Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра радиотехнических систем (РТС)

**РЕАЛИЗАЦИЯ УПРАВЛЕНИЕ МОДУЛЕМ «SIM808-EVB» ПОСРЕДСТВОМ WEB-ИНТЕРФЕЙСА НА БАЗЕ ПЛАТЫ «ORANGE PI»**

Пояснительная записка к курсовой работе

по дисциплине «Цифровые устройства и микропроцессоры»

Выполнил: студент гр.121-3

\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Усанов

« » \_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г.

Руководитель:

Старший преподаватель каф. РТС

\_\_\_\_\_\_\_\_\_\_\_\_\_ П.А. Полянских

« » \_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г.

Томск 2025

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра радиотехнических систем

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  Зав. кафедрой РТС  Аникин А.С.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
|  | «\_\_\_» 2025 г. |
|  |  |

ЗАДАНИЕ

на курсовую работу по дисциплине

«Цифровые устройства и микропроцессоры»

студенту гр.121-3 радиотехнического факультета

Усанову Владиславу Валерьевичу

(Ф.И.О студента)

1. Тема курсовой работы: Реализация управление модулем «SIM808-EVB» посредством WEB-интерфейса на базе платы «Orange Pi»;
2. Цель курсовой работы: Получение данных GPS позиционирования с «SIM808-EVB» и их отображение в WEB-интерфейсе, размещаемом на «Orange Pi»;
3. Задачи курсовой работы:

3.1 Изучение технической документации используемого оборудования с разбором функциональных возможностей.

3.2 Установление связи между оборудованием.

3.3 Разработка WEB-интерфейса.

3.4 Отладка разработанного WEB-интерфейса непосредственно в ходе работы системы.

Руководитель:

|  |  |  |
| --- | --- | --- |
| Ст. преподаватель каф. РТС  (должность) | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Подпись) | П.А. Полянских  (Ф.И.О.) |

Задание принял к исполнению «\_\_\_\_» 2025 г.

|  |  |  |
| --- | --- | --- |
| Студент гр. 121-3 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | В.В. Усанов  (Ф.И.О.) |

**Оглавление**

[Введение 5](#_Toc199100354)

[1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 6](#_Toc199100355)

[1.1 Основные определения 6](#_Toc199100356)

[1.1.1 HTML 6](#_Toc199100357)

[1.1.2 CSS 6](#_Toc199100358)

[1.1.3 JavaScript 6](#_Toc199100359)

[1.1.4 Bootstrap 6](#_Toc199100360)

[1.1.5 POST и GET в CGI 7](#_Toc199100361)

[1.1.6 JSON, AJAX 8](#_Toc199100362)

[1.1.7 Linux 9](#_Toc199100363)

[1.1.8 BASH и основные команды для работы 9](#_Toc199100364)

[1.1.9 Файловая система 10](#_Toc199100365)

[1.1.10 Редактор Vi 11](#_Toc199100366)

[1.1.11 Git, Gitlab 12](#_Toc199100367)

[1.1.12 Telnet 12](#_Toc199100368)

[1.1.13 SSH 13](#_Toc199100369)

[1.1.14 PuTTY 13](#_Toc199100370)

[1.1.15 Make, cmake 14](#_Toc199100371)

[1.1.16 Сеть 14](#_Toc199100372)

[1.1.17 Ip, netmask, gateway, inconfig 16](#_Toc199100373)

[1.1.18 IP, UDP, TCP 18](#_Toc199100374)

[1.1.19 UART 19](#_Toc199100375)

[1.1.20 Orange PI 19](#_Toc199100376)

[1.1.21 АТ – команды 20](#_Toc199100377)

[1.1.22 GSM 22](#_Toc199100378)

[1.1.23 NMEA 22](#_Toc199100379)

[1.1.24 WebSocket 22](#_Toc199100380)

[1.1.25 Lighttpd 23](#_Toc199100381)

[1.1.26 Wstunnel 23](#_Toc199100382)

[1.2 Информация о используемом оборудовании 23](#_Toc199100383)

[1.2.1 SIM808 23](#_Toc199100384)

[1.2.2 Orange Pi3 24](#_Toc199100385)

[2 ПРАКТИЧЕСКАЯ ЧАСТЬ 27](#_Toc199100386)

[2.1 Физическое взаимодействие 27](#_Toc199100387)

[2.1.1 Прямое подключение рабочего ПК к SIM808 27](#_Toc199100388)

[2.1.2 Подключение Orange Pi к ПК 27](#_Toc199100389)

[2.1.3 Подключение Orange Pi к SIM808 27](#_Toc199100390)

[2.1.4 Отправка команд через Orange Pi на SIM808 27](#_Toc199100391)

[2.1.5 Прямое подключение рабочего ПК к SIM808 28](#_Toc199100392)

[2.2 Написание кода WEB-платформы 29](#_Toc199100393)

[Заключение 31](#_Toc199100394)

[Приложение А Демонстрация основных блоков WEB-страницы на языке «HTML» 35](#_Toc199100395)

[Приложение Б Демонстрация некоторых изменений стиля страницы на языке «CSS» 36](#_Toc199100396)

Введение

Комплексы устройств, позволяющих определять расположение по данным GPS, используются в навигационном оборудовании морского, наземного и даже воздушного транспорта.

Основной задачей было реализовать управление модулем «SIM808-EVB» посредством WEB-интерфейса на базе платы «Orange Pi».

Выполнение данной задачи было разделено на отдельные подзадачи, такие как:

* изучение теории, касаемой ОС, на которой велась разработка (Ubuntu);
* изучение характеристик устройств, с которыми, непосредственно, и проводилась работа;
* установление соединения между устройствами;
* написание софта для работы WEB-интерфейса.

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ
   1. Основные определения
      1. HTML

HTML5 — последняя и наиболее мощная редакция языка разметки HTML (HyperText Markup Language/язык разметки гипертекста), включающая новые компоненты, атрибуты и функциональность [1]. Под термином также понимается набор технологий для создания сайтов и веб-приложений.

* + 1. CSS

CSS — это язык описания внешнего вида документа, то есть он отвечает за то, как выглядят веб-страницы: цвет фона и декоративных элементов, размер и стиль шрифтов. (Cascading Style Sheets/каскадные таблицы стилей) [2]. CSS взаимодействует с другим языком разметки — HTML, который отвечает за размещение элементов на странице.

Весь код, написанный на CSS, прописывается в отдельном внешнем файле с расширением “.css”. Его подключают к HTML-странице при помощи тега <link href=”style.css” rel=”stylesheet”>.

* + 1. JavaScript

Интерпретируемый язык программирования, который используют для написания frontend-/backend- частей сайтов, а также мобильных приложений [3].

* + 1. Bootstrap

Открытый и бесплатный HTML-, CSS- и JS-фреймворк, который используют веб-разработчики для быстрой верстки адаптивных дизайнов сайтов и веб-приложений [4]. Включает в себя CSS-/HTML-шаблоны оформления для веб-форм, меток, типографики, кнопок, блоков навигации и других компонентов веб-интерфейса.

* + 1. POST и GET в CGI

CGI (Common Gateway Interface/общий интерфейс шлюза) — стандарт интерфейса, используемого для связи внешней программы с веб-сервером [5]. Программу, которая работает по такому интерфейсу совместно с веб-сервером, принято называть шлюзом, хотя многие предпочитают названия «скрипт» (сценарий) или «CGI-программа».

