устройства NetPing обладают большим потенциалом в плане интеграции в различные системы мониторинга, автоматизированные системы и программные продукты. Такую широкую сферу применения позволяет осуществлять протокол SNMP, который поддерживается любым устройством NetPing. Благодаря этому на базе устройств NetPing можно анализировать, например, графики температуры воздуха на улице в различные интервалы времени, пример которого представлен на рисунке 17.



Рисунок 17 - График, созданный на базе устройства NetPing

Говоря о конкретных проектах, устройство UniPing server solution v4/SMS — это устройство для удаленного мониторинга и слежения за микроклиматом в серверной комнате (рисунок 18). Датчики температуры и влажности позволяют контролировать условия эксплуатации оборудования в серверной комнате. Дополнительно устанавливаются датчики двери, движения, удара, которые предупредят о несанкционированном доступе в серверное помещение. А датчики дыма и протечки позволят получать своевременные уведомления о нештатных ситуациях.



Рисунок 18 – Устройство UniPing server solution v4/SMS

Согласно информации на сайте производителя в некоторых случаях UniPing server solution v4/SMS может автоматически предпринять необходимые действия для восстановления оптимальных условий работы оборудования. Например, при превышении уровня температуры включить резервный кондиционер.

Обычно UniPing server solution v4/SMS используется:

- для мониторинга физических условий работы компьютерного оборудования и ограничения доступа к ящикам с оборудованием, уведомления ответственных лиц о нештатных ситуациях (email, SMS-уведомления, локальные уведомления);
- удаленного управления системами вентиляции и поддержания микроклимата, кондиционерами. Подключенные датчики позволяют отслеживать текущую ситуацию, а удаленное управление розетками 220 В при помощи розеток NetPing AC/DIN позволяет включить необходимую систему без физического присутствия на объекте (рисунок 19).



Рисунок 19 – Пример возможностей UniPing server solution v4/SMS

Для предотвращения чрезвычайных ситуаций при хранении и перевозке медицинских препаратов, продуктов питания, специальных грузов компания Log Tag разработала электронные термоиндикаторы для контроля соблюдения температурного режима.

Для разных целей компания создала множество модификаций, так, например, термоиндикатор Log Tag TRIX-8 (рисунок 20) предназначен для контроля температуры в диапазоне от -40°C до $+85^{\circ}\text{C}$. Позволяет сохранять в памяти до 8000 значений. Главной особенностью представляется гибкая настройка параметров работы: время начала работы, интервал считывания данных, циклическая или фиксированная запись данных, настройка условий активации индикатора ALERT/ТРЕВОГА.



Рисунок 20 - Термоиндикатор Log Tag TRIX-8

С помощью программного обеспечения LogTag Analyzer можно запрограммировать устройство, задав:

- границы температурного режима
- частоту измерений
- количество измерений

После окончания мониторинга температуры прибор можно снова подключить к программе для получения временного графика, файла отчета или сводной таблицы, пример которых изображен на рисунке 21.

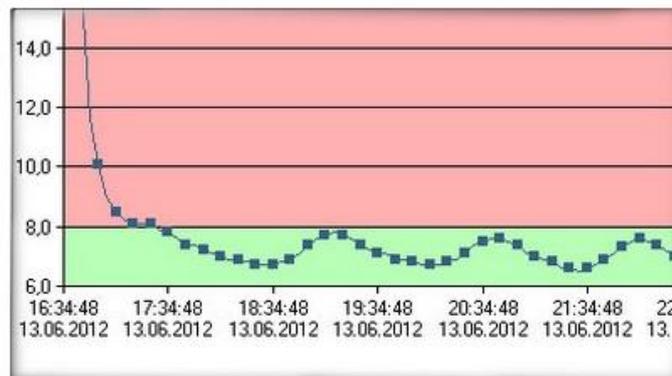


Рисунок 21 – временный график работы прибора

Для подключения термоиндикатора к компьютеру было разработано устройство USB-интерфейс (рисунок 22). После вставки прибора в гнездо, посредством USB-соединения, данные термоиндикатора сохраняются в памяти компьютера в виде специального файла.



Рисунок 22 - USB-интерфейс для подключения к компьютеру

Так как большинство информации о том, как устроены приборы и по каким алгоритмам они функционируют, является коммерческой тайной производителей, то узнать подробнее об их работе нам не представляется возможным.

То, что объединяет все представленные выше устройства, является цена. Для мониторинга температуры в реальном времени и передачи этой информации пользователю требуется создание ЭВМ с хорошими вычислительными возможностями.

Но современные технологии развились настолько, что стало возможным создание похожего изобретения за гораздо меньшие деньги. В данной работе предлагается реализация одного из таких проектов.

Вывод по разделу

Таким образом, мы увидели, что функционал устройства, которое получится у нас в результате данной работы, обширно внедряется и эффективно применяется не только в промышленной сфере, но и в медицинской и офисной. Это подтверждает актуальность данного устройства.

2. Разработка структурной схемы

Для реализации подобной идеи нам понадобится плата, а точнее микроконтроллер желательно со встроенной функцией Wi-Fi для связи с браузером компьютера. Также это поможет лишний раз сэкономить.

Измерять температуру будем специальными датчиками, причем их будет несколько. Это позволит нам следить за изменениями в климате сразу в нескольких местах, и повысить точность измерений, если будем замерять температуру в одном месте. Будут использоваться цифровой, аналоговый датчики и термистор.

Целью проекта является слежение за температурой и вывод этой информации в удобном для пользователя виде, в данном случае в виде графика. К счастью интернет полон способов, благодаря которым это можно осуществить.

Определившись со структурой, начнем подбор отдельных элементов.

В качестве «основы» подойдут недорогие и функциональные микроконтроллеры ESP8266, представленные на рисунке 23, китайский микроконтроллер от производителя Espressif с поддержкой Wi-Fi интерфейса. Часто этот микроконтроллер позиционируют только как Wi-Fi модуль, однако в нашем случае именно к нему в дальнейшем будут подключены датчики температуры.

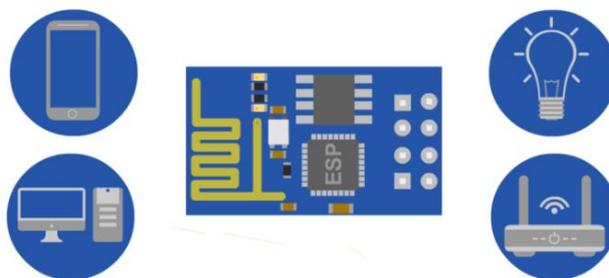


Рисунок 23 – Плата ESP

У этого микроконтроллера нет встроенной flash-памяти, поэтому он работает с внешней flash-памятью по интерфейсу SPI. В большинстве

популярных версий её объём варьируется от 512 Кбайт до 4 Мбайт. При желании и умении микросхему памяти можно перепаять на версию до 32 Мбайт.

Существует около полутора десятка версий микроконтроллера серии ESP и огромное количество плат с ними. Рассмотрим самые популярные из них (таблица 1).

