

# Tbell – автоматизированная система подачи звонков

Фокин Иван Эдуардович,

Ученик МОУ Лицея №13 г. Петрозаводск

2024 год

# Оглавление

<b>1</b>	<b>Введение</b>	<b>3</b>
<b>2</b>	<b>Историческая справка и аналитика существующих решений</b>	<b>4</b>
<b>3</b>	<b>Проектирование системы подачи школьных звонков</b>	<b>7</b>
3.1	Аппаратная часть . . . . .	8
3.1.1	Устройство . . . . .	8
3.1.2	Корпус устройства . . . . .	8
3.2	Програмная часть . . . . .	10
3.2.1	Серверная ветвь . . . . .	10
3.2.2	Пользовательская ветвь . . . . .	12
3.2.3	Краткая инструкция по использованию . . . . .	12
3.2.4	Безопасность . . . . .	15
<b>4</b>	<b>Финансовая составляющая</b>	<b>16</b>
<b>5</b>	<b>Заключение</b>	<b>18</b>
5.1	Планы на развитие проекта . . . . .	18
5.2	Список литературы . . . . .	19
5.3	Ссылки проекта . . . . .	19
5.4	Инструментарий проектирования . . . . .	20

# Глава 1

## Введение

В большинстве школ сегодня звонки подают сотрудники вахты. О точности подачи звонков в этом случае говорить не приходится. Часто случаются ошибки. Более того, часто в школах есть два расписания звонков: для начальной и средней школы, что порождает ошибки. Также расписание звонков может меняться из-за праздников, мероприятий и по другим причинам.

На данный момент существуют автоматизированные системы подачи звонков, но они имеют высокую цену. Школы не спешат их приобретать, так как есть много других важных потребностей.

**Проблема:** ручная подача звонков в образовательных учреждениях, частые ошибки оператора звонков, дорогостоящие и неудобные автоматизированные системы подачи звонков, существующие на данный момент.

**Цель:** создание автоматизированной системы подачи звонков для учебных учреждений.

**Задачи:**

1. Спроектировать устройство для осуществления подачи звонков
2. Выбрать аппаратные решения
3. Провести электротехнические расчёты
4. Собрать готовую электрическую схему
5. Создать программное обеспечение для системы
6. Разместить программное обеспечение в общем доступе, защитив его открытой лицензией GNU public license 3.0

## Глава 2

# Историческая справка и аналитика существующих решений

Традиция звонить в колокольчик на урок появилась ещё в древности. По одной из версий историков, ее придумал древнегреческий философ и педагог Платон для того, чтобы собирать своих учеников на занятие. Сложно представить, но первый школьный звонок Платона был стеклянным. Он представлял собой два сосуда.

Способ обозначения уроков звонком, как и значительная часть концепции школьного образования, был придуман Яном Амосом Коменским. Первоначально такие звонки подавались учителем с помощью ручного колокольчика, небольшого настольного колокола или гонга.

В середине XIX века был изобретён электрический звонок, и в некоторых школах появились такие устройства. Изначально, как и в случае с колоколом, сигнал подавался вручную (замыканием цепи звонка).

В некоторых небольших школах всё ещё используются колокола. В наименее развитых странах в качестве звонка может использоваться гонг кустарного производства. Школа в Сиднее, 1872 год. В виде башни с колоколом. Школа в провинции Герат, Афганистан - качестве звонка используется гонг, сделанный из колёсной ступицы. Во многих небольших школах, особенно в сельской местности, в качестве школьного звонка используется колокол, размещённый на школьном дворе или в специальном сооружении на крыше школы. Звонки подаются вручную. Существуют электрические приводы, автоматически раскачивающие колокол по расписанию.

Электрический звонок является наиболее распространённым, типичная конструкция — с двумя «чашками». Официальное название — звонок громкого боя электрический двухча-

печный. В СССР с 1960-х годов школы массово оснащались звонками модели МЗ-1 и ее дальнейшими модификациями МЗ-2 и МЗМ-1, многие из которых продолжают работать и в настоящее время. Как правило, на каждом этаже школы под потолком размещается одно такое устройство, чтобы его звук был слышен во всех кабинетах. Продолжительность звонка составляет 5-10 сек. Во многих школах, помимо стандартного оповещения о начале/конце урока, с помощью такого устройства подаются дополнительные сигналы: три коротких звонка — на здание осуществлено вооружённое нападение, два коротких и один длинный звонок — пожарная тревога и эвакуация.