Обобщенный алгоритм работы через CGI можно представить в следующем виде:

* Клиент запрашивает CGI-приложение по его URI.
* Веб-сервер принимает запрос и устанавливает переменные окружения, через них приложению передаются данные и служебная информация.
* Веб-сервер перенаправляет запросы через стандартный поток ввода (stdin) на вход вызываемой программы.
* CGI-приложение выполняет все необходимые операции и формирует результаты в виде HTML.
* Сформированный гипертекст возвращается веб-серверу через стандартный поток вывода (stdout). Сообщения об ошибках передаются через stderr.
* Веб-сервер передает результаты запроса клиенту.

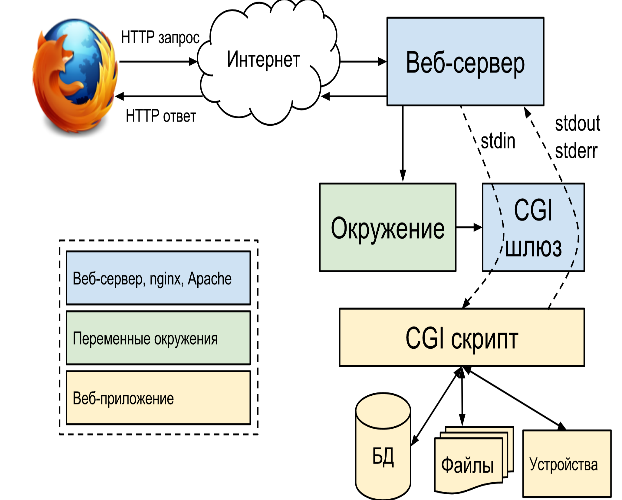


Рисунок 1.1 – Модель передачи запросов в CGI скрипты

Наиболее частая задача, для решения которой применяется CGI — создание интерактивных страниц, содержание которых зависит от действий пользователя. Типичными примерами таких веб-страниц является форма регистрации на сайте или форма для отправки комментария. Другая область применения CGI, остающаяся за кулисами взаимодействия с пользователем, связана со сбором и обработкой информации о клиенте: установка и чтение «печенюшек»-cookies; получение данных о браузере и операционной системе; подсчет количества посещений веб-страницы; мониторинг веб-трафика и т.п.

Основа передачи данных в вебе — это HTTP (HyperText Transfer Protocol), клиент-серверный протокол взаимодействия. Работает он следующим образом:

Клиент, обычно веб-браузер, инициирует запрос к серверу. В ответ он получает необходимый документ — HTML-страницу — или сообщение об ошибке, если запрос сформирован неправильно или документа нет на сервере.

* GET используется для получения данных от сервера. Чтобы создать запрос, введите в адресную строку браузера любой URL сайта (возможен поиск по параметрам). Метод для получения данных с сервера. Он передаётся с URL, поэтому виден в адресной строке браузера или истории посещений сайтов. Применяется в фильтрах поисковиков и онлайн-магазинов.
* POST — метод отправки данных на сервер, например, после заполнения формы регистрации или авторизации на сайте. Информация передаётся в теле запроса и не может быть перехвачена злоумышленниками.
  + 1. JSON, AJAX

JSON (JavaScript Object Notation) — это формат передачи данных, который используется при взаимодействии веб-сервера и браузера [6]. До создания формата JSON веб-сайты работали медленно. Каждый запрос пользователя к серверу требовал повторной отправки в браузер обновленной HTML-страницы. AJAX-запрос, использующий формат JSON, выполняется в фоновом режиме, поэтому страница не перезагружается. Сегодня JSON — это стандарт передачи данных в интернете. Основа JSON — это синтаксис объектов JavaScript, которые используются в языке для представления данных. Данные в формате JSON хранятся как пары «ключ:значение» и упорядоченные списки.

* + 1. Linux

Linux — семейство Unix-подобных операционных систем на базе одноименного ядра [7]. Нет единой операционной системы Linux, как, например, Windows или MacOS. Есть множество дистрибутивов (набор файлов, необходимых для установки ПО), выполняющих конкретные задачи.

Ядро – так называемая главная программа, основная часть ОС. Ядро выступает в качестве посредника между устройствами ПК (видеокартой, процессором, оперативной памятью и пр.) и его ПО. Для этого в коде есть драйверы устройств. Они загружаются в память или подключаются по мере необходимости ресурса определенного устройства. Ядра управляют устройствами, процессами и памятью, обрабатывают системные вызовы.

Системные библиотеки — это специальные программы, дающие доступ к функциям ядра. Для выполнения какой-либо задачи ядро вначале получает системный вызов, исходящий от приложений. Но у каждого ядра свой набор системных вызовов, и они должны понимать формат выполнения задачи. Поэтому программисты разработали стандартную библиотеку процедур, описывающую набор системных вызовов для конкретной ОС.

* + 1. BASH и основные команды для работы

Bash (Bourne-Again SHell) — это командная оболочка для UNIX-подобных операционных систем (UNIX, GNU/Linux, MacOS) [8]. Она дает пользователю систему команд для работы с файлами и папками, поиском, настройкой окружения и позволяет управлять ОС прямо из командной строки. Оболочка принимает команды, которые пользователь ввел в командную строку, и интерпретирует их, то есть переводит в машинный код. Операционная система получает код в качестве инструкций и выполняет их.

Основные команды для работы:

1. sudo … (выполнение команды с правами администратора);
2. sudo -i (переключение терминала в режим администратора);
3. netstat -lntup| grep LISTEN (просмотр процессов, занимающих виртуальные порты системы);
4. kill [port] (прекращение работы процесса, занимающего введённый порт);
5. nc [ip] [port] (подключение к процессу по введенному ip и порту для отправки сообщений);
6. ssh root@[ip] (установление ssh соединения с устройством по ip адресу с правами администратора);
7. cd (переход в корневую папку);
8. cd / (переход по пути, введённому после /);
9. ./ (запуск исполняемого файла, введенному после /);
10. microcom -s[speed] -t[host:port] (подключение к устройствам через последовательный интерфейс RS232);
11. mc (открытие midnight commander);
12. nautilus (открытие проводника);
    * 1. Файловая система

Файловая система — это способ хранить и организовывать информацию на каком-нибудь носителе [9]. От файловой системы зависит, как файлы будут кодироваться, храниться на диске и читаться компьютером. У каждой операционной системы свой тип организации файлов, то есть своя файловая система.

Файл — это блок информации, записанный на накопитель. Не всегда эта информация записана последовательно, иногда физически части одного и того же файла могут находиться в разных секторах жесткого диска. Чтобы знать, что где хранится, у системы есть файловая таблица: там приведена информация обо всех файлах.

Файловая система определяет, как организовать эту таблицу. Способы разные в зависимости от ОС, поэтому у разных операционных систем различаются и файловые системы.

* + 1. Редактор Vi

Редактор vi — это классический текстовый редактор в семействе Linux [10].

Он доступен почти во всех дистрибутивах Linux и работает одинаково на разных платформах и дистрибутивах.