Таблица 1 – Разновидности плат и микроконтроллеров ESP

Название	Фото	Описание
Микроконтроллер ESP-01		Считается самым популярным из серии. Имеет 8 разведённых контактов (VCC, GND, UTXD, URXD, CH_PD, GPIO0, GPIO2, GPIO6) и PCB-антенну (печатный проводник на самой плате). Из разведённых выводов тут присутствуют только 3 GPIO, но не стоит видеть в этом одни минусы. Если нужно будет управлять одним реле или получать данные с датчика температуры, вам не понадобятся все выводы МК, достаточно будет лишь пары.
Микроконтроллер ESP-03		Здесь появляется керамическая антенна. Она считается немного эффективней своего печатного собрата. Также на плате разведены все доступные выводы GPIO.
Микроконтроллер ESP-07		В этой версии в глаза сразу бросается металлический экран (который перед этим появляется на ESP-06). На борту керамическая антенна и разъём для внешней антенны.
Плата WeMos D1 mini		Имеет распайку девяти GPIO-контактов. На плате имеется мост CH34x. Установлен МК с 4 Мбайт flash-памяти. Конструкционно совместима с различными выпускаемыми шилдами реле/датчиками.

Продолжение таблицы 1

<p>Плата NodeMCU v0.9/v1</p>		<p>Первое поколение плат серии NodeMCU. На ней распаяны все 11 GPIO-портов. Некоторые из них обладают дополнительными функциями (UART, I2C, SPI, PWM, ADC). Хотя на плате впаяны контакты, она занимает всю ширину беспаячной макетной платы, что затрудняет работу на ней. МК имеет 4 Мбайт flash-памяти. Также имеется мост CH340.</p>
<p>Плата NodeMCU v3</p>		<p>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.</p>

Все эти (и не только эти) микроконтроллеры выполнены на чипсете ESP8266EX, следовательно, характеристики у них одинаковые:

- 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.
- Встроенные аппаратные ускорители: CCMP (CBC-MAC, режим счётчика), TKIP (MIC, RC4), WAPI (SMS4), WEP (RC4), CRC.
- Поддержка LUA-скриптов.

Все микроконтроллеры по характеристикам не сильно различаются, так что, недолго думая, возьмем плату NodeMCU v3. Так как она является относительно новейшей моделью, то внезапных проблем с ней должно быть меньше всего. Плата имеет 11 GPIO-портов и один вывод ADC0. Это вывод встроенного 10-разрядного аналого-цифрового преобразователя (АЦП). Он понадобится для расчета данных, полученных с аналогового датчика и тиристора.

Для того, чтобы одновременно подключить аналоговый датчик и термистор нам нужно два аналоговых порта, но NodeMCU v3 имеет только один, чего явно недостаточно. Эту проблему решит ADS1015 – 12-разрядный Аналого-Цифровой Преобразователь, представленный на рисунке 24. Он имеет внутренний ИОН (Источник Опорного Напряжения), 4 аналоговых входов. Напряжение питания держится в диапазоне от 2В до 5В, потребляемый ток составляет 150 мА. Также он очень боится перенапряжения, как по входам, так и по питанию. Превышение напряжения более чем на 5% от напряжения питания его моментально сожжет.



Рисунок 24 – ADS1015

Теперь разберемся с датчиками. Как говорилось ранее, было принято решение взять три их разновидности, то есть цифровой, аналоговый и термистор. Так можно будет выяснить, какой из них наиболее удобен в использовании и точен в измерении.

В качестве цифрового датчика возьмем DHT. Есть две разновидности данного датчика: DHT11 и DHT22. Они очень популярны в среде Arduino и часто используются в проектах метеостанций и умного дома. Состоят из двух частей – емкостного датчика температуры и гигрометра. Первый используется для измерения температуры, второй – для влажности воздуха (в данной работе не используется). Использовать будем DHT11, представленный на рисунке 25, так как он обладает достаточной точностью измерений (погрешность может составлять до 5%) в диапазоне от 0°C до 50°C. Питается от 3 до 5 В и стоит дешевле, чем его аналог. Для подключения потребуется резистор на 10 кОм.

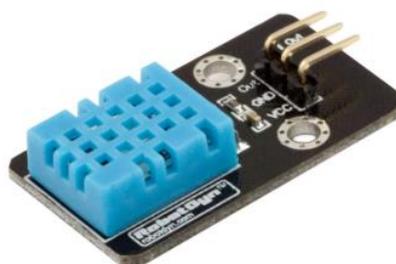


Рисунок 25 – Датчик температуры и влажности воздуха DHT11

Перейдем к аналоговому датчику. Для этой роли возьмем TMP35, изображенный на рисунке 26, использующий технологии твердотельной электроники для определения температуры. То есть, в них установлены термисторы, в которых при повышении температуры увеличивается напряжение в диоде. Это дает возможность генерировать аналоговый сигнал, пропорциональный температуре. Так как различия в характеристиках между TMP35 и TMP36 есть только в диапазоне измерений, выбор был сделан, исходя из того, что нам не потребуется замер при низких температурах.

TMP35 измеряет температуру от 10°C до 125°C и обеспечивает выходное напряжение 250 мВ при температуре 25°C.



Рисунок 26 – Датчик температуры TMP35

Все аналоговые датчики имеют важную характеристику – отношение количества вольт к единице измеряемой величины. Например, в спецификации к TMP35 написано, что каждый градус измеряемой температуры соответствует 10 мВ напряжения на выходе. Так, чтобы преобразовать аналоговый сигнал датчика в градусы, воспользуемся формулой (1):

$$T = \frac{raw}{4096} \cdot 3,3 \cdot 1000, \quad (1)$$

где *raw* – значение АЦП; 4096 – максимальное значение, которое может вернуть АЦП от ADS1015; 3,3 – рабочее напряжение АЦП.

Последним для нас остался термистор. Это резистор, который меняет свое сопротивление с изменением температуры. Существуют два вида термисторов - с NTC (negative temperature coefficient - отрицательный температурный коэффициент) и с PTC (positive temperature coefficient - положительный температурный коэффициент). В большинстве случаев для измерения температуры используются NTC сенсоры, мы поступим также и используем простейший NTC термистор с номинальным сопротивлением

100 кОм при температуре в 25°C. Данный термистор имеет маркировку 3950 (рисунок 27).



Рисунок 27 – Термистор NTC 3950

Чтобы вычислить значение температуры используют формулу Стейнхарта – Харта (2):

$$\frac{1}{T} = A + B \ln(R) + C (\ln(R))^3 \quad (2)$$

Но, так как нам не требуется большой точности, воспользуемся модифицированным уравнением (В-уравнение) (3):

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{B} \cdot \ln\left(\frac{R}{R_0}\right) \quad (3)$$

где T_0 – комнатная температура, для которой указывается номинал термистора, $T_0 = 25^\circ\text{C}$; T – искомая температура; R – измеренное сопротивление термистора; R_0 – номинальное сопротивление термистора.

Вывод по разделу

Таким образом, на основе выбранных элементов и описанной структуры составим структурную схему диспетчерского пульта нагревателя (рисунок 28).

