

ОТДЕЛ ПО ОБРАЗОВАНИЮ ЛЯХОВИЧСКОГО РАЙИСПОЛКОМА
ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«МАЛОГОРОДИЩЕНСКАЯ СРЕДНЯЯ ШКОЛА ЛЯХОВИЧСКОГО РАЙОНА»

**ИССЛЕДОВАНИЕ МЕТОДОВ И СОЗДАНИЕ УСТРОЙСТВА
АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ ШКОЛЬНЫМ ЗВОНКОМ**

Научно-исследовательская работа

Секция «Информатика»

Выполнил
Сергей Максим Валерьевич,
учащийся 11 класса

Руководитель
Косабуцкий Александр
Николаевич,
учитель физики, информатики

Малое Городище, 2022

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
ГЛАВА 1. ВАРИАНТЫ УПРАВЛЕНИЯ ВКЛЮЧЕНИЕМ-ВЫКЛЮЧЕНИЕМ УСТРОЙСТВ ПО ВРЕМЕНИ	5
1.1. Механические реле времени	5
1.2. Электронные реле времени	5
1.3. Реле времени на микроконтроллерах	5
ГЛАВА 2. ВАРИАНТЫ УПРАВЛЕНИЯ ВКЛЮЧЕНИЕМ-ВЫКЛЮЧЕНИЕМ УСТРОЙСТВ С ПОМОЩЬЮ ARDUINO И NODEMCU	6
2.1. Управление временем с помощью команды Delay	6
2.2. Управление временем с помощью часов для Arduino	6
2.3. Управление временем с помощью NTP сервера точного времени	6
ГЛАВА 3. УСТРОЙСТВО И ПРИНЦИП ДЕЙСТВИЯ ЭЛЕКТРОМАГНИТНОГО РЕЛЕ	8
ГЛАВА 4. ОПИСАНИЕ ДЕМОНСТРАЦИОННЫХ УСТРОЙСТВ С РАЗНЫМИ СПОСОБАМИ РЕГУЛИРОВКИ ВКЛЮЧЕНИЯ-ВЫКЛЮЧЕНИЯ ПО ВРЕМЕНИ	9
4.1. Принципиальная схема и программа для устройства на основе Arduino UNO и электромагнитного реле	9
4.2. Принципиальная схема и программа для устройства на основе Arduino UNO, электромагнитного реле и часов RTCDS1307	11
4.3. Принципиальная схема и программа для устройства на основе NodeMCU и электромагнитного	13
ЗАКЛЮЧЕНИЕ	16
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	17
ПРИЛОЖЕНИЕ	18

ВВЕДЕНИЕ

Всем учащимся, наверное, известно чувство волнения и раздражения, когда урок по времени уже закончился, а звонок еще не прозвенел. Хорошо, если эта разница находится в пределах нескольких секунд, но иногда она доходит и до нескольких минут.

Мы решили выяснить, почему звонок звенит не в соответствии со временем окончания урока. Первая причина, конечно, в том, что работник школы позвонил несвоевременно. Вторая, очень частая причина, часы в учреждении могут идти не точно (опаздывать или спешить).

Мы решили выяснить, насколько это трудоемкий процесс – своевременно звонить с урока. Мы приняли решение: в определенный день заменить работника, который подает звонки с уроков. Конечно, если сидеть и смотреть на секундную стрелку часов не отвлекаясь, то позвонить своевременно с урока очень просто. А если заниматься каким-то делом и следить за временем одновременно, то очень сложно позвонить вовремя с урока, работа занимает и увлекает, работнику легко отвлечься и подать звонок не вовремя. В тот день, когда мы были ответственными, последние 15 минут урока я практически не слышал, что говорил учитель, все время думал о том, как позвонить с урока вовремя и следил за временем. Если я на уроке внимательно слушал учителя, то невнимательно следил за временем и звонил с опозданием. Что касается неточности хода часов на вахте, то это неточность очень часто достигает 1-2 минут. Для того, чтобы часы не спешили и не опаздывали, за этим необходимо постоянно следить, подстраивать, менять батарейки и т.д.

Принимая во внимание вышеизложенные рассуждения, можно сделать вывод: чтобы звонок с урока звенел своевременно, нужно затратить дополнительное время и усилия работников школы. Звонить с уроков точно, не опаздывая ни на секунду, в таких условиях вообще невозможно.