Редактор vi работает в трёх режимах:

1. Командный режим. В этом режиме можно перемещать курсор, вырезать, копировать и вставлять текст. Также он сохраняет внесенные изменения.
2. Режим вставки. Этот режим предназначен для вставки текста в файл. Чтобы переключиться в режим вставки, нужно нажать клавишу «i» на клавиатуре.
3. Режим последней строки. В этом режиме курсор перемещается на последнюю строку экрана, и vi ждёт команды. Он позволяет выполнять такие задачи, как сохранение файлов и выполнение команд.

Чтобы запустить редактор vi, нужно открыть терминал и ввести команду vi <filename\_NEW> или <filename\_EXISTING>. Если указан существующий файл, то редактор откроет его для редактирования. В противном случае можно создать новый файл.

* + 1. Git, Gitlab

Git — это система контроля версий, которая помогает отслеживать историю изменений в файлах [11]. Возможности:

1. возврат к любой предыдущей версии кода;
2. просмотр истории изменений;
3. параллельная работа над проектом;
4. backup кода.

Репозиторий — это все файлы, находящиеся под контролем версий вместе с историей их изменения и другой служебной информацией. В GIT используется распределенная система контроля версий — репозиторий продублирован в облаке и на ПК пользователя, и после внесения обновлений пользователя в свой репозиторий, он синхронизируется в облаке.

Коммит — (фиксировать) операция, которая берет все подготовленные изменения и отправляет их в репозиторий как единое целое. Данная операция нужна для разделения процесса синхронизации на отдельные шаги.

Ветвление — позволяет решать отдельные задачи, не вмешиваясь в основную линию разработки.

GitLab — веб-инструмент жизненного цикла DevOps с открытым исходным кодом. Представляет собой систему управления репозиториями кода для Git с собственной вики, системой отслеживания ошибок и тд. Схож с Jira (п.19). Сборка проектов происходит на отдельной платформе Docker.

* + 1. Telnet

Telnet - сетевой протокол для реализации текстового терминального интерфейса по сети (при помощи TCP) [12]. Выполняет функции протокола прикладного (верхнего) уровня OCI. Протокол telnet использовался для удалённого администрирования различными сетевыми устройствами и программными серверами, но уступил SSH из-за безопасности.

Тем не менее он может являться единственной возможностью взаимодействовать через CLI со встраиваемыми системами, например, с маршрутизаторами, так как на них отсутствует ssh.

* + 1. SSH

SSH - (secure shell/безопасная оболочка) — сетевой протокол для удаленного управления ОС с помощью командной строки и передачи данных в зашифрованном виде [13].

**Принцип работы SSH:**

В основе протокола лежит криптография — шифрование канала связи.

Компьютер, с которого устанавливается SSH-соединение, называется клиентом, а удалённый SSH-сервер, к которому осуществляется подключение — хостом.

Подключение по SSH использует логин на сервере, а SSH-пароль является паролем этого пользователя.

SSH-сервер передаёт данные входящих соединений на 22-м TCP-порте сервера хоста, но администратор сервера может назначить любой порт исходя из соображений безопасности.

Для шифрования SSH использует специальные ключи, с помощью которых кодируются все передаваемые между клиентом и хостом данные.

* + 1. PuTTY

PuTTY - это свободно распространяемый клиент для различных протоколов удалённого доступа, включая SSH, Telnet, rlogin [14]. Также имеется возможность работы через последовательный порт.

PuTTY позволяет подключиться и управлять удаленным узлом (например, сервером). В программе реализована только клиентская сторона соединения — сторона отображения, в то время как сама работа выполняется на стороне сервера.

* + 1. Make, cmake

Makefile — это файл с инструкциями для утилиты make, которая нужна для автоматической сборки проекта [15]. Он одновременно является и инструкцией, и исполняемым файлом. Makefile содержит код, описывающий конечные условия — они должны выполняться при сборке проекта.

Makefile решает задачи выполнения цепочки команд для настройки среды разработки, автоматической сборки, запуска наборов тестов и развёртывания.

CMake — это кроссплатформенное программное средство автоматизации сборки программного обеспечения из исходного кода.

Оно не занимается непосредственно сборкой, а лишь генерирует файлы сборки из предварительно написанного файла сценария CMakeLists.txt и предоставляет простой единый интерфейс управления. Помимо этого, способно автоматизировать процесс установки и сборки пакетов.

Ключевой особенностью CMake является возможность размещать выходные данные компилятора (например, объектные файлы) вне дерева исходного кода. Это позволяет выполнять несколько вариаций сборок из одного и того же исходного кода, а также кросс компиляцию.

* + 1. Сеть

Wi-Fi — это технология беспроводной связи, которая создаёт беспроводную локальную сеть в ограниченной области: в пределах дома, офиса или другого пространства [16]. Его используют для беспроводного подключения устройств к интернету. Также с его помощью можно объединить устройства в локальную сеть для обмена данными и управления устройствами.

В основе Wi-Fi лежит набор стандартов IEEE 802.11. Они разрабатываются Институтом инженеров электротехники и электроники (IEEE).

Ethernet — это семейство протоколов и технологий, которые определяют способ организации передачи данных в локальных сетях (LAN). Он описывает правила, согласно которым устройства, подключённые к сети, обмениваются данными, а также задаёт формат кадров и методы доступа к среде передачи.

Ethernet передаёт данные следующим образом:

1. Устройство, которое хочет отправить данные, создаёт кадр данных, содержащий информацию для отправки.
2. Кадр данных отправляется на сетевой адаптер устройства, который преобразует его в электрический или оптический сигнал, подходящий для передачи по кабелю.
3. Сетевой адаптер отправляет сигнал по кабелю в коммутатор или маршрутизатор, который находится в той же локальной сети, что и устройство.
4. Коммутатор или маршрутизатор получает сигнал, проверяет адрес MAC получателя и направляет кадр данных к соответствующему устройству.
5. Устройство-получатель получает кадр данных, проверяет чек-сумму (FCS) и обрабатывает его. 1
6. Ethernet передаёт данные в виде кадров, которые содержат MAC-адреса отправителя и получателя.

Типы кабелей и среды передачи:

* Коаксиальный кабель (ранние версии) — использовался в старых сетях Ethernet.
* Витая пара (Cat5, Cat6, Cat7) — стандарт для современных LAN.
* Оптоволокно — используется в высокоскоростных сетях на больших расстояниях.

IPv4 — это четвёртая версия интернет-протокола IP, которая работает с 1982 года и является основным протоколом в Интернете. IPv4 обеспечивает возможность адресации примерно 4,3 млрд адресов.

IPv6 — это последняя версия интернет-протокола, которая была представлена в декабре 1995 года. IPv6 более продвинут, чем IPv4, и предоставляет лучшую функциональность. Новый протокол может предоставить практически бесконечное количество адресов для устройств и заменяет прошлую версию для обслуживания растущего числа трафика по всему миру и решения проблемы нехватки IP-адресов.

* + 1. Ip, netmask, gateway, inconfig

ip-адрес – это внутренние (серые) адреса (частные адреса), так как ограничены частными сетями [17]. Диапазон сетевых адресов 192.168.0.0 — это выделенный адрес частной сети, не поддерживающий маршрутизацию.

IP-адрес позволяет устройствам в сети общаться друг с другом, отправлять и получать данные. Версии IP-адресов могут быть IPv4 или IPv6.