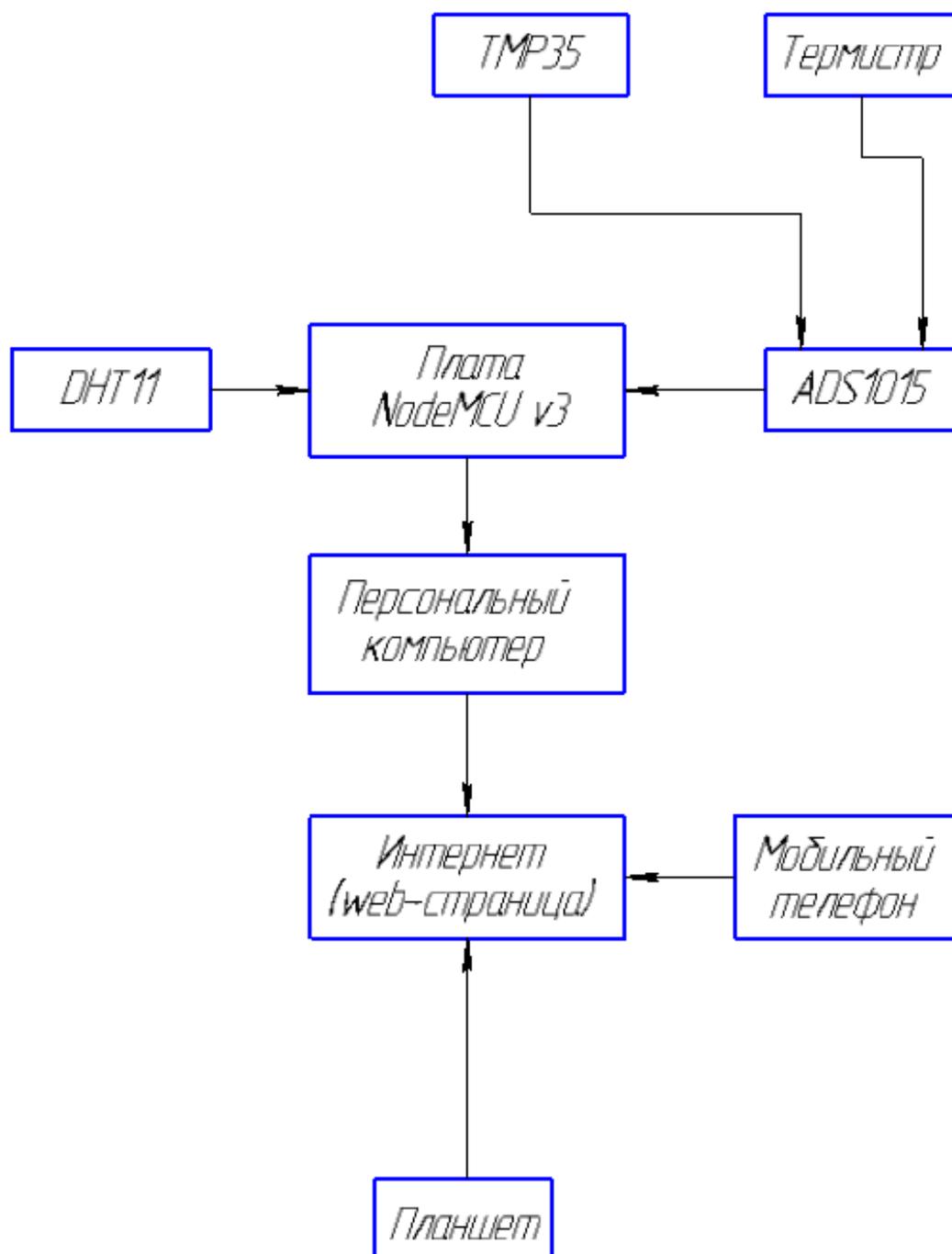


Рисунок 28 – Структурная схема устройства

3. Выбор способа отрисовки графика

Для того чтобы отображать график изменения температуры в реальном времени существует несколько возможностей, которые кажутся довольно интересными. Но, сразу забегая вперед, большинство из них не подходят для целей данного проекта ввиду своей ограниченности. Рассмотрим некоторые из них.

Первым, что мне попало на глаза, является приложение для рисования графиков SFmonitor. Оно было разработано для разработки беспилотных летательных аппаратов, контроля над их системами стабилизации. Работает с информацией, получаемой из последовательного порта компьютера. При помощи библиотеки SerialFlow в SFmonitor можно строить до трех графиков одновременно (рисунок 29). Это является одновременно плюсом и минусом, так как в том случае если мы захотим расширить структурную базу, нам придется искать другой способ отображения графиков. К тому же главным минусом является невозможность посмотреть график с другого устройства, кроме компьютера, с последовательного порта которого происходит передача информации.

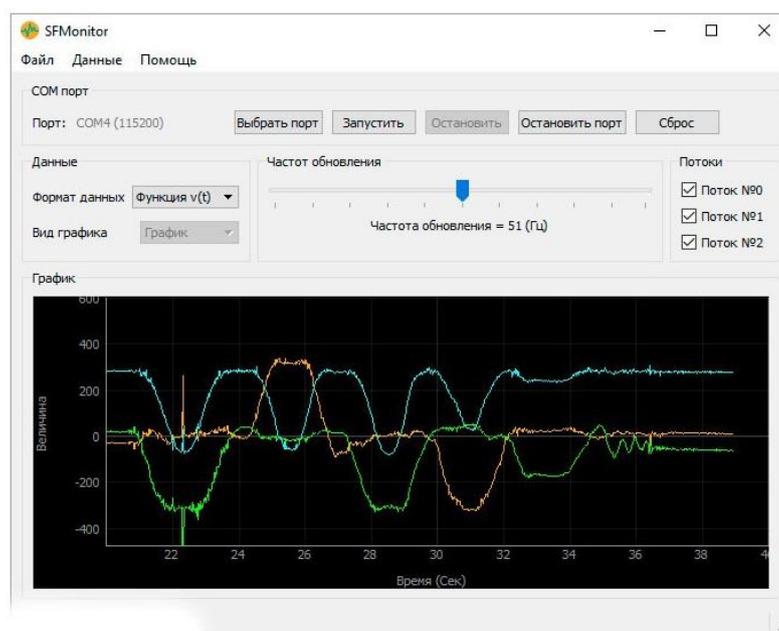


Рисунок 29 – Пример графиков, построенных в приложении SFmonitor

Также для рассмотрения можно взять встроенную в Arduino IDE утилиту «Плоттер по последовательному соединению». Про нее мало чего можно сказать, кроме того, что возможна работа только с одним графиком, и полное отсутствие настроек. Пример графика представлен на рисунке 30.



Рисунок 30 – Пример графика, построенного в Arduino IDE

Уже более подходящим для нас является визуализация данных при помощи EON. Это Фреймворк JavaScript с открытым исходным кодом для вычерчивания графиков, созданный PubNub. Считыванием числовых значений с сенсора занимается Johnny-Five. Для его работы нужно будет скачать библиотеку ConfigurableFirmata, так как для работы с устройствами на основе интерфейса 1-Wire, ему нужен этот модуль. Также нужно будет установить Node.js, который разработан на основе C3.js, который является производным от D3.js – Data-Driven Documents, библиотеки JavaScript для манипуляции с документами. Как сказано на сайте: «D3 помогает вам воплотить данные в жизнь, используя HTML, SVG, CSS». Пересылка данных с сенсора происходит с помощью модуля PubNub. На основе полученных данных выводится график, показанный на рисунке 31. Метод сам по себе весьма интересный, но очень громоздкий, а скачивание и установка дополнительного программного обеспечения может легко запутать, поэтому было принято решение не использовать этот способ.