Распространённой и по сей день является подача сигнала нажатием кнопки: пока кнопка удерживается нажатой, раздаётся звон. Занимался этим дежурный учитель, вахтёр или секретарь. Недостаток такой системы — необходимость ответственного лица строго следить за временем.

В отсутствие компьютерной системы часофикации она может быть обеспечена механически — с помощью первичных и вторичных часов. Первичные часы — механическое устройство, механизм которого, помимо собственно часовой функции, способен замыкать и размыкать электроцепь по расписанию. Например, в советских часах «Стрела ЭВЧС-24» для этого используется 24-часовой диск с 288 отверстиями и дополнительный недельный диск с 7 отверстиями, позволяющий отключить работу сигналов в соответствующие дни. Штифт, вставленный в отверстие, в назначенное время замыкает цепь, нагревающую металлическую пластину, которая мгновенно расширяется, выгибается, касается контакта и замыкает вторичную цепь (при этом первичная цепь размыкается, пластина за 5-20 секунд остывает и приходит в исходное положение, что приводит к размыканию вторичной цепи). Вторичная цепь, в свою очередь, подаёт ток на звонки.

Наиболее современное решение — подача звонков через динамики, размещённые в каждом классе. Через эти же динамики можно подавать сигнал тревоги, объявления и т. д., причём необязательно во все сразу, а только в некоторые выбранные кабинеты. В случае установки такой системы возможна хакерская атака на школьную систему часофикации. Это возможно, если она подключена к компьютерной сети или если звонки поддерживают беспроводное управление. Вычислить хулигана в этом случае довольно трудно, требуются специалисты в области информационной безопасности (учитель информатики таковым не является).

В октябре 2010 в школе Maskie Academy Стонхейвена, Великобритания, школьные звонки были запрещены, поскольку они якобы взбудораживают учеников и делают их беспокойными. Кроме того, отмена звонков повышает личную ответственность и внимание школьников, учит их самостоятельно следить за временем. Впоследствии такое же решение было принято

ещё в нескольких школах Великобритании. В финских школах после реформы 2016 года отсутствуют уроки в привычном понимании этого слова. Соответственно, звонки были отменены, поскольку стали не нужны. Но в российских реалиях об отмене звонков в школах говорить не приходится.

На данный момент существуют несколько систем автоматизированной подачи звонков. Являются ли они полной альтернативой tbell — нет. Мой проект придерживается другой философии и пользовательского опыта, следовательно это просто другие устройства. Всё же рассмотрим некоторые из них.

Начну с "Автомата подачи школьных звонков МК-135". Недостатки: физический интерфейс управления, ограничение количества звонков в день.

Устройство "Звонок-М5". Недостатки: физический интерфейс управления, высокая цена.

Такой список можно продолжить довольно долго, но в этом уже не будет особого смысла. Исходя из выявленных недостатков других систем, предпочтений учеников, учителей и работников школ я приступил к разработке tbell.

## Глава 3

# Проектирование системы подачи ШКОЛЬНЫХ ЗВОНКОВ

Проектирование системы начнем с названия. Tbell – система автоматической подачи звонков для образовательных учреждений. Система состоит из программных и аппаратных решений. Все тексты программ, а также иные файлы защищены лицензией GNU public license, которая подразумевает свободное распространение, доработку и просто изучение проекта, одним словом open source.

Программная часть – программный код, управляющий устройством подачи звонков и конфигуратором расписания.