Мы задались вопросом: возможно ли, применив современные информационные технологии, автоматизировать процесс срабатывания звонка и добиться абсолютной точности в его работе.

Мы выдвинули гипотезу, что с помощью современных информационных технологий можно по времени включать-выключать любые устройства и делать это можно с точностью до секунды.

Цель работы: создание электронного устройства, автоматизирующего школьный звонок в определенное время.

Для достижения цели необходимо решить следующие задачи:

изучить варианты управления включением-выключением устройств по времени;

изучить варианты управления включением-выключением устройств по времени с помощью Arduino и Node MCU;

изучить устройство и принцип действия электромагнитного реле;

изучить правила и овладеть навыками пайки радиодеталей;

создать демонстрационные устройства с разными способами регулировки включения-выключения по времени;

создать электронное устройство, автоматизирующее школьный звонок в соответствии с заданным временем, подключить данное устройство к школьному звонку;

получить отзывы работников и учащихся учреждения образования о работе автоматизированного устройства (звонка).

Объект исследования: информационные технологии получения данных о времени, способе задержки времени, устройство и принцип действия электромагнитного реле, способы пайки радиодеталей.

Предмет исследования: часы для микроконтроллера Arduino, микроконтроллер Node MCU, электромагнитное реле, школьный звонок.

ГЛАВА 1. ВАРИАНТЫ УПРАВЛЕНИЯ ВКЛЮЧЕНИЕМ-ВЫКЛЮЧЕНИЕМ УСТРОЙСТВ ПО ВРЕМЕНИ

1.1. Механические реле времени

Механические реле времени представляют собой часовой механизм с кулачками-контактами, которые нужно замкнуть в то время, когда устройство должно включиться. Недостатком таких устройств является длительный интервал между включениями, обычно он равен 15 минутам.

1.2. Электронные реле времени

Электронные реле времени — это электронные устройства, которые срабатывают по встроенным электронным часам. Настраивать реле можно обычно с точностью до минуты. Точность срабатывания зависит от точности хода электронных часов. Погрешность обычных электронных часов составляет около 20 секунд в месяц. Для того, чтобы они точно работали, их работу нужно постоянно корректировать. Еще одна характеристика электронных реле — это количество программ включения-выключения. У самых простых реле их может не хватить для наших целей. И самый большой недостаток таких реле — их высокая стоимость. В нашей стране цена самых простых устройств — сто и более рублей.

1.3. Реле времени на микроконтроллерах

Реле времени на микроконтроллере — это реле со встроенным программируемым микроконтроллером и электромагнитное реле. Время может считываться с электронных часов или с интернета, если есть в микроконтроллере встроенный модуль Wi-Fi. Сделать микроконтроллер реле времени можно с помощью программы, где можно задавать практически неограниченное количество включений-выключений устройства, точность срабатывания можно довести с точностью до секунды, ориентируясь на время в Интернет-сети. В Республике Беларусь такие устройства найти в продаже крайне сложно.

ГЛАВА 2. ВАРИАНТЫ УПРАВЛЕНИЯ ВКЛЮЧЕНИЕМ-ВЫКЛЮЧЕНИЕМ УСТРОЙСТВ СПОМОЩЬЮ ARDUINOINODEMCU

2.1. Управление временем с помощью команды Delay

В нашей работе мы будем рассматривать три способа управления временем: с помощью команды Delay; RTC часов для Arduino DS1307; NTP сервера точного времени.

Самый простой вариант управления устройства с помощью команду Delay. Команда Delay(x) просто задерживает выполнение программы на x миллисекунд. Максимальное значение x около 30 минут. Для того, чтобы создать график включения-выключения устройства, эту команду нужно будет использовать в цикле. Недостаток этого метода в том, что в программе кроме команды Delay есть и другие команды, выполнение которых тоже занимает определенное время, которое сложно точно подсчитать. Если имеются еще условные операторы, то оно будет непостоянным. Поэтому точность отсчета времени этим методом будет страдать. Но еще большим недостатком этого метода является то, что если отключится питание, то при новом включении программа начнет выполняться с произвольного времени, её нужно будет настраивать заново. Поэтому этот метод мы решили использовать как вспомогательный, когда нужно задержать время внутри программы на непродолжительный период.