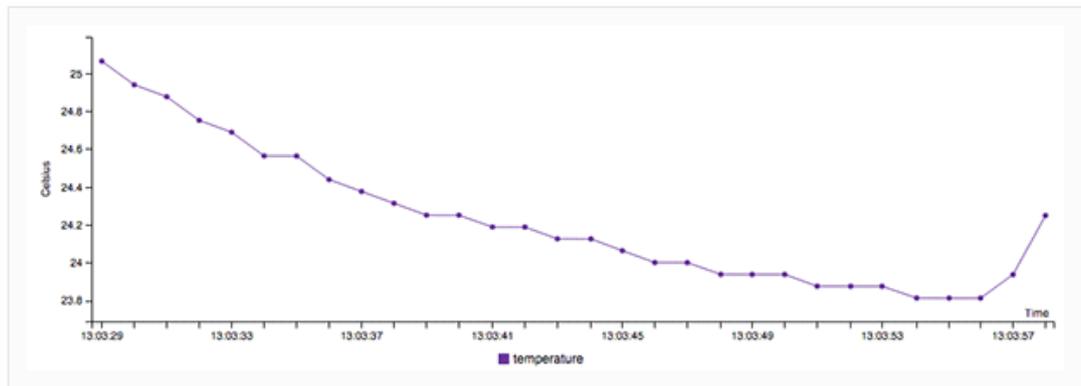


Рисунок 31 – Пример графика, построенного с помощью EON

В качестве того, что будет отображать данные с датчиков, была выбрана Google Charts - это библиотека диаграмм на основе JavaScript, предназначенная для улучшения веб-приложений за счет добавления интерактивных возможностей построения диаграмм. Она поддерживает широкий спектр графиков (рисунок 32). Графики создаются с использованием SVG в стандартных браузерах, таких как Chrome, Firefox, Safari, Internet Explorer (IE).



Рисунок 32 – Примеры диаграмм и графиков, которые доступны в Google Charts.

Для данного проекта используется только Line Chart. Преимущество, которое дает Google Charts, состоит в том, что саму библиотеку не обязательно загружать в микроконтроллер. Достаточно просто прописать обращение к Google Charts в глобальном интернете и браузер сам отобразит график, согласно переданным данным и настройкам для библиотеки на веб-странице.

Вывод по разделу

В этом разделе мы просмотрели различные способы рисования графиков, рассмотрели их минусы и плюсы, какие проблемы могут сопровождать нас при построении графика. В конце выбрали наиболее подходящий для нас способ.

4. Разработка принципиальной схемы

Схема подключения чипа внутри микроконтроллера NodeMCU v3 приведена на рисунке 33.

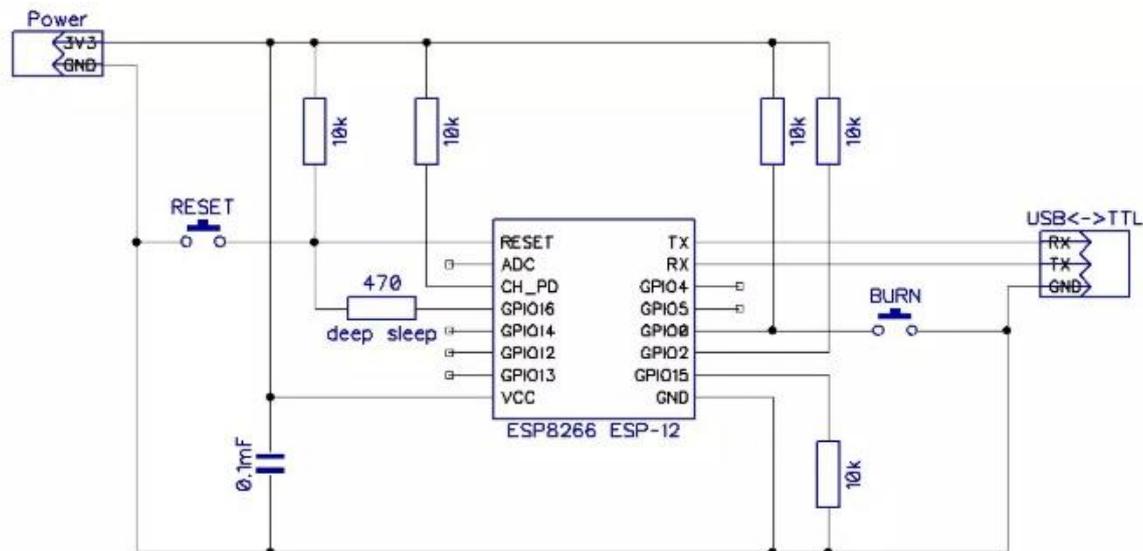


Рисунок 33 - Схема подключения чипа

Здесь Vcc и GND – это выводы непосредственного питания микроконтроллера. Вывод GPIO16 используется для пробуждения чипа ESP8266 из режима глубокого сна. RESET предназначен для сброса микроконтроллера. Выводы TX и RX нужны для асинхронного последовательного интерфейса, они устанавливают связь с другими устройствами по шине UART. Но так как мы передаем данные по Wi-Fi, то эти выводы остаются неподключенными.

При подключении модуля ADS1015 выводы VDD и GND также используются для питания самого устройства. SCL и SDA нужны для общения с ведущим устройством по протоколу I2C, подключаются к свободным GPIO-портам микроконтроллера. Дополнительно устанавливать подтягивающие резисторы на шину нет необходимости, так как они уже предусмотрены в конструкции модуля. Вывод ADDR задает один из 4-х возможных адресов модуля. Но данная функция оказывается полезна только при подключении нескольких модулей, поэтому использоваться не будет. A0,

A1, A2 и A3 – входы АЦП, к которым будут подключены датчики температуры. На рисунке 34 представлена электрическая схема модуля.

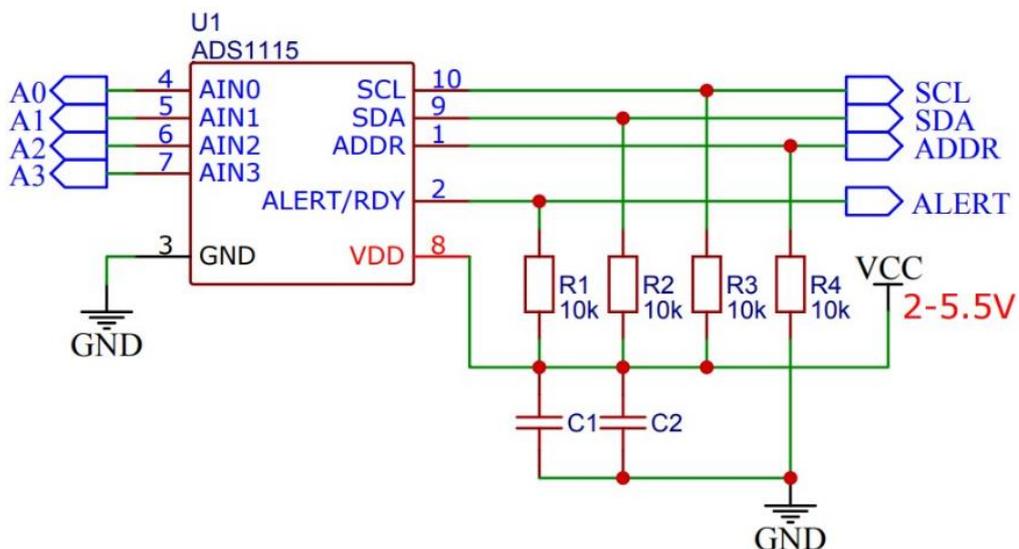


Рисунок 34 - Схема подключения модуля ADS1015

Конденсаторы C1 и C2 предназначены для устранения помех в цепи питания микросхемы АЦП. Резистор R1 подтягивает выход компаратора к низкому уровню. Следовательно, сигнал оповещения будет эквивалентен логической единице. Резисторы R2 и R3 подтягивают I2C шину до уровня питания, что освобождает от применения внешних сопротивлений.