Аппаратная часть – физическое устройство, представленное в виде корпуса с электронными компонентами, осуществляющее подачу звонков по расписанию.<sup>1</sup>

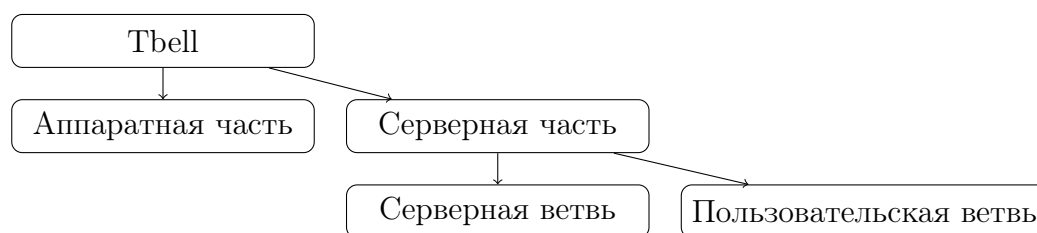


Рис. 3.1: Схема проекта

---

<sup>1</sup>определения авторские

## 3.1 Аппаратная часть

### 3.1.1 Устройство

Устройство по расписанию коммутирует цепь звонка, также есть кнопка для его ручной подачи. Расписание устройство получает при передаче в него строки-расписания. Устройство ведёт подсчёт времени посредством тактирования кристалла микроконтроллера, для увеличения точности раз в несколько часов устройство сверяется с точным временем NTP сервера. Это обеспечивает идеальную точность подачи звонка. Рассмотрим само устройство подробнее.

Компоненты, использованные в устройстве:

1. Микроконтроллер esp01s на базе чипа ESP8266EX. Он отвечает за все математические операции, подключение к беспроводной сети wi-fi, получение и хранение расписания и управление всеми другими компонентами.
2. Электромагнитное реле. Оно отвечает за коммутацию цепи звонка.
3. Разъём питания типа usb-c. Через него устройство получает питание.
4. Стабилизатор напряжения 3v3 ams1117-3v3. Он преобразует напряжение, полученное через usb (5v) в пригодные для микроконтроллера 3v3 (3.3v).
5. Другие компоненты (резисторы, конденсаторы, транзисторы, оптрон). Отвечают за стабильность работы устройства или являются "обвязкой".

Все компоненты размещены на компактной печатной плате рис. 3.3 . Печатная плата выполнена с использованием медного полигона для группы GND (условная земля), что уменьшает помехи, вызванные наведённым напряжением.

### 3.1.2 Корпус устройства

В соответствии с разработанными схемами электрических цепей и размерами плат, были разработаны 3Д модели и чертежи конструкции корпуса для электронных компонентов.

Корпус был спроектирован посредством САПР FreeCad. Образец напечатан на школьном 3Д-принтере с использованием пластика PLA.

Корпус состоит из короба и крышки, которая вставляется в паз короба. На крышку нанесено название проекта. В коробе предусмотрено отверстие для вывода проводов, разъёма питания и кнопки ручной подачи звонка.