2.2. Управление временем с помощью часов для Arduino

С помощью инструкции на сайте <https://iarduino.ru/> устанавливаем необходимые драйвера и подключаем RTC часы DS1307 к микроконтроллеру Arduino UNO[5]. Показания часов мы можем снимать в промежутках времени, начиная с миллисекунды и более. В результате эксперимента мы установили, что здесь мы можем запрограммировать практически неограниченное количество включений-выключений через любые промежутки времени с точностью до миллисекунды. Недостаток этого метода, как мы уже упоминали ранее, заключается в том, что при замене батарейки в часах нужно перенастраивать программу и точность часов плюс-минус 20 секунд в месяц. Вывод: этот метод можно использовать при должном не очень частом обслуживании устройства.

2.3. Управление временем с помощью NTP сервера точного времени

С помощью инструкций на сайте <https://radioprogram.ru/> устанавливаем нужные драйвера и подключаемся к NTP серверу точного времени [6]. Здесь мы пользуемся микроконтроллеров NodeMCU, где есть Wi-Fi модуль для связи с Интернетом-сетью. С помощью этого метода мы можем получить время с точностью до миллисекунды в любое время. При возникновении проблем (нет

связи с интернетом, отсутствует электричество и др.) при устранении неполадок время восстановится без корректировок. То есть, с помощью этого метода можно сделать абсолютно точное включение-выключение устройства практически неограниченное количество раз в сутки или в другой промежуток времени. Далее в работе мы рассмотрим схемы подключения и программы разных вариантов управления.

ГЛАВА 3. УСТРОЙСТВО И ПРИНЦИП ДЕЙСТВИЯ ЭЛЕКТРОМАГНИТНОГО РЕЛЕ

Школьный звонок работает от электрической сети 220 вольт, поэтому управлять им безопасно можно только через реле. В этой главе мы рассмотрим устройство электромагнитного реле и его принцип действия.

Основа электромагнитного реле - это электромагнит, который при протекании по нему небольшого тока замыкает или размыкает силовую цепь. То есть, можно маленьким напряжением замыкать или размыкать цепи с большими токами и большим напряжением. В наших устройствах микроконтроллер будет подавать напряжение в заданное время на электромагнит реле и реле будет замыкать или размыкать силовую цепь устройства. Более подробно можно ознакомиться на сайте <https://arduinomaster.ru/datchiki-arduino/podklyuchenie-rele-k-arduino> [1].

ГЛАВА 4. ОПИСАНИЕ ДЕМАНСТРАЦИОННЫХ УСТРОЙСТВ С РАЗНЫМИ СПОСОБАМИ РЕГУЛИРОВКИ ВКЛЮЧЕНИЯ ВЫКЛЮЧЕНИЯ ПО ВРЕМЕНИ

4.1. Принципиальная схема и программа для устройства на основе ArduinoUNO и электромагнитного реле

Принципиальная схема имеет следующий вид.



Схема 4.1 – управление временем с помощью ArduinoUno

Из данной схемы мы видим, что параллельно включателю звонка подключено электромагнитное реле, включением-выключением которого управляет микроконтроллер ArduinoUNO [2]. Фрагмент программы, которая управляет микроконтроллером имеет вид:

```
// Звонокнаурок 1
digitalWrite(4, LOW);
    delay (5000);
digitalWrite (4, HIGH);
    delay (30000);
digitalWrite (4, LOW);
    delay (5000);
digitalWrite (4, HIGH);

// Урок 1
```

```
Delay (1200000-40000);
    delay (1500000);
    // Звоноксурока
digitalWrite(4, LOW);
    delay (5000);
digitalWrite(4, HIGH);
    // Перерыв 1
    Delay (900000-5000);
    // Звонокнаурок 2
digitalWrite(4, LOW);
    delay (5000);
digitalWrite (4, HIGH);
    delay (30000);
digitalWrite (4, LOW);
    delay (5000);
digitalWrite(4, HIGH);
    // Урок 2
Delay (1200000-40000);
    Delay (1500000);
    // Звоноксурока 2
digitalWrite(4, LOW);
    delay (5000);
digitalWrite (4, HIGH);
    // Перерыв 2
    Delay (900000-5000);
    // Звонокнаурок 2
digitalWrite (4, LOW);
    delay (5000);
digitalWrite (4, HIGH);
```

```
    delay (30000);
    digitalWrite (4, LOW);
    delay (5000);
    digitalWrite (4, HIGH);
    // Урок 2
    Delay (1200000-40000);
    Delay (1500000);
    // Звонокурока 2
    digitalWrite (4, LOW);
    delay (5000);
    digitalWrite (4, HIGH);
    // Перерыв 2
    Delay (900000-5000);
```