У всех IP адресов есть две части сеть и узел.

Сеть – это та часть IP, которая не меняется во всей сети и все адреса устройств начинаются именно с номера сети.

Узел – это изменяющаяся часть IP. Каждое устройство имеет свой уникальный адрес в сети, он называется узлом.

Netmask (маска подсети) — это 32-битное число, указывающее, какая часть IP-адреса относится к сети, а какая — к устройствам (хостам), находящимся в этой сети. Она используется для разделения сети на меньшие отрезки или для определения диапазона адресов устройств внутри сети.

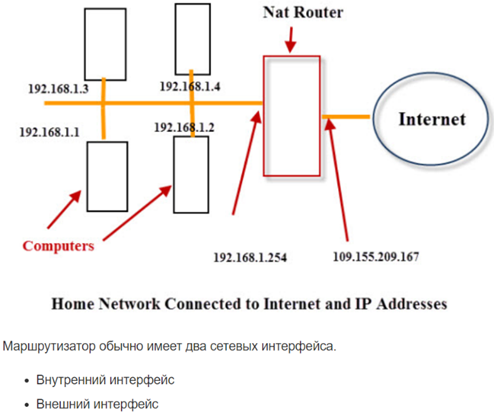


Рисунок 1.2 – Домашняя сеть, подключенная к интернету

Функции маски подсети:

1. Разделяет IP-адрес на две части: сетевой и хостовый адрес;

2. Определяет, сколько устройств может находиться в одной подсети;

3. Облегчает маршрутизацию пакетов данных.

Маска подсети не является частью IP-пакета. Она позволяет узнать, при отправке данных отправка информации будет происходить внутри сети или за её пределы.

Gateway (шлюз) — это специализированное устройство или программное обеспечение, которое позволяет взаимодействовать сетям, построенным на основе разных протоколов и технологий. Он преобразует данные из одного протокола или формата в другой.

Шлюз поддерживает ограниченный доступ к определенным приложениям, сканирует и фильтрует данные, помогает в сборе информации из других частей сети, участвует в диагностике и устранении неполадок.

Шлюзы могут быть аппаратными (например, маршрутизатор) или программными (устанавливаются на существующее оборудование).

Ipconfig — это утилита командной строки для управления сетевыми интерфейсами. Утилита позволяет определить, какие значения конфигурации были получены с помощью DHCP, APIPA или другой службы настройки IP-адресов или заданы администратором вручную.

В операционных системах Microsoft Windows ipconfig используется для отображения сведений о текущем подключении и управления клиентскими службами DHCP и DNS.

В операционных системах Linux и UNIX информация о подключении отслеживается несколькими утилитами, основной из которых является ifconfig.

* + 1. IP, UDP, TCP

IP-протокол: это протокол сетевого уровня [18]. Задача этого уровня – доставка IP-пакетов от отправителя к получателю. Кроме собственных данных, пакеты этого уровня имеют IP-адреса отправителя и получателя. Этот вид IP протокола является внешним.

Внешний (белый) IP-адрес или общедоступный IP-адрес — это IP-адрес интерфейса маршрутизатора, подключенного к Интернету. IP-адрес, назначенный внешнему интерфейсу, будет иметь маршрутизируемый IP-адрес и будет назначен вашим интернет-провайдером.

Подсеть – группа IP-адресов, объединенных в логическую единицу для упрощения управления сетью. Подсеть описывается маской подсети, которая указывает, какие биты в IP-адресе относятся к сети и хосту.

Любой цифровой IP-адрес можно связать буквенно-цифровым именем -> DNS (Domain Name Server).

TCP/UDP: протоколы транспортного уровня. Он находится над сетевым. На нем к пакету добавляются порты отправителя и получателя. Порт – номер приложения, работающего на этом компьютере.

TCP – протокол с установлением соединения с гарантированной доставкой пакетов. Если пакет не дошел, то он посылается повторно.

UDP – протокол без установленного соединения и с негарантированной доставкой пакетов.

Модель OSI представлена на рисунке 1.3.

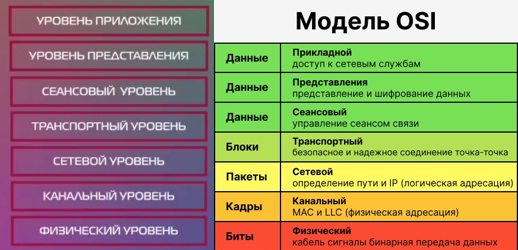


Рисунок 1.3 – Модель OSI

* + 1. UART

УАПП (универсальный асинхронный приемопередатчик) — узел вычислительных устройств, предназначенных для организации связи с другими цифровыми устройствами [19]. Преобразует передаваемые данные в последовательный вид, чтобы их можно было передать по одной физической цифровой линии другому аналогичному устройству. Распространен протокол COM-порт/RS232.

* + 1. Orange PI

Orange Pi — это серия одноплатных ARM-компьютеров, аналог Raspberry Pi [20].

В серию Orange Pi входят три устройства: Orange Pi Mini, Orange Pi, Orange Pi Plus. Модели различаются габаритами, весом и количеством разъёмов.

Платы Orange Pi предлагают стандартные интерфейсы, такие как HDMI, USB, Ethernet и контакты GPIO (ввода-вывода общего назначения). Компания также предоставляет собственную операционную систему Orange Pi OS, основанную на Debian Linux.

Платы Orange Pi популярны среди преподавателей и разработчиков, которые могут изучать электронику и программирование. Они могут работать под управлением различных операционных систем, таких как Android, дистрибутивов Linux и даже Windows в некоторых случаях.

* + 1. АТ – команды

AT-команды [21]:

AT — префикс всех команд и также пустой запрос настройки соединения;

A/ — модем повторяет предыдущую командную строку (команда используется без префикса);

ATA — инициация режима ответа (переход в DATA MODE). С задержкой 2.1с модем выдаст тональный сигнал ответа;

ATB(0/1) — ATA, но установка параметра скорости связи 1200 бод;

ATD — набор номера;

ATE(0/1) — команда эхо (выкл/вкл);

ATH(0/1/2) — команда ожидания;

ATI(0/1/2) — выдача идентификационного номера модема (241/231/OK)

ATL(0/1/2/3) — громкость звука;

ATM(0/1/2/3) — управление громкостью громкоговорителя;

ATN(0/1) — выбор скорости;

ATS0 — настроить число гудков перед автоматическим ответом на звонок

AT&V — показывает текущую конфигурацию

AT+GCAP — запрашивает полный перечень возможностей TA

AT+GMI — запрашивает идентификацию производителя

AT+GOI — запрос глобальной идентификации объектов

AT+GSN — возвращает TA серийный номер идентификации (IMEI)

AT+ICF — установка TE-TA контроля характера кадров

AT+IFC — установка TE-TA местного контроля потока данных

и тд

Рабочие

AT+CCLK? - время

AT+CMTE? - температура

AT - ok

ATI — SIM808 R14.18

ATE - \*echo

AT&V — большой список

AT+GCAP: +GCAP++ +CGSM

AT+GMI: SIMCOM\_Ltd

AT+GOI: SIM808

AT+GSN: 865067024840210

AT+GSV: некий системный перечень

AT+CGPSPWR(?/=0/=1) — проверка состояния включения GPS-модуля

//////Запуск GPS

AT+CGPSPWR? (статус)

AT+CGPSPWR=1 (вкл)

AT+CGPSSTATUS? (пока не поймает (если PPC моргает, то тайм синхронизировался со спутником))

AT+CGPSINF=0/8/32 где:

полный перечень <mode>, <longitude>, <latitude>, <altitude — высота>, <UTC time — yyyymmddHHMMSS>, <TTFF>, <num>, <speed>, <course>

AT+CGPSOUT=0/8/32 GSA RMC

AT+CGPSPWR=0 (выкл)

AT+COPS=? - GSM список операторов

* + 1. GSM

GSM — глобальный стандарт сотовой цифровой связи [22]. В нем предусмотрено разделение каналов как по частоте (FDMA), так и по времени (TDMA). Основные возможности:

* передача различных данных;
* передача речевой информации;
* обмен SMS;
* отправка факсимильных сообщений.

