



**ПРОФЕССИОНАЛЬНОЕ
ЗАВТРА**



VI Всероссийский сетевой конкурс студенческих проектов с участием студентов с инвалидностью

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФГБОУ ВО «Сибирский государственный индустриальный университет»
Института информационных технологий и автоматизированных систем**

Направление «Профессиональное завтра в профессии»

Номинация «Профессионально ориентированный проект»

**«Разработка системы автоматизированной подачи звонков в образовательной
организации»**

Выполнил:

Грищенко Денис Александрович

Руководитель:

Бабичева Надежда Борисовна, к.т.н.,

доцент каф. ПИТиП

Новокузнецк, 2023

Резюме Грищенко Дениса Александровича

Дата рождения: 12.07.1996

- Образование среднее профессиональное: НГГТКИ колледж-интернат

Дополнительная информация:

- В данный момент получаю высшее образование бакалавр по направлению информатика и вычислительная техника в университете СИБГИУ
- Окончил НГГТКИ колледж-интернат – слесарь по ремонту бытовой техники (2014-2017гг.)
- Окончил НГГТКИ колледж-интернат – информационные системы (2017-2021гг.)
- Получил во время второй профессии специальность ЭВМ

Опыт работы: мастер по ремонту ПК и ноутбуков «Сапфир» (2021-2021г.)

Достижения:

- Диплом в 1 Областной студенческой научно-практической конференции в рамках национальных проектов «Поддержка талантливой молодежи» и «Рабочие кадры для передовых технологий-2020» (2018г.)
- Диплом 3 степени за победу в 4 Областном первенстве по боулингу среди людей с ограниченными возможностями в классе «Смешанная команда» (2019г.)
- Сертификат за участие в 4 Областном первенстве по боулингу среди людей с ограниченными возможностями в классе «Смешанная команда» (2019г.)
- Диплом за участие в открытом первенстве по КибАтлетике «Сибирь-2019» (2019г.)
- Диплом 1 степени за победу в 5 Областном первенстве по боулингу среди людей с ограниченными возможностями в классе «Смешанная команда» (2019г.)
- Сертификат за участие в 5 Областном первенстве по боулингу среди людей с ограниченными возможностями в классе «Смешанная команда» (2019г.)
- Диплом 2 место в компетенции «Администрирование баз данных», 7 Региональный чемпионат по профессиональному мастерству среди инвалидов и лиц с ограниченными возможностями здоровья «Абилимпикс» (2022г.)

- Сертификат участника в компетенции «Администрирование баз данных», 7 Региональный чемпионат по профессиональному мастерству среди инвалидов и лиц с ограниченными возможностями здоровья «Абилимпикс» (2022г.)
- Диплом по шахматам в традиционной Спартакиаде первокурсника (2022г.)

Содержание

1 ОБЩАЯ ЧАСТЬ	7
1.1 Техническое задание на разработку	7
1.2 Обзор прототипов.....	7
1.3 Выбор и обоснование средств реализации	10
2 СПЕЦИАЛЬНАЯ ЧАСТЬ	14
2.1 Разработка принципиальной схемы проекта и реализация проекта.....	14
2.2 Программирование микроконтроллера.....	19
2.3 Руководство пользователя	23
ЗАКЛЮЧЕНИЕ	26
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	27
ПРИЛОЖЕНИЯ.....	28

ВВЕДЕНИЕ

Ни для кого не секрет, что XXI век – век информатизации общества. Сейчас существенно увеличивается роль компьютерной техники. Этот фактор затрагивает и образовательную сферу. Например, сейчас большинство образовательных учреждений уже имеют развитую компьютерную базу.

В России и мире существует несколько десятков различных реализаций автоматизированных звонковых систем. Первые такие системы появились ещё при СССР, причём кое-где до сих пор работали. Многие из них просто автоматизируют подачу традиционных звонков громкого боя. Некоторая часть позволяет вместо обычных звонков использовать систему проводного вещания, которая обычно входит в состав системы оповещения в школе.

В текущие время вопрос автоматизации и замена ручного труда наиболее актуален. Автоматизация подачи звонков при наличие правильного программного и аппаратного механизма может выполнять не только прямую функцию, но и дополнительные. Например, предупреждение учеников о начале уроков за минуту до начала (например, в столовой), механизировать предупреждающий сигнал в экстренных случаях, например, учебная пожарная тревога, разработать функцию голосового объявления, возможно, добавить музыкальное сопровождение. Минимальный набор функций такой программы может сводиться только к воспроизведению школьного звонка, максимальный – неограничен.

Использование автоматизированной системы подачи звонков в образовательном учреждении позволит: 1. Автоматизировать процесс подачи звуковых сигналов; 2. исключить человеческий фактор и минимизировать ошибки.

Цель данной работы – это создание и дальнейшее совершенствование автоматизированная система подачи звонков в образовательной организации на Arduino (АСПЗ).

