

Tbell – система автоматической подачи звонков

Фокин Иван

2024 г.

Введение

Историческая справка

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 сообщающихся сосуда.

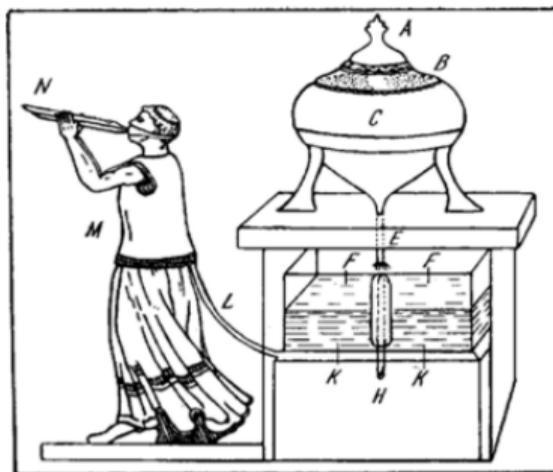


Рис. 1: Звонок Платона

В большинстве школ сегодня звонки подают сотрудники вахты. О точности подачи звонков в этом случае говорить не приходится. Часто случаются ошибки. Более того, во многих школах, в том числе и в нашей, два расписания звонков: для начальной и средней школы. Также расписание звонков может меняться (праздники, мероприятия и другие причины). На данный момент существуют автоматизированные системы подачи звонков, но они имеют высокую цену, а также неудобный интерфейс управления (в некоторых случаях физический). Школы не спешат их приобретать, у них есть много других важных потребностей.

Проблема: ручная подача звонков в образовательных учреждениях, частые ошибки оператора звонков, дорогостоящие автоматизированные системы подачи звонков.

Цель: создание автоматизированной системы подачи звонков для учебных учреждений.

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

Tbell – система автоматической подачи звонков для образовательных учреждений. Проект включает в себя как программные, так и аппаратные решения, которые будут рассмотрены далее. Все тексты программ, а так же иные файлы защищены лицензией GNU public license, которая подразумевает свободное распространение, доработку и просто изучение проекта, одним словом (двумя) open source.



Рис. 2: Логотип проекта GNU

Структура проекта

Tbell можно разделить на эти части.

- Аппаратная часть
- Программная часть

Рассмотрим эти части подробнее

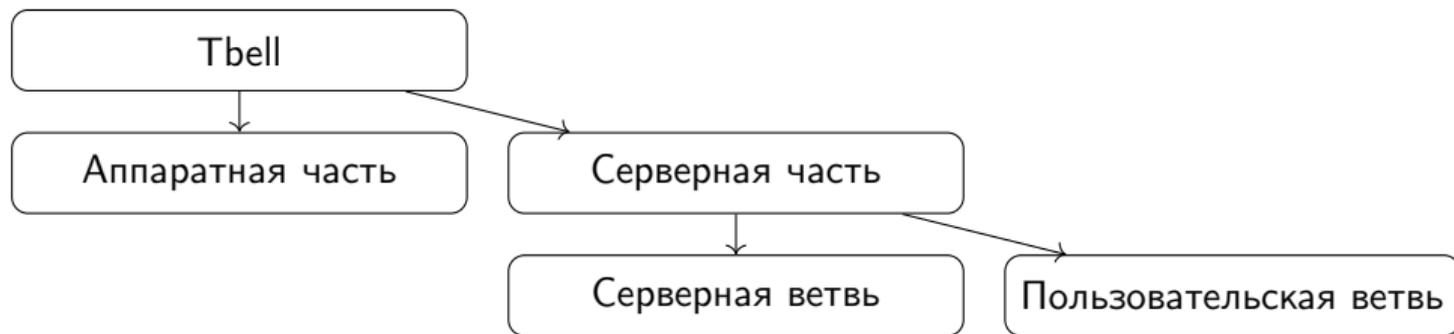


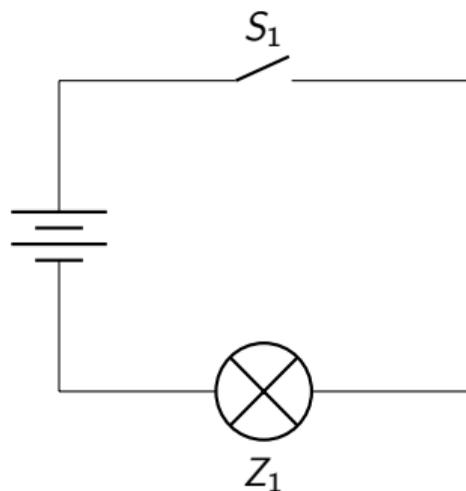
Рис. 3: Схема проекта

Аппаратная часть

Аппаратная часть

Представляет собой устройство подачи звонков. Устройство имеет разъём питания, кнопку для ручной подачи звонка и клеммник для подключения к цепи звонка.