Сеть GSM делится на две системы:

* Switching System — коммутационная подсистема;
* Base Statiom System — система базовых станций. SS выполняет функции обслуживания вызовов и установления соединений, а также отвечает за реализацию всех назначенных абоненту услуг. BSS отвечает за функции, относящиеся к радиоинтерфейсу.
  + 1. NMEA

NMEA - протокол передачи данных, используемый для связи между собой морского навигационного оборудования (GPS-модули) [23]. Обеспечивает обмен информацией между различными навигационными устройствами, определяя формат и структуру сообщений для передачи данных. Позволяет объединить множество приборов в одну сеть и передавать информацию одновременно.

* + 1. WebSocket

WebSocket - независимый веб-протокол, который позволяет создавать интерактивное соединение между сервером и клиентом, и обмениваться сообщениями в реальном времени [24]. В отличии от HTTP, веб-сокеты позволяют работать с двунаправленным потоком данных, поэтому технология уникальна. Сервер не только может отвечать на запрос клиента, но и самостоятельно передавать новую информацию по мере поступления. Обмен данными происходит в рамках одного соединения в режиме реального времени.

* + 1. Lighttpd

Lighttpd (также «lighty», «лайти») — веб-сервер, разрабатываемый с расчётом на скорость и защищённость, а также соответствие стандартам [25]. Это свободное программное обеспечение, распространяемое по лицензии BSD.

Некоторые возможности: поддержка сжатия, отдаваемого содержимого «на лету», HTTP-аутентификации, перезаписи URL, SSL и автоматической балансировки нагрузки.

Зачастую lighttpd используется для отдачи статического содержимого, в то время как его генерацией занимается более сложный веб-сервер.

* + 1. Wstunnel

WStunnel — это протокол туннелирования WebSocket через TCP, который позволяет безопасно и эффективно передавать данные между соединениями WebSocket и TCP [26].

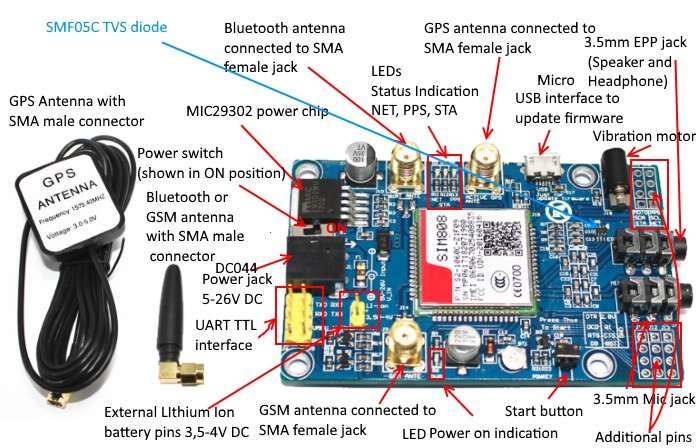
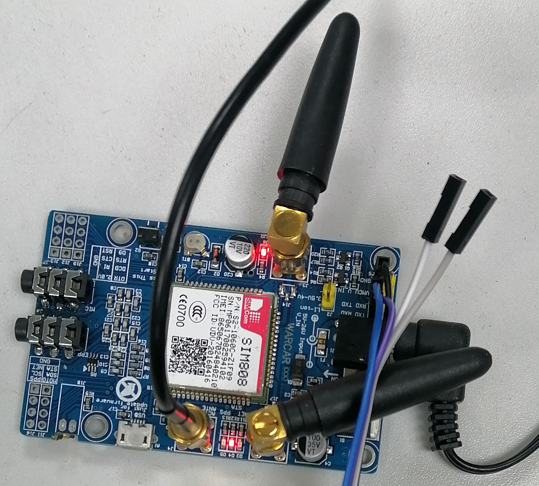
Он обеспечивает безопасный и надёжный канал коммуникации, обеспечивая шифрование трафика WebSocket и позволяя обходить политики сетевой безопасности.

Преимуществами протокола являются безопасность, совместимость, надёжность и эффективность.

* 1. Информация о используемом оборудовании
     1. SIM808

Основная информация о плате расширения «SIM808» [27]:

* Четырехдиапазонный 850/900/1800/1900 МГц;
* Многоканальная связь GPRS класса 12: макс. 85,6 кбит/с (при передаче данных);
* Мобильная станция GPRS класса B;
* Управляется с помощью AT-команд (3GPP TS 27.007, 27.005 и расширенные AT-команды SIMCOM);
* Поддерживает часы реального времени;
* Диапазон питающих напряжений 5V ~ 12V;
* Встроенный GPS/GNSS и поддерживает A-GPS;
* Поддерживает логический уровень от 3,0 В до 5,0 В;
* Низкое энергопотребление, 1 мА в спящем режиме;
* Поддерживает протокол GPS NMEA;
* Стандартная SIM-карта.

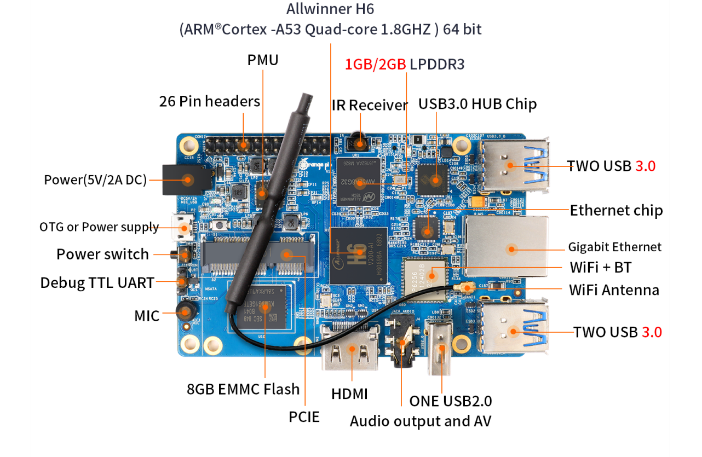
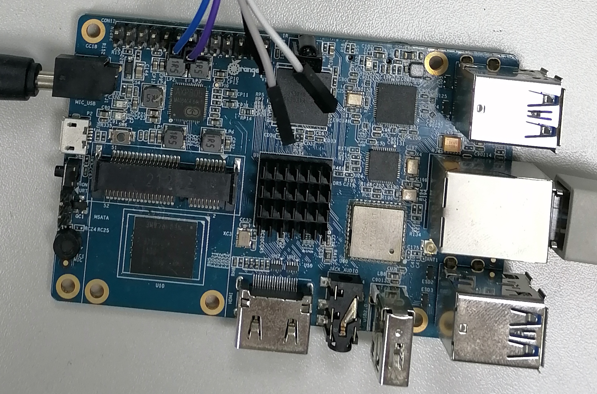
 