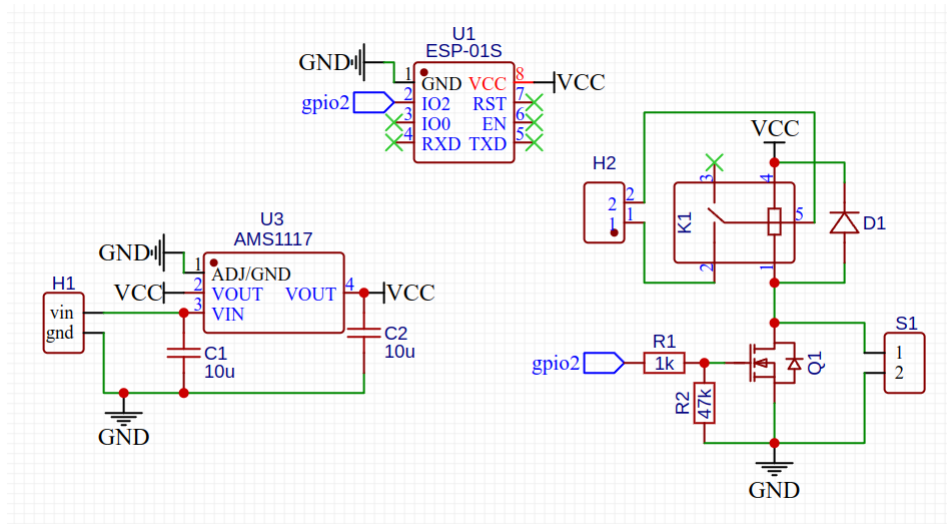


Рис. 3.2: Принципиальная схема

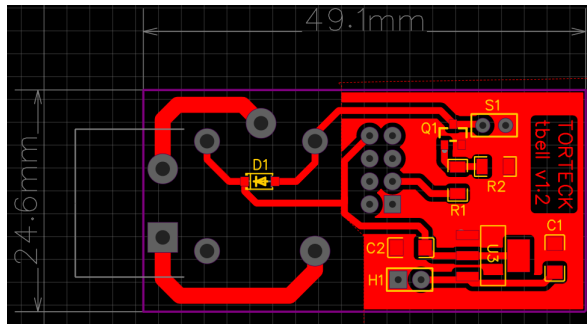


Рис. 3.3: Схема печатной платы

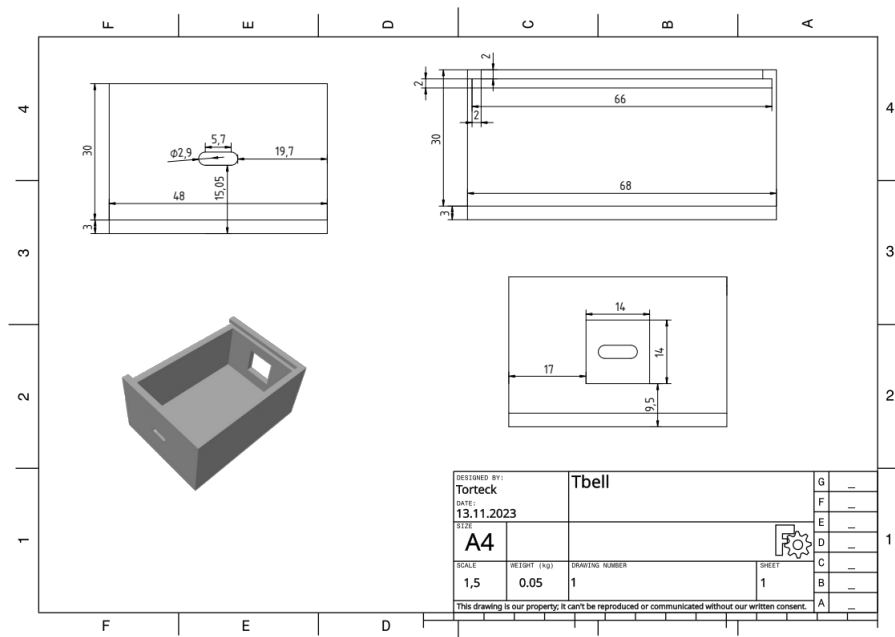


Рис. 3.4: Чертёж корпуса

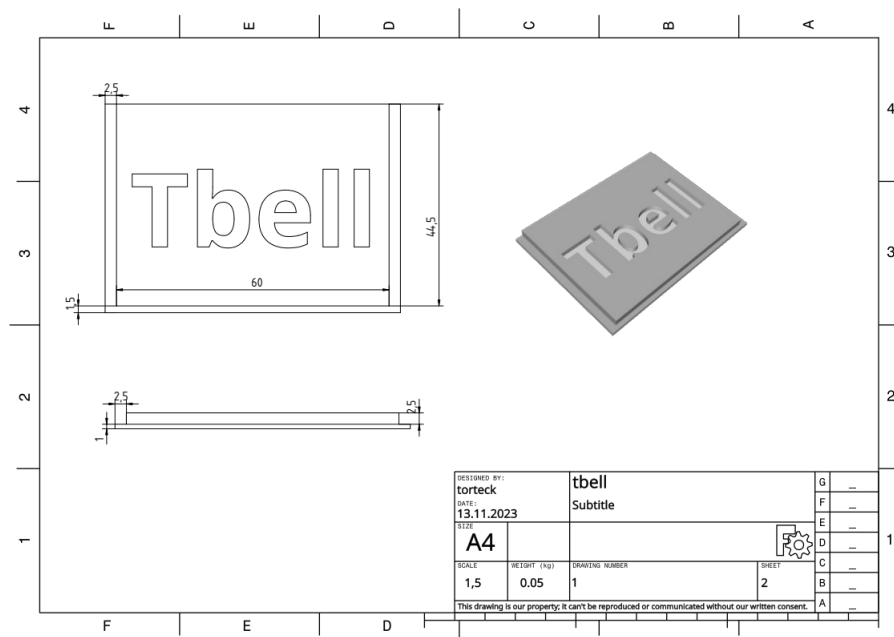


Рис. 3.5: Чертёж крышки

## 3.2 Программная часть

Программная часть, в свою очередь, подразделяется на две ветви. Такое разбиение даёт наибольшую отказоустойчивость, не мешая удобству пользователя.