Устройство устанавливается в разрыв цепи звонка.



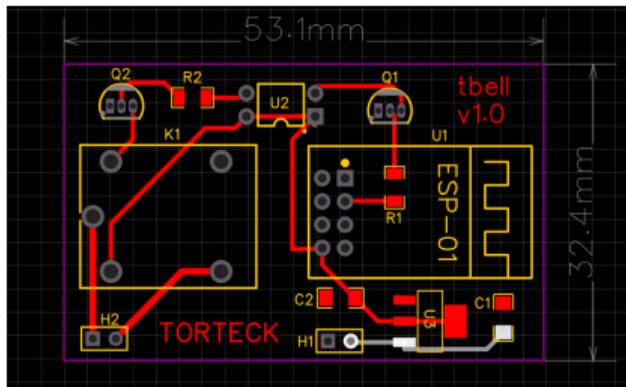
S_1 - Устройство подачи звонков
 Z_1 - Звонок (несколько звонков, соединённых параллельно)

Рис. 4: Схема подключения устройства подачи звонков

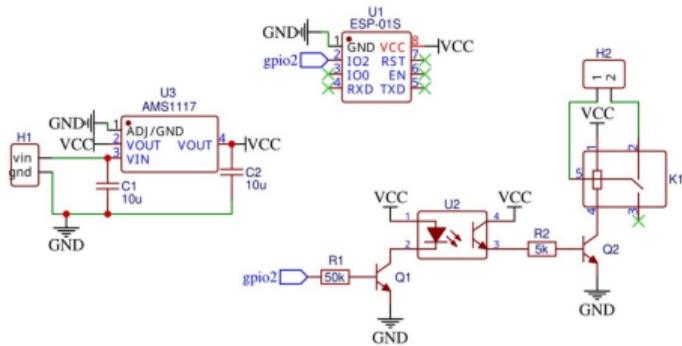
- Микроконтроллер esp01s на базе чипа ESP8266EX
- Электромагнитное реле
- Стабилизатор напряжения 3v3 ams1117-3v3
- Разъём питания типа usb-c

Печатная плата

Все компоненты размещены на небольшой печатной плате



(a) Схема печатной платы



(b) Принципиальная схема

Рис. 5: Схемы устройства

Принцип действия

Немного подробнее о схемотехнике. В момент подачи звонка микроконтроллер подаёт высокий уровень на контакт GPIO2 (пин ввода-вывода 2). Далее электрический ток протекает по цепи база-эмиттер при транзистора Q1, чем "открывает" оптрон U2. Оптран, в свою очередь позволяет протечь току по цепи база-эмиттер при транзистора Q2, чем замыкает контакт реле.