Рисунки 1.4 (а-б) – Изображение платы расширения «SIM808» с обозначением разъемов и ее реальный вид

* + 1. Orange Pi3

Основная информация о плате ARM «Orange Pi 3» [28]:

|  |  |
| --- | --- |
| Процессор | H6 Четырехъядерный 64-битный процессор ARM Cortex™-A53 с тактовой частотой 1,8 ГГц |
| Графический процессор | • Высокопроизводительный многоядерный графический процессор Mali T720  • OpenGL ES3.1/3.0/2.0/1.1  • Microsoft DirectX 11 FL9\_3  • ASTC (адаптивное масштабируемое сжатие текстур)  • Производительность с плавающей запятой более 70 GFLOPS |
| Память +  Встроенное Хранилище | Четыре типа:  • 1 ГБ LPDDR3 (совместно с графическим процессором)+EMMC (по умолчанию пусто) • 2 ГБ LPDDR3 (совместно с графическим процессором)+EMMC (по умолчанию пусто) • 1 ГБ LPDDR3 (совместно с графическим процессором)+8 ГБ флэш-памяти EMMC • 2 ГБ LPDDR3 (совместно с графическим процессором)+8 ГБ флэш-памяти EMMC |
| Wi-Fi + BT | AP6256, IEEE 802.11 a/b/g/n/ac, BT5.0 |
| Бортовая сеть | 10/100M/1000M, Ethernet RJ45 |
| Сетевой чип | RTL8211 |
| Аудиовход | МИКРОФОН |
| Аудиовыход | Разъем HDMI 2.0a и 3.5 мм AV |
| Видеовыход | HDMI 2.0a и CVBS |
| Декодирование видео | • Профиль H265/HEVC Main/Main10@Level5.2 высокого уровня; 4K@60 кадров в секунду, до 6Kx4K@30 кадров в секунду • Профиль H264/AVC BP/MP/HP@level5.1, MVC, 4K@30 кадров в секунду • Профиль VP9 0/2, 4K@30 кадров в секунду • Профиль AVS+/AVS JIZHUN@level 6.0, 1080P@60 кадров в секунду |
| PCIE | • Поддерживает режим RC • Поддерживает полосу x1 Gen2 (5,0 Гбит/с) • Соответствует спецификации PCI Express Base 2.0 |
| Источник питания | Вход постоянного тока, microUSB (OTG) |
| PMU | AXP805 |
| Порты USB 2.0 | 1\*USB 2.0 Host, 1\*USB OTG 2.0 |
| Порты USB 3.0 | 4 \* Хост USB 3.0 |
| Низкоуровневые периферийные устройства | 26 Контактный |
| Вывод GPIO (1x3) | UART, Наземный |
| IR | ДА |
| Клавиша | Мощность (SW4) |
| Поддерживаемая ОПЕРАЦИОННАЯ СИСТЕМА | Android7.0, Ubuntu, Debian |

а) б)

Рисунки 1.5 (а-б) – Изображение платы «Orange Pi3» из описания на сайте производителя и ее реальный вид

1. ПРАКТИЧЕСКАЯ ЧАСТЬ
   1. Физическое взаимодействие
      1. **Прямое подключение рабочего ПК к SIM808**

Проводим подключение по портам «RX» (прием) и «TX» (передача) + «GND» (земля).

Вводим в командную строку «sudo putty».

Провести подключение к порту, на котором подключено устройство. Для поиска порта закоротить выходы с ПК сами на себя и проверить прохождение сообщений (dmesg), найдя номер порта (ttyUSB1).

Послать базовые сообщения (AT, ATI).

* + 1. Подключение Orange Pi к ПК

Проверить подключение микропк к локальной сети (физически).

Открыть терминал и ввести параметры подключения (ssh root@172.16.141.175), ввести пароль.

* + 1. Подключение Orange Pi к SIM808

Открыть документацию по плате Orange Pi (конкретно по этой модели, так как расположение и количество портов изменяется). Найти, какие порты отвечают за «RX», «TX» и «GND» (ориентироваться по метке со стороны пайки контактов).

Подключить к плате «SIM808» через эти порты, с учетом маркировки портов.

* + 1. Отправка команд через Orange Pi на SIM808

После пункта 2.1.3 пишем в консоль:

microcom -s 9600 (скорость) /dev/ttyS2 (порт (узнать порт можно опять замкнув RX-TX)).

Послать базовые сообщения (AT, ATI) для проверки соединения.

Для отключения соединения ввести сочетание клавиш «ctrl+x».

* + 1. **Прямое подключение рабочего ПК к SIM808**

Финальный вид подключения всех элементов схематически изображен на рисунке 2.1:

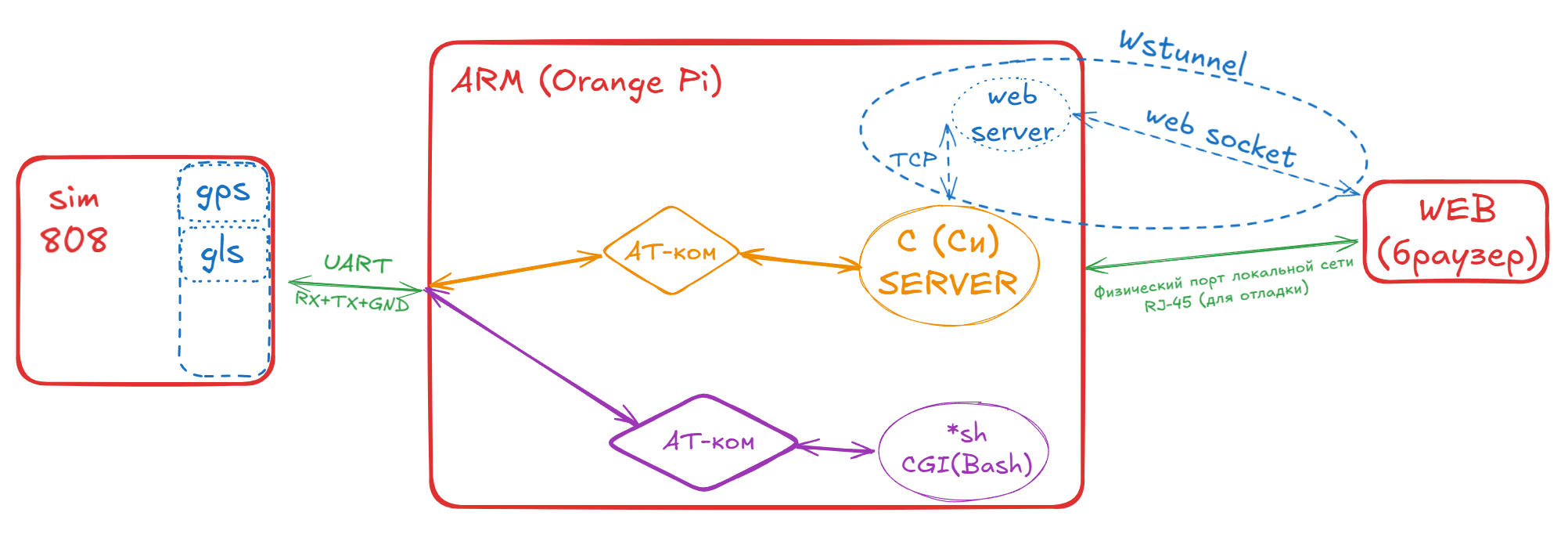


Рисунок 2.1 – Схематическое изображение соединения всей системы

Элементы схемы:

* Красным цветом изображены физические устройства;
* Пути зеленого цвета – физические подключения;
* Синим цветом показаны элементы, которые осуществляют систему передачи данных между WEB-страницей и сервером, расположенным на ARM (Orange Pi);
* Оранжевым цветом показано расположение сервера и его связь между устройствами;
* Желтым цветом указано, где происходит отправка AT-команд на «SIM808»;
* Фиолетовым цветом показана возможность разработчика непосредственно с платы ARM (Orange Pi) провести подключение к «SIM808», или написать скрипт для автоматизации выполнения некоторой команды (к примеру – сбор данных о температуре устройства);
* Голубым цветом изображены возможности платы «SIM808».
  1. Написание кода WEB-платформы

Для написания кода использовалась программа «Microsoft Visual Studio Code».

Изначально код создавался на компьютере под архитектуру «x86», но после первичной отладки был конвертирован под архитектуру «ARM» для дальнейшего размещения и работы, непосредственно, на «Orange Pi».

Стек проведен с использованием следующих элементов:

* HTML (от англ. Hyper Text Markup Language) — язык гипертекстовой разметки;
* CSS (от англ. Cascading Style Sheets) — каскадные таблицы стилей;
* JavaScript – интерпретируемый язык программирования, который используют для написания frontend- и backend-частей сайтов;
* C – компилируемый типизированный язык программирования общего назначения.
* Bash – (Bourne-Again Shell) командная оболочка для UNIX-подобных операционных систем. Она позволяет управлять ОС через текстовые команды, работать с файлами и папками, искать информацию и настраивать окружение. В данной работе использовалась для создания CGI-скриптов, а также для настройки «lighttpd».

Страница написана на языке гипертекстовой разметки «HTML». Для улучшения визуальной составляющей к странице подключается «CSS». Функционал страницы обеспечивается серверной составляющей, написанной на языках «C» и «JS» («JavaScript»). Обеспечение связи между WEB-страницей и сервером установлено посредством развертки «lighttpd», при помощи которого настроен протокол туннелирования «WebSocket» через «TCP».

В итоге была написана система отправки команд с WEB-страницы браузера на «SIM808». Сама же WEB-страница изображена ниже на рисунке 2.2:

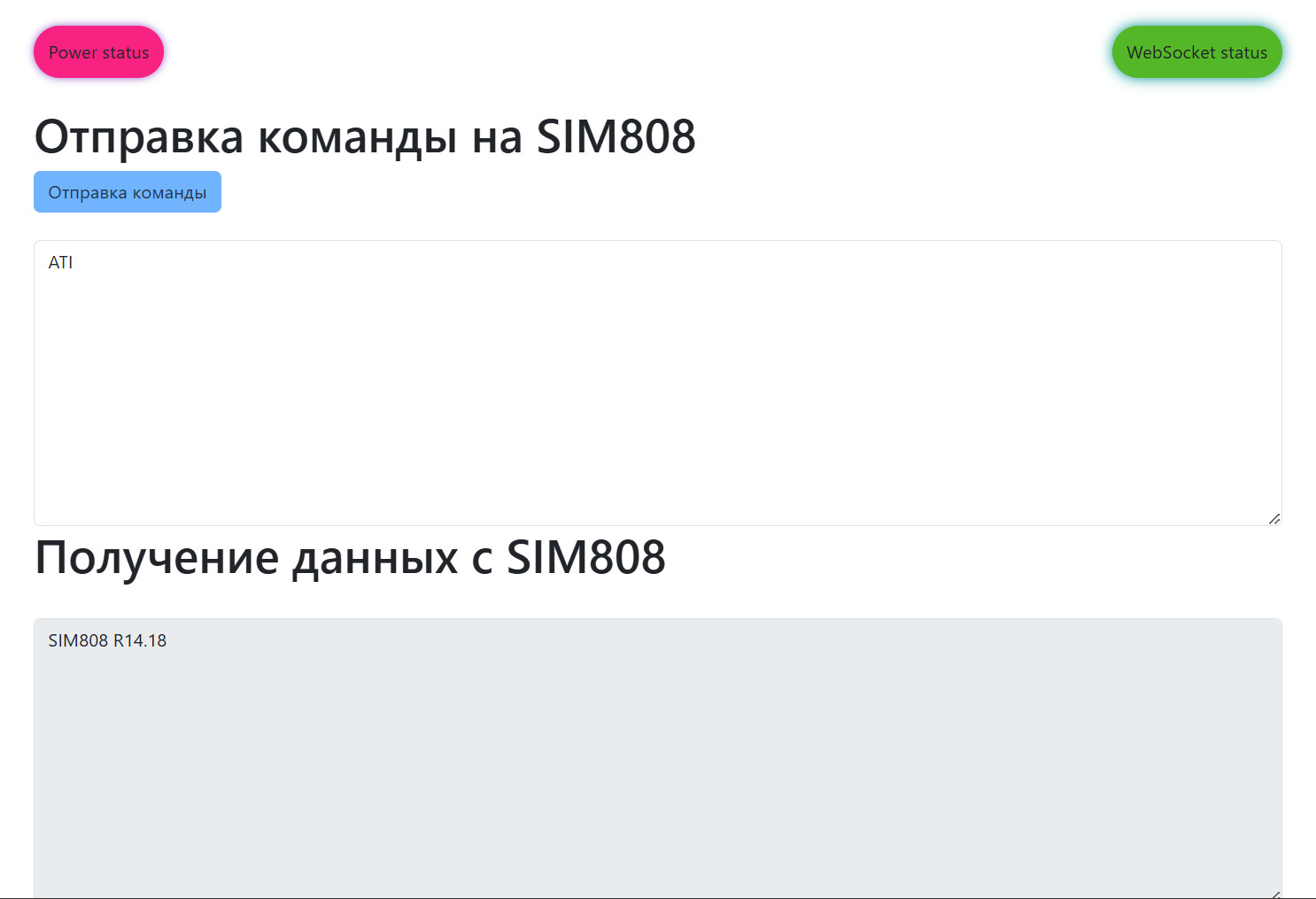


Рисунок 2.2 – Разработанная WEB-страница

Некоторые элементы кода представлены в Приложении А и Приложении Б.

Заключение

По итогу выполненной работы было изучено много теоретического материала, связанного с работой сети, а также с развертыванием своей локальной сети.

Изучив весь материал, была создана своя система, включающая в себя ПК (для отладки, или в качестве клиентского оборудования, на котором происходит подключение к WEB-странице) и платы Orange Pi, SIM808. Взаимодействие происходит благодаря написанному софту на языках C, JS, BASH, HTML, CSS.