Схема включения цифрового датчика DHT11 представлена на рисунке 35. Выводы Vcc и GND подключаются к выводам Vcc и GND микроконтроллера соответственно. Вывод Data к аналоговому пину для передачи данных. NC остается висеть в воздухе и не используется. Для правильной работы датчика между выходом сигнала и питанием впаян резистор на 10 кОм.

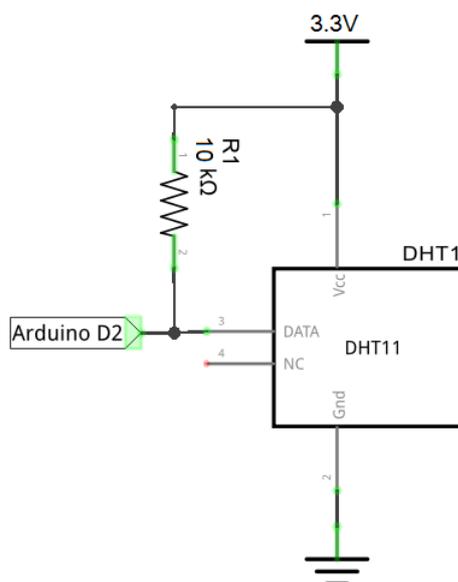


Рисунок 35 – Схема подключения DHT11

Выход сигнала V_{out} аналогового датчика TMP35 подключается к аналоговому входу модуля ADS1015, который в свою очередь подключен к микроконтроллеру. Остальные два контакта $+V_s$ и GND используются в качестве питания. Схема подключения изображена на рисунке 36.

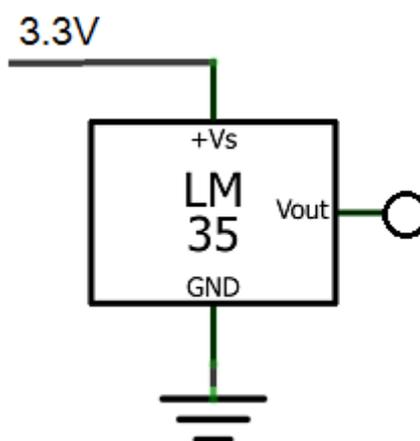


Рисунок 36 – Схема подключения TMP35

Для подключения термистора NTC 3950 используем резистор в 100 кОм в качестве номинального резистора. Одним концом термистор подключается к питанию микроконтроллера, а вторым к GND. Также этот

конец присоединяется к аналоговому входу модуля ADS1015 через резистор. Схема изображена на рисунке 37.

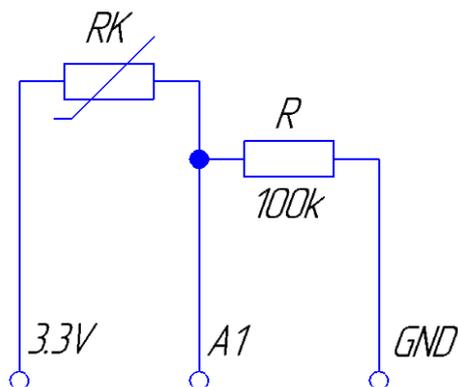


Рисунок 37 – Схема подключения термистора NTC 3950

Вывод по разделу

Таким образом, мы просмотрели схемы включения всех элементов, используемых в данном устройстве. Объяснили, как они работают и в чем состоит их принцип. Полная схема электрическая принципиальная приведена на рисунке 38. Оформление схемы производилось в системе автоматизированного проектирования КОМПАС-3D.

Для начала нужно подключить необходимые библиотеки для работы с микроконтроллером ESP8266:

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
```

Следующим, что мы подключим, будет библиотека для считывания времени с NTP-сервера в интернете. Network Time Protocol – сетевой протокол для синхронизации внутренних часов компьютера с использованием сетей с переменной латентностью, основанных на коммутации пакетов:

```
#include <time.h>
```

Далее задаем название и пароль нашей Wi-Fi сети, чтобы микроконтроллер смог подключиться и получить свой статический IP. Также задаем порт для web-сервера:

```
const char* ssid = "*****";
const char* password = "*****";
ESP8266WebServer server(8001);
```

Следующим обозначаем переменную, отвечающую за количество точек. В данном случае их количество будет равно 24 с расчетом на каждый час. Задаем переменные и даем установку на запись данных с датчиков и времени на каждую точку:

```
const uint16_t lengt=24; //max kol toчек trenda
uint16_t tick=0;
time_t tnow[lengt];
float h[lengt], t[lengt], w[lengt];
```

Теперь прописываем функцию для обращения к библиотеке Google Charts. Здесь происходит построение web-страницы и загрузка скриптов:

```
void handleRoot() {
  String trendstr;

  trendstr = F("<html>\
<head>\
<script type='text/javascript'
src='https://www.gstatic.com/charts/loader.js'></script>\
<meta http-equiv='refresh' content='1000'/>\
<title>Daily temperature</title>\
<script type='text/javascript'>\
google.charts.load('current', {'packages':['corechart']});\
google.charts.setOnLoadCallback(drawChart);\
\

```

Далее даем названия переменным, данные которых используются в отображении при рисовании графика:

```
function drawChart() {\
  var data = new google.visualization.DataTable();\
  data.addColumn('datetime', 'Time');\
  data.addColumn('number', 'DHT's Temperature, C');\
  data.addColumn('number', 'TMP's Temperature, C');\
  data.addColumn('number', 'Re's Temperature, C');\
\

```

Следующим действием задаем условия, благодаря которым будет происходить сама отрисовка:

```
data.addRows(["]);
uint16_t k, y=0;
for (int i=1; i <= lengt; i++){
  k = tick-1 + i;
  if (h[k]>0){
```

```

if (y>0) trendstr += ",";
y ++;
if (k>lengt-1) k = k - lengt;

```

И задаем массивы для отрисовки:

```

trendstr += "[new Date(";
trendstr += String(tnow[k]-(2*3600));
trendstr += "*1000), ";
trendstr += t[k];
trendstr += ", ";
trendstr += h[k];
trendstr += ", ";
trendstr += w[k];
trendstr += "]\n";
}
}
trendstr += F(")\n";
\

```

Задаем ширину графика, название, расположение легенды и формат даты, которая отображает, когда было произведено измерение температуры:

```

var options = {width: '100%',\
title: 'Graph of Temperature',\
curveType: 'function',\
legend: { position: 'bottom' },\
hAxis: {format: 'dd.MM.yyyy HH:mm',\
gridlines: {\
count: 10,\
},\
}\
};\
\

```

Заканчиваем прописывание обращения к библиотеке Google Charts:

```

var chart = new
google.visualization.LineChart(document.getElementById('curve_chart'));\

```

```

var formatter = new google.visualization.DateFormat({pattern:
'dd.MM.yyyy HH:mm'});\
formatter.format(data, 0);\
chart.draw(data, options);\
}\
</script>\
</head>\
<body>\
<div id='curve_chart' style='width: 100%; height: 600px'></div>\
</body>\
</html>");
server.send ( 200, F("text/html"), trendstr );
}

```

Последним указывается частота записи точек графика в миллисекундах.

В данном случае запись происходит раз в час:

```

const long interval = 3600000;
unsigned long previousMillis = 0;

```

Запись точек происходит с временными метками. Время для них каждый раз берется с серверов NTP. Цифра «2» обозначает часовой пояс:

```

configTime(2*3600, 0, "pool.ntp.org", "time.nist.gov");

```

Следующим шагом устанавливается Интернет-соединение и запускается Wi-Fi.

5.2. Работа с модулем ADS1015

Все происходит весьма просто и быстро. Для корректной работы модуля подключаем две библиотеки:

```
#include <Wire.h>
#include <Adafruit_ADS1015.h>
```

Далее задаем один из 4-х возможных адресов. Учитывая то, как был подключен модуль, то адрес будет 0x48. Запускаем само устройство:

```
Adafruit_ADS1015 ads(0x48);
ads.begin();
```

Теперь объявляем адреса, на которые будут поступать информация с аналоговых датчиков:

```
int16_t adc0, adc1;
```

5.3. Работа с датчиком DHT11

Объявляем пин, который будет использоваться для работы с датчиком, и его тип. Запускаем:

```
#define DHTPIN 5
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

dht.begin();
```

Так как нам нужна только температура, значения влажности мы игнорируем, поэтому для считывания информации используется одна команда:

```
h[tick] = dht.readTemperature();
```

5.4. Работа с датчиком TMP35

Датчик подключается к модулю ADS1015 и чтение данных происходит с пина A0. Так как датчик измеряет напряжение, для пересчета его в градусы Цельсия воспользуемся формулой (1), представленной ранее. Напомню, что число «4096» - это максимальное число, которое может вернуть 12-разрядный Аналого-Цифровой Преобразователь на модуле ADS1015:

```
t[tick] = Solid();

int Solid();
  adc0 = ads.readADC_SingleEnded(0);
  temp = ( adc0/4096.0 )*3.3*1000/10;
  delay(1000);
  return adc0;
```

5.5. Работа с термистором

Процесс аналогичен предыдущему. Термистор подключается к пину A1 модуля ADS1015. Пересчет напряжения в градусы Цельсия производится по формулам (2) и (3):

```
w[tick] = Snake();

int Snake() {
  adc1 = ads.readADC_SingleEnded(1);
  float tr = 4096.0 / adc1 - 1;
  tr = SERIAL_R / tr;