Работа программы рассчитана на неделю. Включить устройство нужно за пять минут до начала учебных занятий в понедельник и выключить в пятницу по окончании. Если в пятницу не выключить звонок, он будет звонить на уроки и в субботу, и в воскресенье. Как мы уже говорили, недостатком этого метода является невозможность точно учесть время выполнения команд программы и при отключении питания в течении недели программа будет работать уже неверно. Но даже с такими недостатками время будет отсчитываться достаточно точно, а в случае непредвиденных обстоятельств, остается возможность звонить старым способом.

4.2. Принципиальная схема и программа для устройства на основе ArduinoUNO, электромагнитного реле и часов RTCDS1307

Принципиальная схема устройства

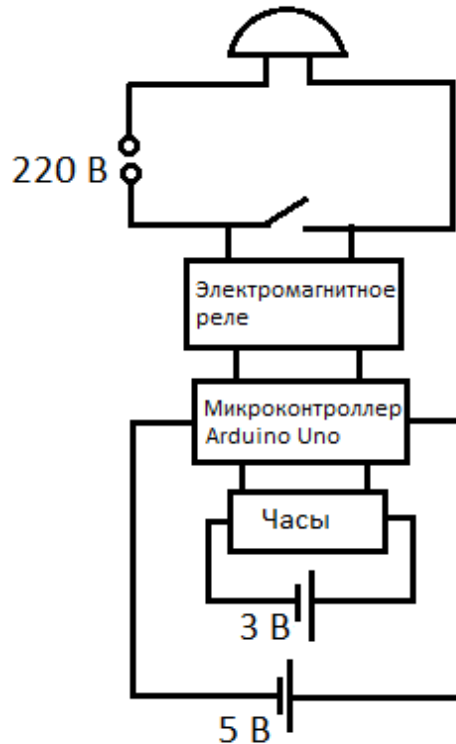


Схема 4.2 – управление временем с помощью ArduinoUno и часов DS1307

Из схемы 4.2. мы видим, что параллельно включателю звонка подключено электромагнитное реле, включением-выключением которого управляет микроконтроллер ArduinoUNO, который время берет с часов RTC DS1307. Подробнее о подключении часов к Arduino можно почитать здесь <https://robotchip.ru/obzor-chasov-realnogo-vremeni-ds1307/>[3].

Фрагмент программы, которая управляет микроконтроллером имеет вид

```

if ((s==0)&&(m==0)&&(ch==8)){ // 0 урок
    digitalWrite(4, LOW);
    delay (5000);
    digitalWrite (4, HIGH);
    delay (30000);
    digitalWrite (4, LOW);
    delay (5000);
    digitalWrite (4, HIGH);}
if ((s==0)&&(m==45)&&(ch==8)){
    digitalWrite (4, LOW);

```

```

        delay (5000);
        digitalWrite(4, HIGH); }
if ((s==0)&&(m==50)&&(ch==8)){// 1 урок
        digitalWrite (4, LOW);
        delay (5000);
        digitalWrite(4, HIGH);
        delay (30000);
        digitalWrite (4, LOW);
        delay (5000);
        digitalWrite(4, HIGH);}
if ((s==0)&&(m==35)&&(ch==9)){
        digitalWrite (4, LOW);
        delay (5000);
        digitalWrite (4, HIGH);}

```

Работа этого устройства рассчитана на всю четверть. На каникулах его необходимо отключать. Программа будет нуждаться в перенастройке только при замене батарейки в часах. Точность программы связана только с точностью хода часов. Как мы уже говорили выше, точность таких часов плюс-минус 20 секунд в месяц. Поэтому этим методом можно пользоваться, но программу нужно будет перенастраивать 2-3 раза в год.