### 3.2.1 Серверная ветвь

Серверная ветвь или конфигуратор — программное обеспечение для создания расписания. После создания расписания пользователь получит строку-расписание. Данная строка представляет собой JSON вида `{День_недели:[время_звонка1, время_звонка2], ...}`. Серверная ветвь реализована за счёт веб-страницы рис. 3.6. Она доступна по адресу <https://torteckyt.github.io/tbellcf/>. Но конфигуратор не использует вычислений на сервере, а значит, может быть скачан на устройство пользователя и работать автономно, без подключения к сети интернет. Например, в школе с нестабильным интернет-соединением удобно загрузить конфигуратор к себе на устройство и редактировать расписание автономно. Работа с конфигуратором подробно описана в инструкции.

Как пример кода из этой ветви приведу листинг 3.1. Функция принимает два параметра: уменьшаемое время `a` и вычитаемое время `b` в формате `"hh:mm"` или `"h:mm"`. Работу функции продемонстрирую на примере `time_minus("13:04", "0:07") == "12:57"`.

```

1 function time_minus(a, b) {
2     if(a.length == 4) {
3         var a_mins = Number(a.slice(2));
4         var a_hrs = Number(a.slice(0, 1));
5     }
6     else {
7         var a_mins = Number(a.slice(3));
8         var a_hrs = Number(a.slice(0, 2));
9     }
10
11     if(b.length == 4) {
12         var b_mins = Number(b.slice(2));
13         var b_hrs = Number(b.slice(0, 1));
14     }
15     else {
16         var b_mins = Number(b.slice(3));
17         var b_hrs = Number(b.slice(0, 2));
18     }
19
20     ans_mins = String(((a_hrs*60+a_mins)-(b_hrs*60+b_mins))%60);
21     ans_hrs = String(Math.floor(((a_hrs*60+a_mins)-(b_hrs*60+b_mins))/60));
22
23     if(ans_mins.length==1) {
24         var ans = ans_hrs + ":0" + ans_mins;
25     }
26     else {
27         var ans = ans_hrs + ":" + ans_mins;
28     }
29     return ans;
30 }

```

Листинг 3.1: Функция вычитания времени

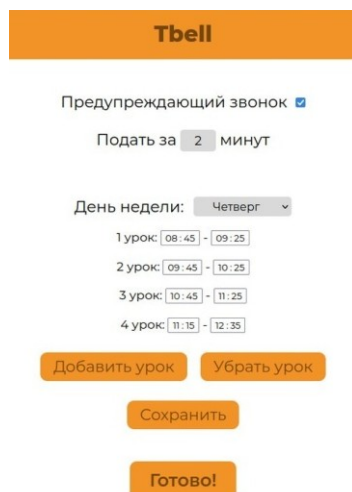


Рис. 3.6: Конфигуратор расписания

### 3.2.2 Пользовательская ветвь

Пользовательская ветвь — микропрограмма устройства подачи звонков, в том числе и интерфейс загрузки строки-расписания рис. 3.8. Представляет собой веб-страницу, размещённую на устройстве подачи звонков. Работа с пользовательским интерфейсом устройства также подробно описана в инструкции.

Устройство должно подключаться к сети wifi, значит должно знать название точки wifi (ssid) и пароль. Рассмотрим этот механизм на блок-схеме рис. 3.7.

Как пример кода из этой ветви приведу листинг 3.2. Данная функция обрабатывает страницу /change. При получении запроса сначала проверяется подключение к сети wifi, если его нет, пользователь будет перенаправлен на главную страницу, ведь расписание менять пока рано. Далее из запроса достаётся пароль редактирования, указанный пользователем и строка-расписание. Затем указанный пароль сверяется с верным, если всё хорошо, то новая строка-расписание записывается в энергонезависимую память eeprom. В конце пользователю показывается страница с сообщением об успешной смене расписания, устройство перезагружается.