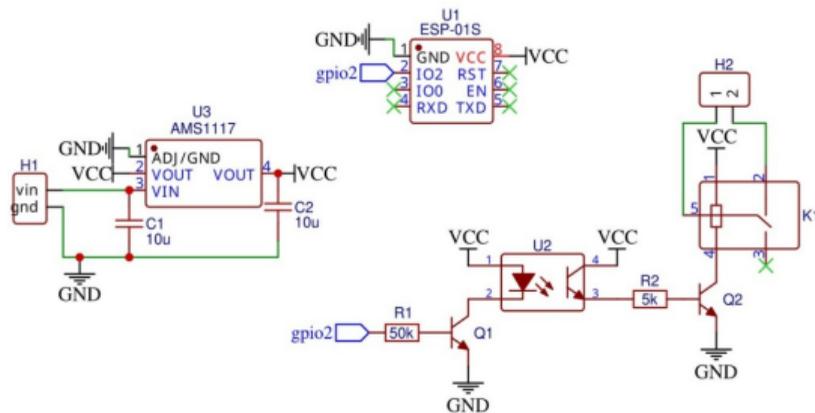
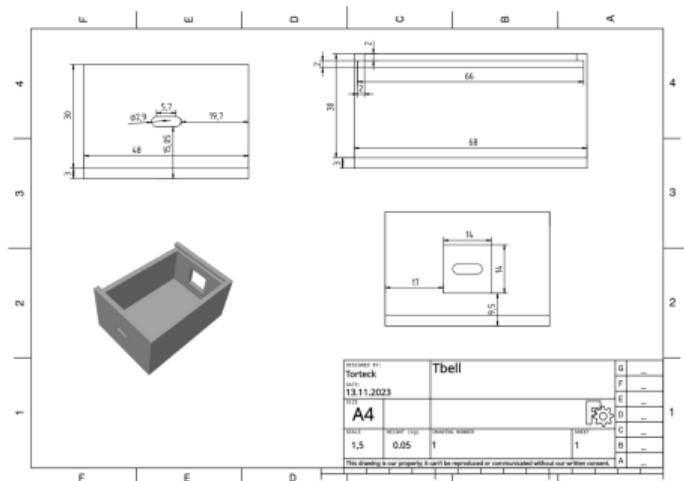
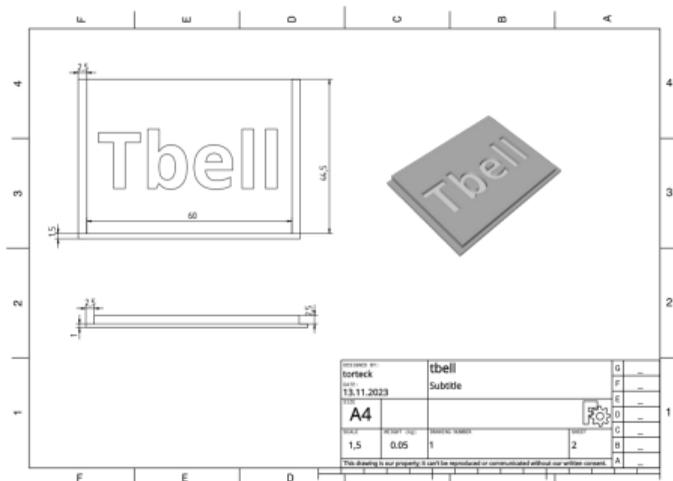


Рис. 6: принципиальная схема

Для устройства был разработан корпус, изготавливаемый на 3Д принтере.



(a) Корпус



(b) Крышка

Рис. 7: Чертежи корпуса

Программная часть

Программная часть состоит из двух дополняющих друг друга ветвей, такое разбиение даёт наибольшую отказоустойчивость, при этом не мешая удобному управлению.

- 1 Серверная ветвь (конфигуратор расписания)
- 2 Пользовательская ветвь (включение составленного расписания в работу)

Теперь рассмотрим каждую ветвь детальнее.

Серверная ветвь

Представляет собой веб-страницу, которая полностью работает на устройстве пользователя, т.е. работа с сервисом возможна даже в офлайн режиме.

Tbell

Предупреждающий звонок

Подать за минут

День недели:

1 урок -

2 урок -

3 урок -

4 урок -

Рис. 8: Конфигуратор расписания

Для создания серверной ветви были использованы следующие технологии:

- HTML 5
- CSS 3
- JavaScript



(a) HTML 5



(b) CSS 3



(c) JavaScript

Рис. 9: Логотипы технологий

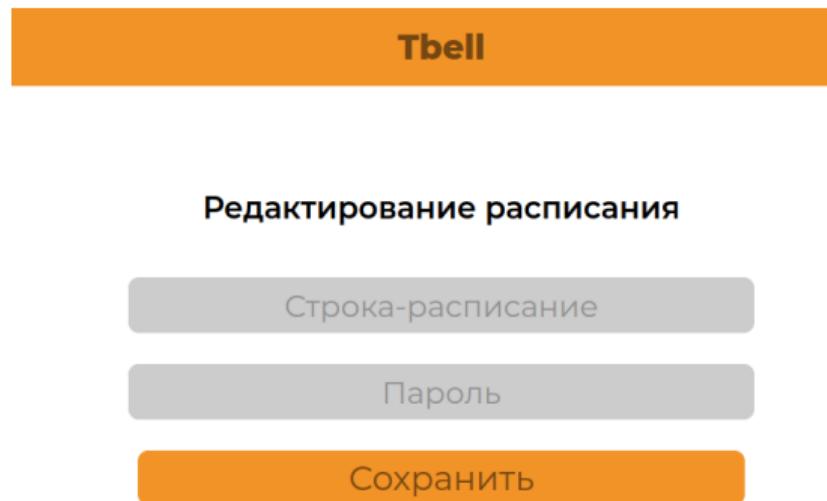
```
1 function time_minus(a, b) {  
2     if(a.length == 4) {var a_mins = Number(a.slice(2)); var a_hrs = Number(a.slice(0, 1));}  
3     else {var a_mins = Number(a.slice(3)); var a_hrs = Number(a.slice(0, 2));}  
4     if(b.length == 4) {var b_mins = Number(b.slice(2)); var b_hrs = Number(b.slice(0, 1));}  
5     else {var b_mins = Number(b.slice(3)); var b_hrs = Number(b.slice(0, 2));}  
6  
7     ans_mins = String(((a_hrs*60+a_mins)-(b_hrs*60+b_mins))%60);  
8     ans_hrs = String(Math.floor(((a_hrs*60+a_mins)-(b_hrs*60+b_mins))/60));  
9     if(ans_mins.length==1) {var ans = ans_hrs + ":0" + ans_mins;}  
10    else {var ans = ans_hrs + ":" + ans_mins;}  
11  
12    return ans;  
13 }
```