Система позволяет отправлять команды на плату расширения «SIM808» через WEB-интерфейс. Механизм общения между платой расширения «SIM808» и WEB-интерфейсом разработан и расположен на плате ARM «Orange Pi 3».

В дальнейшем данную систему можно доработать для трассирования данных GPS с «SIM808» непосредственно на WEB-страницу.

**Список использованных источников**

1. HTML [Электронный ресурс]: сайт MDN Web Docs URL: <https://developer.mozilla.org/ru/docs/Web/HTML> (Дата обращения: 06.04.2025);
2. CSS [Электронный ресурс]: сайт HTML Academy URL: <https://htmlacademy.ru/blog/css> (Дата обращения: 07.04.2025);
3. JavaScript [Электронный ресурс]: сайт Learn JavaScript URL: <https://learn.javascript.ru/> (Дата обращения: 08.04.2025);
4. Bootstrap [Электронный ресурс]: оф. сайт Bootstrap URL: <https://getbootstrap.com/docs/5.3/getting-started/introduction/> (Дата обращения: 09.04.2025);
5. POST и GET в CGI [Электронный ресурс]: сайт GeeksforGeeks URL: <https://www.geeksforgeeks.org/get-and-post-methods-using-cgi/> (Дата обращения: 10.04.2025);
6. JSON, AJAX [Электронный ресурс]: сайт MDN Web Docs URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/JSON> (Дата обращения: 11.04.2025);
7. Linux [Электронный ресурс]: сайт Linux Foundation URL: <https://www.linuxfoundation.org/> (Дата обращения: 12.04.2025);
8. BASH и основные команды для работы [Электронный ресурс]: сайт Linuxize URL: <https://linuxize.com/post/bash-commands/> (Дата обращения: 13.04.2025);
9. Файловая система [Электронный ресурс]: сайт IBM Docs URL: <https://www.ibm.com/docs/ru/aix/7.2?topic=management-file-systems> (Дата обращения: 14.04.2025);
10. Редактор Vi [Электронный ресурс]: сайт Vim Docs URL: <https://www.vim.org/docs.php> (Дата обращения: 15.04.2025);
11. Git, Gitlab [Электронный ресурс]: оф. документация Git URL: <https://git-scm.com/doc> (Дата обращения: 16.04.2025);
12. Telnet [Электронный ресурс]: сайт GeeksforGeeks URL: <https://www.geeksforgeeks.org/telnet-command-in-linux/> (Дата обращения: 17.04.2025);
13. SSH [Электронный ресурс]: сайт SSH.com URL: <https://www.ssh.com/academy/ssh> (Дата обращения: 18.04.2025);
14. PuTTY [Электронный ресурс]: оф. сайт PuTTY URL: <https://www.putty.org/> (Дата обращения: 19.04.2025);
15. Make, cmake [Электронный ресурс]: сайт CMake Docs URL: <https://cmake.org/documentation/> (Дата обращения: 20.04.2025);
16. Сеть [Электронный ресурс]: сайт Cisco URL: <https://www.cisco.com/c/en/us/solutions/enterprise-networks/networking-basics.html> (Дата обращения: 21.04.2025);
17. IP, netmask, gateway, ifconfig [Электронный ресурс]: сайт Linux Handbook URL: <https://linuxhandbook.com/ifconfig/> (Дата обращения: 22.04.2025);
18. IP, UDP, TCP [Электронный ресурс]: сайт Cloudflare URL: <https://www.cloudflare.com/learning/ddos/glossary/user-datagram-protocol-udp/> (Дата обращения: 23.04.2025);
19. UART [Электронный ресурс]: сайт Embedded URL: <https://embedded.com/uart-serial-communication/> (Дата обращения: 24.04.2025);
20. Orange PI [Электронный ресурс]: оф. сайт Orange Pi URL: <http://www.orangepi.org/> (Дата обращения: 25.04.2025);
21. АТ–команды [Электронный ресурс]: сайт SIMCom Docs URL: <https://simcom.ee/documents/> (Дата обращения: 26.04.2025);
22. GSM [Электронный ресурс]: сайт GSMA URL: <https://www.gsma.com/aboutus/> (Дата обращения: 27.04.2025);
23. NMEA [Электронный ресурс]: сайт GPS World URL: <https://www.gpsworld.com/what-exactly-is-gps-nmea-data/> (Дата обращения: 28.04.2025);
24. WebSocket [Электронный ресурс]: сайт MDN Web Docs URL: <https://developer.mozilla.org/ru/docs/Web/API/WebSocket> (Дата обращения: 29.04.2025);
25. Lighttpd [Электронный ресурс]: оф. сайт Lighttpd URL: <https://www.lighttpd.net/> (Дата обращения: 30.04.2025);
26. Wstunnel [Электронный ресурс]: GitHub репозиторий URL: <https://github.com/erebe/wstunnel> (Дата обращения: 01.05.2025).
27. Плата расширения «SIM808» [Электронный ресурс]: 3drob URL: https://3drob.ru/arduino\_moduli/besprovodnye\_moduli/gsmgprs\_gps\_bluetooth\_modul\_sim808\_s\_antennami (Дата обращения: 01.05.2025).
28. Плата ARM «Orange Pi 3» [Электронный ресурс]: Orange Pi 3 URL: http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/details/Orange-Pi-3.html (Дата обращения: 01.05.2025).

Приложение А  
Демонстрация основных блоков WEB-страницы на языке «HTML»

<section id="listDir">

<h1>Отправка команды на SIM808</h1>

<div class="row">

<div class="col">

<button type="button" class="btn btn-primary" id="btnListDir">Отправка команды</button>

</div>

</div>

<br>

<div class="row">

<div class="col">

<textarea class="form-control" id="conRepDir" rows="10" disabled></textarea>

</div>

</div>

</section>

<section id="parseDate">

<h1>Получение данных с SIM808</h1>

<br>

<div class="row">

<div class="col">

<textarea class="form-control" id="conRepDate" rows="10" disabled></textarea>

</div>

</div>

</section>

Приложение Б  
Демонстрация некоторых изменений стиля страницы на языке «CSS»

html {

font-family: 'Proxima Nova', Arial;

font-size: 0.85rem;

/\* применение для всех 4 сторон \*/

}

main {

/\* (Отступ от хедера до мейна) \*/

padding: 20px;

}

/\* МОЙ БЛОК ИЗМЕНЕНИЙ \*/

header {

background-color: #333;

color: white;

padding: 10px 0;

text-align: center;

}

.led {

display: inline-block;

border-radius: 90px;

box-shadow: 0px 0px 6px #b115e6;

background-color: #f102dd;

padding: 0.7em 0.8em;

}

/\* Цветокор кнопок \*/

.btn.btn-primary {

--bs-btn-color: #21354d;

--bs-btn-bg: #70b3ff;

--bs-btn-border-color: #70b3ff;

--bs-btn-hover-color: #2b2f33;

--bs-btn-hover-bg: #5f97d8;

--bs-btn-hover-border-color: #5f97d8;

--bs-btn-focus-bg: #70b3ff;

--bs-btn-active-color: #2b2f33;

--bs-btn-active-bg: #6199da;

--bs-btn-active-border-color: #70b3ff;

}