### 3.2.3 Краткая инструкция по использованию

Перейдите по ссылке <https://tortekyt.github.io/tbellcf/>. Создайте расписание. Затем нажмите «Готово!», строка-расписание будет автоматически скопирована в ваш буфер обмена. Подключите ваше устройство подачи звонков к питанию и в разрыв цепи звонка. Включите wifi на своём устройстве. Подключитесь к сети «tbell» с паролем «tbell\_password».

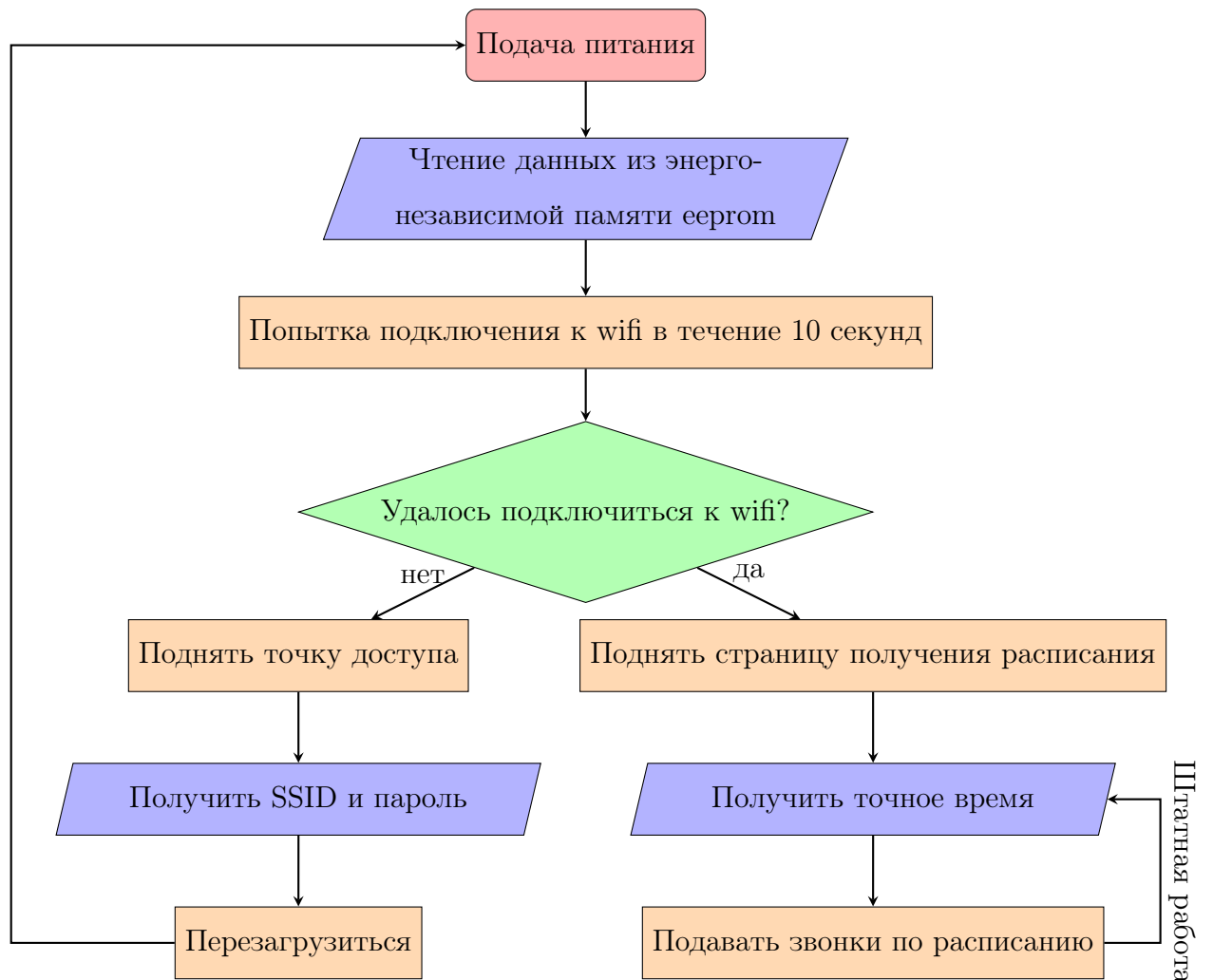


Рис. 3.7: Логическая схема подключения к wifi

Tbell

**Редактирование расписания**

Строка-расписание

Пароль

Сохранить

Рис. 3.8: Интерфейс загрузки расписания

```

1 server.on("/change", HTTP_GET, [] (AsyncWebServerRequest *request) {
2     if(WiFi.status() != WL_CONNECTED) {request->redirect("/");}
3     else {
4         String input_enc = request->getParam("enc")->value();
5         String input_password = request->getParam("password")->value();
6
7         Serial.println(input_enc);
8         Serial.println(input_password);
9
10        if(input_password == change_password) {
11            EepromStream eepromStream(enc_addr, 3072);
12            deserializeJson(enc_json, input_enc);
13            serializeJson(enc_json, eepromStream);
14            EEPROM.commit();
15
16            request->send_P(200, "text/html", success);
17            rbt = true;
18        }
19        else {
20            request->send_P(200, "text/html", unsuccess);
21        }
22    }
23 });

```

Листинг 3.2: Функция загрузки расписания

Перейдите по адресу `http://192.168.4.1:5000/`. Введите данные от основной сети wifi. Устройство автоматически перезагрузится и подключится wifi. Подключитесь к основной сети wifi. Перейдите `http://[tbell_ip]:5000/`. Вставьте строку-расписание, созданную в начале. Поздравляю, вы настроили tbell! Более подробно читайте в полной инструкции.

### 3.2.4 Безопасность

Здесь расскажу о том, как система защищена от несанкционированного проникновения из вне.