4.3. Принципиальная схема и программа для устройства на основе NodeMCU и электромагнитного

Принципиальная схема этого устройства



Схема 4.3 – управление временем с помощью NodeMcu

Здесь из схемы мы видим, что параллельно выключателю звонка подключено электромагнитное реле, включением-выключением которого управляет микроконтроллер NodeMCU, который берет время с NTP сервера точного времени [4]. Фрагмент программы, которая управляет микроконтроллером имеет вид

```
if ((s==0)&&(m==0)&&(ch==8)){// 0 урок
```

```
    digitalWrite(4, LOW);
```

```
        delay (5000);
```

```
    digitalWrite(4, HIGH);
```

```
        delay (30000);
```

```
    digitalWrite(4, LOW);
```

```
        delay (5000);
```

```
    digitalWrite(4, HIGH);}

```

```
if ((s==0)&&(m==45)&&(ch==8)){
```

```
    digitalWrite(4, LOW);
```

```
        delay (5000);
```

```
    digitalWrite (4, HIGH);
```

```
if ((s==0)&&(m==50)&&(ch==8)){// 1 урок
```

```
    digitalWrite (4, LOW);
```

```
        delay (5000);
    digitalWrite (4, HIGH);
        delay (30000);
    digitalWrite (4, LOW);
        delay (5000);
    digitalWrite (4, HIGH);}
if ((s==0)&&(m==35)&&(ch==9)){
    digitalWrite (4, LOW);
        delay (5000);
    digitalWrite (4, HIGH);}
```

Это самое совершенное устройство, время берет с NTPсервера точного времени. При отключении питания не нужно перенастраивать, звонок срабатывает с точностью до секунды и не нуждается в дополнительных настройках. Вот такое устройство мы и подключили к школьному звонку. Но сначала были созданы и испытаны модели устройств, где вместо звонка, который подключается к напряжению 220 вольт использовались диоды, подключаемые к выводу микроконтроллера 5 вольт. Демонстрация первой модели – <https://youtu.be/EOC4yo923B8>. Демонстрация второй модели – <https://youtu.be/aaoY5lGt1is> .

ЗАКЛЮЧЕНИЕ

В процессе исследовательской работы мы изучили способы автоматического управления школьным звонком в том числе и с помощью микроконтроллеров Arduino и Node MCU, изучили устройство и принцип действия электромагнитного реле и порядок его подключения к микроконтроллерам, создали демонстрационную модель управления временем с помощью Arduino и часов DS1307 и с помощью микроконтроллера NodeMCU. И завершающим этапом работы стало создание реального устройства управления школьным звонком. Модели устройства были испытаны в мае 2022 года, а с 10 сентября 2022 года более эффективное устройство на микроконтроллере NodeMCU подключено к школьному звонку и успешно им управляет. На созданное устройство мы получили положительные отзывы от работников и учащихся учреждения образования (смотри ссылку: <https://youtu.be/sIQB1ZPJINI>).

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Arduinomaster [Электронный ресурс] – Режим доступа <https://arduinomaster.ru>. – Дата доступа: 25.10.2022
2. Робототехника Ардуино[Электронный ресурс] – Режим доступа <https://роботехника18.рф/%D1%80%D0%BE%D0%B1%D0%BE%D1%82%D0%BE%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0-%D0%B0%D1%80%D0%B4%D1%83%D0%B8%D0%BD%D0%BE>. - Дата доступа: 25.10.2022
3. ROBOCHIP [Электронный ресурс] – Режим доступа <https://robotchip.ru/> - Дата доступа: 25.10.2022
4. NodeMCU (ESP8266) для начинающих: что такое, как подключить [Электронный ресурс] – Режим доступа <http://edurobots.ru/2017/04/nodemcu-esp8266>. - Дата доступа: 25.10.2022
5. IARDUINO [Электронный ресурс] – Режим доступа <https://iarduino.ru/>- Дата доступа: 25.10.2022
6. RADIOPROG [Электронный ресурс] – Режим доступа <https://radioprogram.ru/>- Дата доступа: 25.10.2022