  float steinhart;
  steinhart = tr / THERMISTOR_R; // (R/Ro)
  steinhart = log(steinhart); // ln(R/Ro)
  steinhart /= B; // 1/B * ln(R/Ro)
  steinhart += 1.0 / (NOMINAL_T); // + (1/To)
  steinhart = 1.0 / steinhart; // Invert
  Serial.println(steinhart);
  delay(1000);
  return adc1;
```

Вывод по разделу

Таким образом, мы написали код программы данного устройства. Полностью код представлен в приложении А. Выше сказано, как именно прописывать обращение к библиотеке Google Charts, как вырисовывать график на web-странице, как устанавливать связь с NTP-сервером. Объяснили, для чего нужен модуль ADS1015, рассказали о специфической особенности, такой как адрес, и как правильно его подключать. Также описали, как происходит работа каждого датчика в отдельности, прописали обращение к ним. Показали, как снимаются данные и как происходит пересчет напряжения в случае с термистором и TMP35.

6. Описание алгоритма работы устройства

Подключаем ESP8266 к компьютеру посредством USB-кабеля, для того чтобы включить микроконтроллер. После запуска по ходу выполнения программы он подключается к сети Wi-Fi и вырисовывает график на веб-странице, на котором дальше будут показаны линии изменения температуры. Далее датчики высылают информацию раз в час, зависит от того, как настроена программа, и на графике вырисовываются три тенденции, каждая соответствует определенному датчику, названия которых указаны в легенде. На оси абсцисс вырисовываются временные метки, когда были сделаны измерения из-за связи с NTP сервером.

Вывод по разделу

Таким образом, была объяснена работа разрабатываемого устройства. Блок-схема алгоритма изображена на рисунке 39.

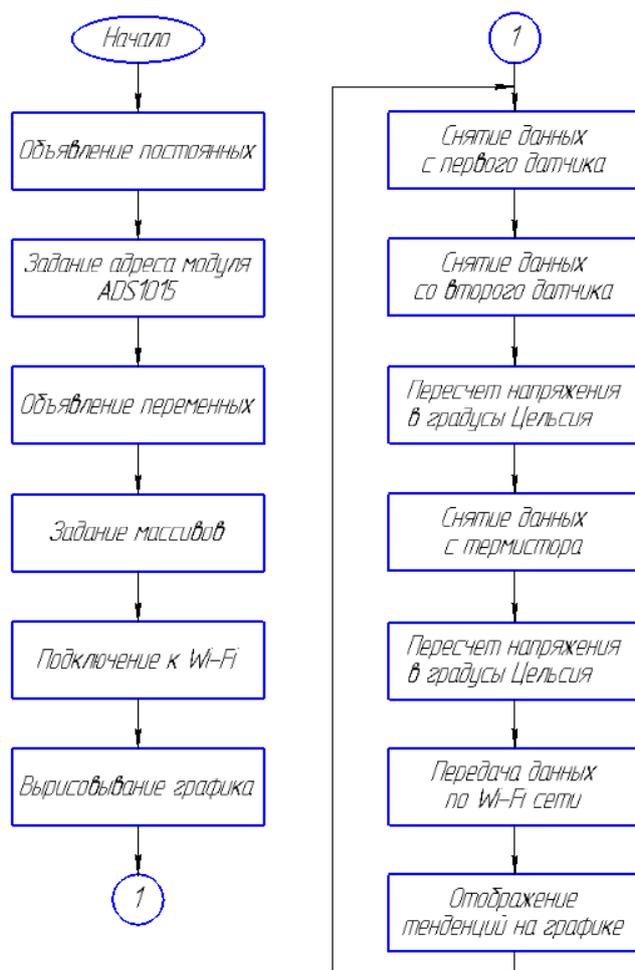


Рисунок 39 – Блок-схема алгоритма

7. Интерфейс пользователя

В качестве интерфейса выступает сам браузер, а именно нарисованный на web-странице график изменения температуры. Единственным возможным действием для пользователя выступает кнопка обновления страницы. Нажимая на нее, добавляются новые точки и дорисовываются линии.

Это может являться недостатком, если Вы захотите следить за температурой каждую минуту, так как Вам постоянно придется обновлять web-страницу.

Вывод по разделу

Таким образом, Вы можете представить себе взаимодействие пользователя с прибором. Пример результата работы программы представлен на рисунке 40.

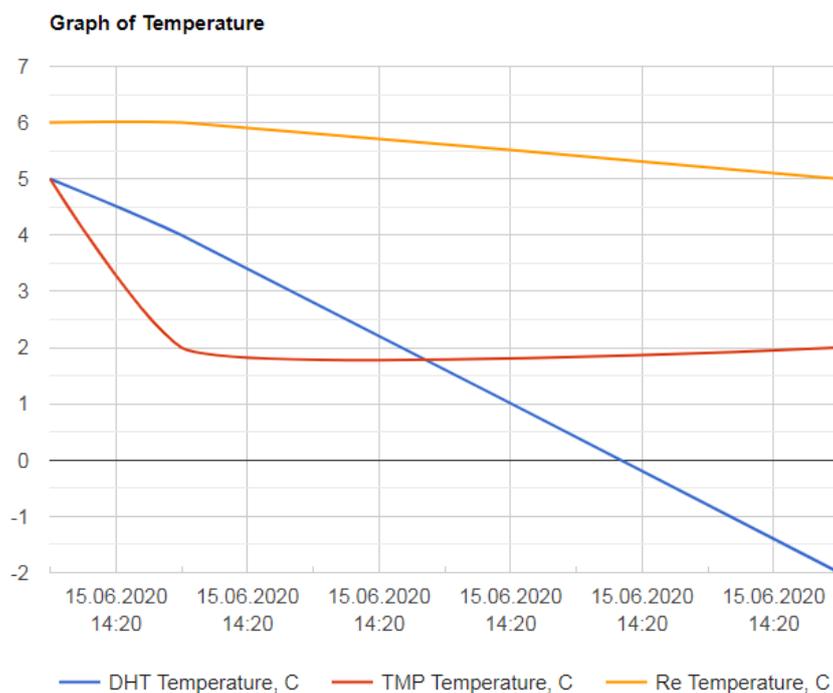


Рисунок 40 – Результат работы программы

8. Экономический расчет

Стоимость каждого отдельно взятого элемента и общая стоимость полученного устройства показаны в таблице 2.

Таблица 2 – Расчет стоимости компонентов

№	Наименование	Количество	Цена, руб.
1	Микроконтроллер ESP8266 NodeMCU v3	1	299,78
2	Модуль ADS1015	1	96
3	Датчик DHT11	1	76,95
4	Датчик TMP35	1	60
5	Термистор NTC3950	1	54,51

Итого: 587,24 руб.

Вывод по разделу

Выше перечислены все элементы, присутствующие в разрабатываемом устройстве, их цены и количество. Сравнивая с аналогами, которые используются в настоящее время можно сказать, что данный прибор намного дешевле.

Заключение

В конечном итоге, спроектированное устройство отвечает заявленным требованиям, имеют требуемый функционал при относительно небольшой стоимости компонентов. Программный код отлажен и позволяет осуществить слежение за изменением температуры со всеми требуемыми функциями.

В данной ВКР были проанализированы известные решения, обозначены требования к проектируемому устройству, составлена структурная схема, проведен поиск и подбор компонентной базы, протестирована работоспособность приобретенных модулей, составлены электрическая принципиальная схема, алгоритм действий программы, составлен и отлажен программный код, создан и нарисован интерфейс для простого понимания конечному пользователю.

В дальнейшем возможно улучшение устройства, которое поможет сравнить его с аналогами.

Так, например, можно увеличить количество и разнообразие датчиков, что поможет следить не только за температурой, но и за давлением, влажностью, дымом, движением. Такие датчики как DHT11 и DHT22 уже в состоянии измерять и влажность, и температуру. Так как библиотека Google Charts не ограничивает нас в числе линий, нарисованных на графике, остается только прописать обращение для них в коде. Можно расширить функционал устройства, подключив датчики двери, протечки, наличия 220В, удара. Но это уже никак не будет связано с мониторингом температуры, а только добавит новые возможности устройству.

Также я считаю возможным добавление к функционалу данного прибора способности предупреждать о приближении к опасной температуре. Подобное используется в нескольких устройствах, описанных выше. Например, оповещение пользователя о возможном чрезвычайном происшествии при помощи SMS.

Список используемой литературы и используемых источников

1. Google Charts [Электронный ресурс] // Примеры кода для графиков: <https://developers.google.com/chart/interactive/docs/gallery/linechart?hl=ru>
2. Приборы для измерения температуры FLUKE [Электронный ресурс] // Визуальные пирометры и термометры: <https://flukeshop.ru/pribory-dlya-izmereniya-temperatury>
3. Термоиндикаторы LogTag [Электронный ресурс] // Контроль температуры: <http://logtagrussia.ru/>
4. Микроконтроллер ESP8266 [Электронный ресурс] // Прошивка и пример использования: <https://tproger.ru/articles/about-esp8266/>
5. NetPing [Электронный ресурс] // Устройства мониторинга температуры в серверной комнате: <http://www.netping.ru/Blog/kak-organizovat-monitoring-temperature-v-servernoj-komnate>
6. Модуль ADS1015 [Электронный ресурс] // ADS1115 analog –to-digital converter and ESP8266: <http://www.esp8266learning.com/ads1115-analog-to-digital-converter-and-esp8266.php>
7. ESP-12E [Электронный ресурс] // Basic flashing connection: https://www.letscontrolit.com/wiki/index.php?title=File:ESP-12E_Basic_flashing_connection.jpg
8. NodeMCU распиновка [Электронный ресурс] // Назначение пинов на микроконтроллере: <https://myrobot.ru/wiki/index.php?n=Experiences.NodemcuPinout>
9. Датчики и измерения [Электронный ресурс] // Считывание показаний датчиков с помощью Arduino: <https://radioprogram.ru/post/224>
Thermosafer XST200 Smart [Электронный ресурс] // Характеристика: <https://www.ebay.com/itm/Thermosafer-XST200-Smart-Monitoring-Body-Bluetooth-Contact-Thermometer-Korea/392200897609>

10. Датчик LM35 [Электронный ресурс] // Подключение к Arduino:
<https://robotclass.ru/tutorials/arduino-temperature-sensor/>
11. ADS1115 (ADS1015) [Электронный ресурс] // Внешний I2C ADS АЦП: https://flprog.ru/supported_hardware/prochie-moduli/ads1115-ads1015-vneshnij-i2c-adc-acp-podderzhivaetsja-prog/
12. NTP-server [Электронный ресурс] // Настройка: <https://www.ntp-servers.net/>
13. Датчик DHT11 [Электронный ресурс] // DHT11 Humidity and Temperature Sensor: <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>
14. TMP35 [Электронный ресурс] // Data Sheet: https://www.analog.com/media/en/technical-documentation/data-sheets/TMP35_36_37.pdf
15. Термистор NTC3950 [Электронный ресурс] // Data Sheet: <https://www.makeralot.com/download/Reprap-Hotend-Thermistor-NTC-3950-100K.pdf>
16. NodeMCU V3 [Электронный ресурс] // PIN Mapping and Flash Layout:
<https://docs.zerynth.com/latest/official/board.zerynth.nodemcu3/docs/index.html>
17. Построение графика [Электронный ресурс] // WI-FI ESP8266 Суточный график температуры и влажности:
http://geekmatic.in.ua/grafik_temperaturi_esp8266
18. Устройства измерения температуры [Электронный ресурс] // Промышленный импорт: <https://dmliefer.ru/katalog/kip/pribery-dlja-izmerenija-temperature>
19. Датчик DS18B20 [Электронный ресурс] // Measure Temperature on the Web With Arduino Ethernet and DS18B20:
<https://evercare.ru/news/medica-2019-termometr-dlya-nepreryvnogo-monitoringa-temperature-tela>

20. Thermosafer XST200 Smart [Электронный ресурс] // Термометр для непрерывного мониторинга температуры: <https://evercare.ru/news/medica-2019-termometr-dlya-nepreryvnogo-monitoringa-temperature-tela>
21. Улли Соммер Программирование микроконтроллерных плат Arduino/Freeduino под ред. Добина Григория. М. : ВHV, 2012. 256 с.
22. Ревич Ю. В. Занимательная электроника. М. : ВHV, 2018. 672 с.
23. Виктор Петин Проекты с использованием контроллера Arduino, 2-е издание. М. : ВHV, 2015. 464 с.
24. Теро Карвинен, Киммо Карвинен, Вилле Валтокари Делаем сенсоры. Проекты сенсорных устройств на базе Arduino и Raspberry Pi. М. : Вильямс, 2015. 448 с.
25. Jennie Schaeffer, Rikard Lindell Arduino in Museum Exhibition: Lessons Learned When Working With Design Students Inexperienced in Coding. P. : TEI'15, 2015. Pages 715-720.
26. Fernando S. Perilla Fire Safety and Alert System Using Arduino Sensors with IoT Integtation. P. : ICSCA 2018, 2018. Pages 199-203
27. Lai Ah-Fur, Chen Chien-Hung, Lai Horng-Yih Developing an Arduino Simulation-based Learning System and Evaluating its Suitability. P. : ICEBT 2018, 2018. Pages 38-42.
28. Caleb Helbling, Samuel Z. Guyer Juniper: a functional reactive programming language for the Arduino. P. : FARM 2016, 2016. Pages 8-16
29. David Sirkin, Nikolas Martelaro, Wendy Ju Make This! Introducing to Electronics Prototyping Using Arduino. P. : CHI EA'17, 2017. Pages 1224-1227.