**Попытка получения пароля редактирования** Это, пожалуй, самый очевидный фронт атаки. Перебор пароля является плохой идеей: так как в реальном устройстве стандартный пароль будет изменён на случайный, длиной не менее 20 символов и содержащий буквенные символы, цифры и спецсимволы. По лучшим оценкам такой пароль придётся перебирать сотни тысяч лет. Вычислить такого хулигана не составит труда. Именно здесь идеально подойдёт фраза: "Да я тебя по ip вычислю!". *Важная ремарка:* никакой пароль не защитит если его будут хранить в незащищенном месте и предавать как и кому попало.

**Попытка получения прошивки устройства** Если наш хулиган сможет получить доступ к устройству, а затем выкачать из него дамп прошивки, то он ничего не получит. Прошивка храниться в двоичном виде, восстановление из него обычного кода невозможно. Этот план был изначально обречён на провал.

Об очевидно варварских методах атаки рассказывать нет смысла, ведь человек всегда сможет доказать превосходство ньютонов над интеллектом.

## Глава 4

### Финансовая составляющая

Хотя сейчас экономическую ситуацию, а, следовательно, и цены на компоненты тяжело назвать стабильными, попытаюсь дать смету на изготовление одного устройства. На данный момент времени заказ печатных плат на производстве по адекватным ценам невозможен в связи с отказом китайских компаний от сотрудничества.

Наименование	Цена (руб.)
Микроконтроллер	90
Реле	35
Оптрон	5
Разъём usb type-c	20
Резисторы и конденсаторы	1
Стабилизатор напряжения	4
Кнопка	13
Печатная плата	100
Пластик PLA	20
<b>Итого:</b>	<b>288</b>

Таблица 4.1: Смета затрат на материалы

В этих расчётах изготовление печатной платы предполагается методом ЛУТ (лазерно-утюжная технология) или методом фоторезист. Несколько лет назад за немного меньшие деньги можно было заказать изготовление такой платы на производстве. Цены компонентов взяты в розницу, при заказе оптом выйдет значительно дешевле. Также не учтены затраты на сборку (время сборки одного устройства 30 минут). В среднем работу специалиста можно оценить из расчета 500 рублей в час. Значит работа сборщика системы будет стоить 250 руб-



лей . В общей сложности, без учета амортизации, электричества и других мелких расходов, себестоимость продукции составит около 600 рублей.