ПРИЛОЖЕНИЯ

<pre>#include <NTPClient.h> #include <ESP8266WiFi.h> #include <WiFiUdp.h> const char *ssid = "ZTE1"; const char *password = "29042019"; const long utcOffsetInSeconds = 10800; char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"}; // Определение NTP-клиента для // получения времени WiFiUDPntpUDP; NTPClienttimeClient(ntpUDP, "pool.ntp.org", utcOffsetInSeconds); void setup() { pinMode(4, OUTPUT); Serial.begin(115200); digitalWrite(4, HIGH); WiFi.begin(ssid, password); while (WiFi.status() != WL_CONNECTED) { delay (500); } timeClient.begin(); }</pre>	<pre>if ((s==0)&&(m==50)&&(ch==10)){ // 3 урок digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } if((s==0)&&(m==35)&&(ch==11)){ digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); }</pre>
<pre>void loop() { digitalWrite(4, HIGH); timeClient.update(); }</pre>	<pre>if ((s==0)&&(m==45)&&(ch==11)){ // 4 урок digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); }</pre>

<pre>ints,m,ch; s=timeClient.getSeconds(); m=timeClient.getMinutes(); ch=timeClient.getHours(); if ((daysOfTheWeek[timeClient.getDay()] =="Saturday") (daysOfTheWeek[timeClient.getDay()] =="Sunday")) {goto stop;}</pre>	<pre>delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } if ((s==0)&&(m==30)&&(ch==12)) { digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); }</pre>
<pre>if ((s==0)&&(m==0)&&(ch==8)) { // 0 ypok digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } if ((s==0)&&(m==45)&&(ch==8)) { digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); }</pre>	<pre>if ((s==0)&&(m==45)&&(ch==12)) { // 5 ypok digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } if ((s==0)&&(m==30)&&(ch==13)) { digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); }</pre>
<pre>if ((s==0)&&(m==50)&&(ch==8)) { // 1 ypok digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } if ((s==0)&&(m==35)&&(ch==9)) { digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); }</pre>	<pre>if ((s==0)&&(m==45)&&(ch==13)) { // 6 ypok digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } if ((s==0)&&(m==30)&&(ch==14)) { digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); }</pre>

<pre> if ((s==0)&&(m==50)&&(ch==9)) { // 2 урок digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } if ((s==0)&&(m==35)&&(ch==10)) { digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } </pre>	<pre> if ((s==0)&&(m==50)&&(ch==14)) { // 7 урок digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } if ((s==0)&&(m==35)&&(ch==15)) { digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } } stop: delay(1000);} </pre>
--	---

Таблица 1. Программа управления школьным звонком с помощью Arduino Uno

<pre> #include <iarduino_RTC.h> iarduino_RTC time(RTC_DS1307); void setup() { pinMode(4, OUTPUT); digitalWrite (4,HIGH); delay(300); time.begin(); time.settime(30 ,11,11,22,03,22,3); // 30 сек, 11 мин, 11 часов, 22, марта, 2022 года, среда } void loop(){ time.gettime(); // читаем время с часов, обновляя значения всех переменных. digitalWrite (4,HIGH); ints,m,ch,d; s=time.seconds; m=time.minutes; </pre>	<pre> if ((s==0)&&(m==45)&&(ch==11)) { // 4 урок digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } if ((s==0)&&(m==30)&&(ch==12)) { digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } </pre>
---	---

<pre>ch=time.Hours; d=time.weekday; if ((d==0) (d==6)) {goto stop;}</pre>	
<pre>if ((s==0)&&(m==0)&&(ch==8)){ // 0 ypok digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } if ((s==0)&&(m==45)&&(ch==8)){ digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); }</pre>	<pre>if ((s==0)&&(m==45)&&(ch==11)){ // 4 ypok digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } if ((s==0)&&(m==30)&&(ch==12)){ digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); }</pre>
<pre>if ((s==0)&&(m==50)&&(ch==8)){ // 1 ypok digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } if ((s==0)&&(m==35)&&(ch==9)){ digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); }</pre>	<pre>if ((s==0)&&(m==45)&&(ch==12)){ // 5 ypok digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } if ((s==0)&&(m==30)&&(ch==13)){ digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); }</pre>
<pre>if ((s==0)&&(m==50)&&(ch==9)){ // 2 ypok digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); }</pre>	<pre>if ((s==0)&&(m==45)&&(ch==13)){ // 6 ypok digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); }</pre>