```

#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <time.h>
#include <OneWire.h>
#include <DHT.h>
#include <Wire.h>
#include <Adafruit_ADS1015.h>

#define DHTPIN 14
#define DHTTYPE DHT11// DHT 11
#define B 3950 // B-коэффициент
#define SERIAL_R 102000 // сопротивление последовательного резистора, 102 кОм
#define THERMISTOR_R 100000 // номинальное сопротивление термистора, 100 кОм
#define NOMINAL_T 25 // номинальная температура (при которой TR = 100 кОм)

DHT dht(DHTPIN, DHTTYPE);

Adafruit_ADS1015 ads(0x48);

const char* ssid = "AIST 135";
const char* password = "1820250732";
ESP8266WebServer server(8001);

const uint16_t lengt=24; //max kol toчек trenda
uint16_t tick=0;
time_t tnow[lengt];
float h[lengt], t[lengt], w[lengt];

void handleRoot() {
String trendstr;

trendstr = F("<html>\
<head>\
<script type='text/javascript' src='https://www.gstatic.com/charts/loader.js'></script>\
<meta http-equiv='refresh' content='1000'/>\
<title>Daily temperature</title>\
<script type='text/javascript'>\
google.charts.load('current', {'packages':['corechart']});\
google.charts.setOnLoadCallback(drawChart);\
\
function drawChart() {\
var data = new google.visualization.DataTable();\
data.addColumn('datetime', 'Time');\
data.addColumn('number', 'DHT's Temperature, C');\
data.addColumn('number', 'TMP's Temperature, C');\
data.addColumn('number', 'Re's Temperature, C');\
\
data.addRows(["]);
uint16_t k, y=0;
for (int i=1; i <= lengt; i++){
k = tick-1 + i;

if (h[k]>0){
if (y>0) trendstr += ",";

```

Продолжение Приложения А

```

y ++;
if (k>lengt-1) k = k - lengt;
trendstr += "[new Date(";
trendstr += String(tnow[k]-(2*3600)); //2- chasovoy poyas
trendstr += "*1000), ";
trendstr += t[k];
trendstr += ", ";
trendstr += h[k];
trendstr += ", ";
trendstr += w[k];
trendstr += "]";
}
}
trendstr += F(")];\
\
var options = { width: '100%',\
title: 'Graph of Temperature',\
curveType: 'function',\
legend: { position: 'bottom' },\
hAxis: {format: 'dd.MM.yyyy HH:mm',\
gridlines: {\
count: 10,\
},\
},\
};\
\
var chart = new google.visualization.LineChart(document.getElementById('curve_chart'));\
var formatter = new google.visualization.DateFormat({pattern: 'dd.MM.yyyy HH:mm'});\
formatter.format(data, 0);\
chart.draw(data, options);\
}\
</script>\
</head>\
<body>\
<div id='curve_chart' style='width: 100%; height: 600px'></div>\
</body>\
</html>");
server.send ( 200, F("text/html"), trendstr );
}

const long interval = 3600000; //interval zapisi danih dlya trenda
unsigned long previousMillis = 0;
void setup(void){

pinMode(adc0,INPUT);
pinMode(adc1,INPUT);

Serial.begin(115200);
WiFi.begin(ssid, password);

configTime(2*3600, 0, "pool.ntp.org", "time.nist.gov"); //NTP sayti dlia schitivaniya vremeni

Serial.println("");
// Wait for connection

```

Продолжение Приложения А

```

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.print("Connected to ");
Serial.println(ssid);
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
server.on("/", handleRoot);
server.begin();
Serial.println("HTTP server started");
ads.begin();
dht.begin();
delay(2000);
tochka();
}
void loop(void){
  server.handleClient();
  int16_t adc0, adc1;
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
    tochka();
  }
}
void tochka(){
  h[tick] = dht.readTemperature();
  t[tick] = Solid();
  w[tick] = Snake();
  tnow[tick] = time(nullptr);
  Serial.print(String(ctime(&tnow[tick])));
  if (tick<lengt-1) tick ++; else tick = 0;
}
int Solid();
  adc0 = ads.readADC_SingleEnded(0);
  temp = ( adc0/4096.0 )*3.3*1000/10;
  delay(1000);
  return adc0;
int Snake() {
  adc1 = ads.readADC_SingleEnded(1);
  float tr = 4096.0 / adc1 - 1;
  tr = SERIAL_R / tr;
  float steinhart;
  steinhart = tr / THERMISTOR_R; // (R/Ro)
  steinhart = log(steinhart); // ln(R/Ro)
  steinhart /= B; // 1/B * ln(R/Ro)
  steinhart += 1.0 / (NOMINAL_T); // + (1/To)
  steinhart = 1.0 / steinhart; // Invert
  Serial.println(steinhart);
  delay(1000);
  return adc1;
}

```