Листинг 1: Функция вычитания времени

```
time_minus("13:04", "0:07") == "12:57"
```

Пользовательская ветвь

Представляет собой прошивку для устройства подачи звонков. Для пользователя предусмотрена веб-страница, которая полностью работает на устройстве подачи звонков, работа с ней возможна без доступа к сети интернет.



The image shows a user interface for editing a schedule. At the top, there is an orange header bar with the text "Tbell". Below this, the title "Редактирование расписания" (Editing schedule) is centered. Underneath the title, there are three input fields: "Строка-расписание" (Schedule line), "Пароль" (Password), and "Сохранить" (Save). The "Сохранить" button is highlighted in orange, while the other two fields are in a light gray color.

Рис. 10: Интерфейс загрузки расписания

Для создания пользовательской ветви были использованы следующие технологии:

- C++
- Arduino.h



(a) C++



(b) Arduino.h

Рис. 11: Логотипы технологий

```
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         Serial.println(input_enc);
7         Serial.println(input_password);
8
9         if(input_password == change_password) {
10            EepromStream eepromStream(enc_addr, 3072);
11            deserializeJson(enc_json, input_enc);
12            serializeJson(enc_json, eepromStream);
13            EEPROM.commit();
14
15            request->send_P(200, "text/html", success);
16            rbt = true;
17        }
18        else {
19            request->send_P(200, "text/html", unsuccess);
20        }
21    }
22 });
```

Листинг 2: Функция загрузки расписания

Заключение

Финансовая составляющая

Проект является открытым, как упоминалось ранее, и каждый в праве собрать своё устройство, что является несомненным плюсом т.к. каждый пользователь вносит свой вклад в проект. Вы всегда можете обсудить или предложить изменение каких-либо решений. Ссылки указаны в официальной репозитории проекта.

Планируемая цена за готовое устройство менее 1000 рублей.

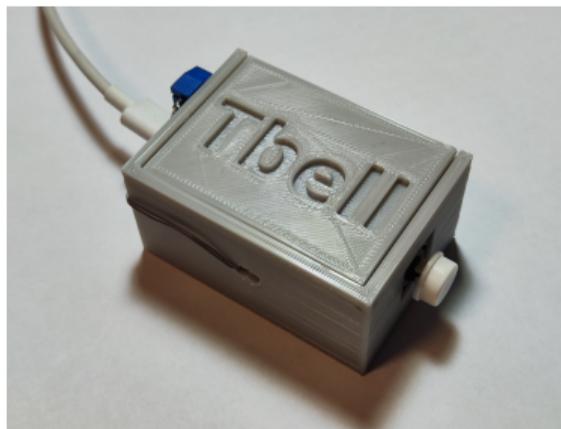


Рис. 12: Прототип устройства

- PlatformIO для vim – написание микропрограммы для устройства
- Vim – написание всего кода, включая серверную и пользовательскую ветви, микропрограмму, а так же текстовых документов и данной презентации
- EasyEDA – разработка печатной платы, принципиальной схемы
- L^AT_EX - текстовый процессор для создания текстовых документов
- Beamer - пакет для L^AT_EX, движок этой презентации

Спасибо за внимание!

Остались вопросы?

- Задайте!
- iivanofoks@gmail.com
- Телеграм: @torteck
- <https://github.com/TorteckYT/tbell>