<pre> delay(5000); digitalWrite(4, HIGH); } if ((s==0)&&(m==35)&&(ch==10)){ digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } </pre>	<pre> delay(5000); digitalWrite(4, HIGH); } if ((s==0)&&(m==30)&&(ch==14)){ digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } </pre>
<pre> if ((s==0)&&(m==50)&&(ch==10)){ // 3 урок digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } if((s==0)&&(m==35)&&(ch==11)){ digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } </pre>	<pre> if ((s==0)&&(m==50)&&(ch==14)){ // 7 урок digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } if ((s==0)&&(m==35)&&(ch==15)){ digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } stop: delay(1000); } </pre>

Таблица 2. Программа управления школьным звонком с помощью ArduinoUno и часов DS1307

<pre> #include <NTPClient.h> #include <ESP8266WiFi.h> #include <WiFiUdp.h> const char *ssid = "ZTE1"; const char *password = "29042019"; const long utcOffsetInSeconds = 10800; </pre>	<pre> if ((s==0)&&(m==50)&&(ch==10)){ // 3 урок digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } </pre>
--	--

<pre> char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"}; // Определение NTP-клиента для получения времени WiFiUDPntpUDP; NTPClienttimeClient(ntpUDP, "pool.ntp.org", utcOffsetInSeconds); void setup() { pinMode(4, OUTPUT); Serial.begin(115200); digitalWrite(4, HIGH); WiFi.begin(ssid, password); while (WiFi.status() != WL_CONNECTED) { delay(500); } timeClient.begin(); } </pre>	<pre> if((s==0)&&(m==35)&&(ch==11)){ digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } </pre>
<pre> void loop() { digitalWrite(4, HIGH); timeClient.update(); ints,m,ch; s=timeClient.getSeconds(); m=timeClient.getMinutes(); ch=timeClient.getHours(); if ((daysOfTheWeek[timeClient.getDay()] =="Saturday") (daysOfTheWeek[timeClient.getDay()] =="Sunday")) {goto stop;} </pre>	<pre> if ((s==0)&&(m==45)&&(ch==11)){ // 4 урок digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } if ((s==0)&&(m==30)&&(ch==12)){ digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } </pre>

<pre> if ((s==0)&&(m==0)&&(ch==8)) { // 0 ypok digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } if ((s==0)&&(m==45)&&(ch==8)) { digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } </pre>	<pre> if ((s==0)&&(m==45)&&(ch==12)) { // 5 ypok digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } if ((s==0)&&(m==30)&&(ch==13)) { digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } </pre>
<pre> if ((s==0)&&(m==50)&&(ch==8)) { // 1 ypok digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } if ((s==0)&&(m==35)&&(ch==9)) { digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } </pre>	<pre> if ((s==0)&&(m==45)&&(ch==13)) { // 6 ypok digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } if ((s==0)&&(m==30)&&(ch==14)) { digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } </pre>
<pre> if ((s==0)&&(m==50)&&(ch==9)) { // 2 ypok digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } if ((s==0)&&(m==35)&&(ch==10)) { </pre>	<pre> if ((s==0)&&(m==50)&&(ch==14)) { // 7 ypok digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); delay(30000); digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } if ((s==0)&&(m==35)&&(ch==15)) { </pre>

<pre>digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); }</pre>	<pre>digitalWrite(4, LOW); delay(5000); digitalWrite(4, HIGH); } } stop: delay(1000);} </pre>
--	--