За счёт выбранных инженерных решений проект получился экономически целесообразным. Даже со стоимостью в 3000 рублей это будет дешевле, чем аналогичные системы.

Ещё раз упомяну, что энтузиасты смогут собрать свой tbell самостоятельно, ведь проект является открытым.

# Глава 5

## Заключение

В ходе данного проекта я создал систему автоматизированной подачи звонков для образовательных учреждений. Данный проект решает проблему ручной подачи звонков.

Особым преимуществом созданной системы является его цена, простота использования и компактность. Поэтому проект можно считать успешным, задачи выполнены а цели достигнутыми.

### 5.1 Планы на развитие проекта

Тестирование системы в реальных условиях даст более точную картину достоинств и недостатков разработки. Сейчас ведутся переговоры по поводу установки устройства в одну из школ.

Хотя tbell и реализован, финальную точку ставить ещё рано, ведь во время воплощения данного проекта появилось много новых идей, которые рано или поздно будут реализованы и войдут в состав tbell.

Удобным инструментом станет интеграция с электронным дневником: расписание звонков из него можно будет получить, нажав несколько кнопок. Это, увы, не так просто с технической стороны. Нужно или договариваться о создании api на стороне дневника, или заниматься парсингом, что не очень надёжно.

Планируется разделить проект на две группы: классическую и расширенную. Классическая группа - это устройство, встраиваемое в существующую систему подачи звонков. Фактически, классическая группа уже реализована. Расширенная группа подразумевает другой подход. Старые звонки будут заменены на новые, имеющие динамик и беспроводной интерфейс. Такое решение позволит выбрать мелодию для звонка, а так же отключать определённые звонки, если в этой части школы, например, проходит экзамен. Через новые звонки

можно будет передавать оповещения, в том числе и пожарные.

## 5.2 Список литературы

1. Документация Arduino.h - URL: <https://www.arduino.cc/reference/en/> (обращение 05.06.2024)
2. Документация PlatformIO - URL <https://docs.platformio.org/en/latest/> (обращение 05.06.2024)
3. Документация JS от MDN - URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript/> (обращение 05.06.2024)
4. Документация HTML от MDM - URL: <https://developer.mozilla.org/ru/docs/Web/HTML/> (обращение 05.06.2024)
5. Документация CSS от MDM - URL: <https://developer.mozilla.org/ru/docs/Web/CSS/> (обращение 05.06.2024)
6. Документация FreeCad - URL: [https://wiki.freecad.org/Main\\_Page/](https://wiki.freecad.org/Main_Page/) (обращение 05.06.2024)
7. Документация EasyEDA - URL: <https://docs.easyeda.com/en/FAQ/Editor/index.html/> (обращение 05.06.2024)
8. Википедия школьный звонок - URL: <https://dub.sh/8I0sTe5/> (обращение 05.06.2024)
9. "Автомат подачи школьных звонков МК-135 URL: <https://250v.ru/goods/Schetchik-impulsov-CT6Y-5> (обращение 05.06.2024)
10. "Звонок-М5 URL: <https://www.zvonok5m.ru/> (обращение 05.06.2024)

## 5.3 Ссылки проекта

1. Репозиторий проекта, его описание и инструкции - URL: <https://github.com/TorteckYT/tbell> (обращение 05.06.2024)
2. Репозиторий конфигулятора - URL: <https://github.com/TorteckYT/tbellcf> (обращение 05.06.2024)

3. Сайт конфигуратора - URL: <https://torteckyt.github.io/tbellcf/> (обращение 05.06.2024)

## 5.4 Инструментарий проектирования

1. PlatformIO - среда разработки программы для микроконтроллера
2. Vim - редактор кода, IDE (интегрированная среда разработки от англ. integrated development environment)
3. EasyEDA - средство создания печатных плат, САПР (система автоматизированного проектирования)
4. FreeCad - среда 3Д моделирования, САПР (система автоматизированного проектирования)
5. Л<sup>A</sup>T<sub>E</sub>X- текстовый процессор для создания текстовых документов, в том числе этого