Таблица 3. Программа управления школьным звонком с помощью Node Mcu

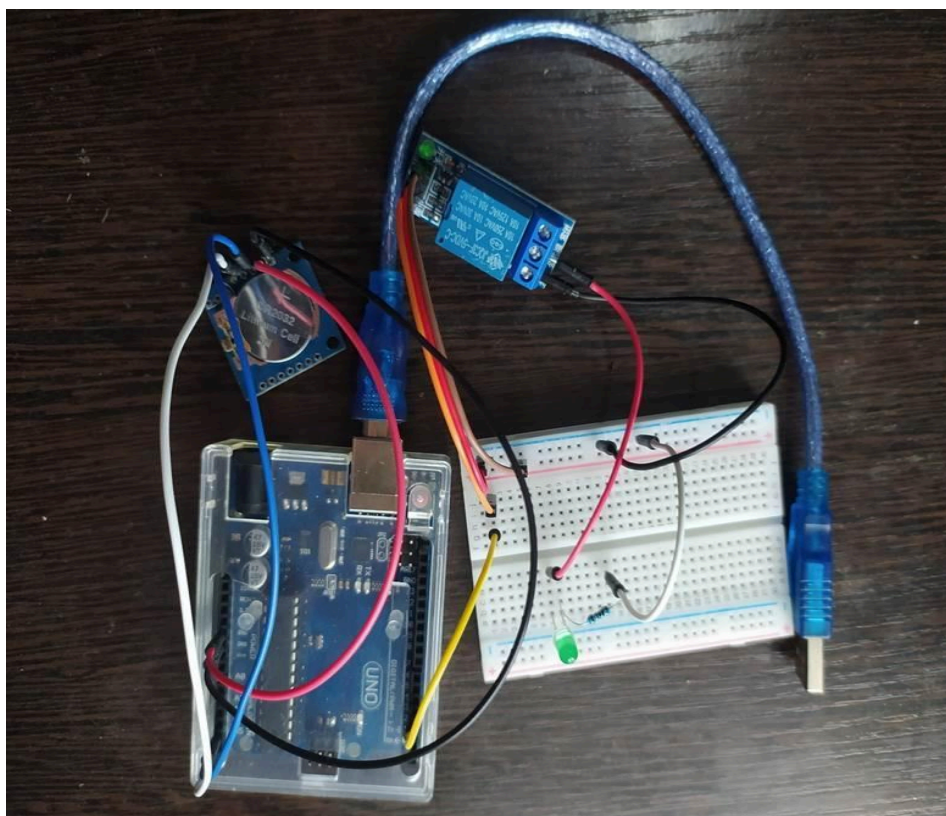


Фото 1. Модель управления школьным звонком с помощью Arduino Uno и часов DS1307

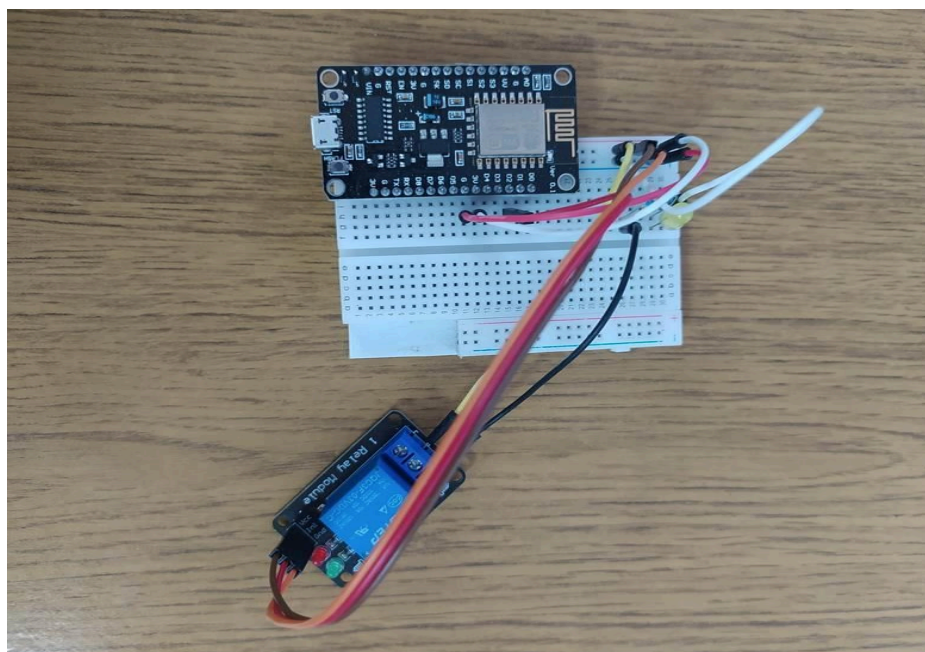


Фото 2. Модель управления школьным звонком с помощью Node Mcu



Фото 3. Подключенное устройство управления школьным звонком с помощью Node Mcu