Для достижения поставленной цели были сформулированы следующие задачи:

- проанализировать существующие прототипы устройств;
- выбрать средства реализации проекта;
- разработать принципиальную схему устройства;
- реализовать проект;
- проанализировать работу звонков и написать программный код для устройства;
- написать руководство пользователя;
- рассчитать себестоимость проекта.

Актуальность данной темы заключается в том, что разрабатываемое устройство отличается от подобных гибкостью настройки расписания звонков, независимостью от используемой платформы, удобством эксплуатации управляющим персоналом.

Спроектированное устройство найдет свое практическое применение в колледже ФКПОУ «НГГТКИ» Минтруда России, а также других схожих организациях.

Объект исследования: конструктивные и программные особенности автоматизированной системы подачи звонков в колледже на Arduino;

Предмет исследования: платформа быстрой разработки электронных устройств Arduino;

1 ОБЩАЯ ЧАСТЬ

1.1 Техническое задание на разработку

Разработка автоматизированной подачи звонка с использованием микроконтроллеров.

Прототипом данного устройства можно считать Авторингер-52СМ ИНФОЛАЙТ Устройство автоматической подачи звонков. Этот проект и стал основанием для данной дипломной работы.

Требования к функциональным характеристикам, следующие:

1. Проект должен быть реализован с помощью микроконтроллеров.
2. Система должна отражать и производить работу в реальном времени.
3. Система должна позволять задать порядок звонков по настраиваемому расписанию.

Требования к информационному обеспечению, следующие:

К устройству должна быть предоставлена документация, содержащая техническое описание и руководство пользователя.

Требования к программному обеспечению, следующие:

Программирование микроконтроллера предусматривается под OS Windows не ниже версии 7.

Требования к техническому обеспечению, следующие:

Рабочее напряжение микроконтроллера должно быть не менее 5В и предельное напряжение питания должно составлять от 6 до 20В. Объем памяти микроконтроллера должно быть не мене 16КБ, оперативная память должна быть не менее 1КБ.

1.2 Обзор прототипов

На начальном этапе создании автоматизированной подачи школьного звонка производился сбор и анализ информации, об аналогичных аппаратных средствах. На этом этапе происходит разбор ресурса, производится систематизация информации и выявление слабых и сильных сторон. После анализа решается, какие из достоинств необходимо реализовать в данном

проекте и как избежать допущенных ошибок, если таковые имеются. В сети интернет подобных прототипов очень мало и информация о них ограничена небольшими статьями.

Схема анализа прототипов:

- дизайн прототипа и удобство его использования;
- используемые микроконтроллеры на прототипах;
- используемые датчики.

ИНФОЛАЙТ Авторингер-52СМ - представляет собой прибор автоматической подачи звонков. Данное устройство предназначено для эксплуатации в учебных заведениях, в частности в среднеобразовательных школах. Вид устройства смотрите на рисунок 1.

Устройство оснащено цифровым дисплеем, на котором отображается текущий режим работы. Управление и программирование осуществляется при помощи клавиш на передней панели. Прибор имеет 9 встроенных мелодий, которые в свою очередь можно запрограммировать не только на разные уроки, но и на начало и конец уроков. Также имеется встроенная память, которая позволяет программировать желаемое расписание. Она включает в себя 32 сигнала на каждый день, что составляет 16 уроков.

Для вывода сигнала в приборе предусмотрен аудио выход RСА, который подключается к трансляционному усилителю мощности. Также имеется встроенное реле, которое предназначено для управления звонком громкого боя.

Питание устройства осуществляется от электросети ~220 В. При пропадании основного питания прибор автоматически переключается на встроенные резервный источник питания, сохраняя при этом все настройки и их параметры. Дополнительно в него интегрирован высокочастотный фильтр от сетевых помех.

Краткая характеристика:

- устройство автоматической подачи школьных звонков;
- цифровой дисплей;

- управление при помощи клавиш на передней панели;
- 9 встроенных мелодий звонков;
- память для программирования желаемого расписания;
- программирование 32 сигналов на каждый день;
- аудио выход RCA для подключения к трансляционному усилителю;
- реле управления звонком громкого боя;
- встроенный резервный источник питания;
- высокочастотный фильтр от сетевых помех.

Полная характеристика в таблице 1.



Рисунок 1- Схема подключения ИНФОЛАЙТ Авторингер-52СМ

Таблица 1 - Технические характеристики ИНФОЛАЙТ Авторингер-52СМ:

Тип устройства:	прибор автоматической подачи звонков
Назначение:	для образовательных школ
Дисплей:	цифровой
Органы управления:	кнопки на передней панели
Количество мелодий звонков:	9
Аудио выход на усилитель:	1 x RCA
Питание:	~220 В / 50 Гц
Потребляемая мощность:	1 Вт
Максимальная нагрузка на выходе реле:	4 А / ~220 В
Корпус:	пластик
Диапазон рабочих температур:	от +5° до +50° С
Размеры:	185 x 140 x 100 мм
Вес:	не более 1 кг
Цвет:	белый

1.3 Выбор и обоснование средств реализации

Для реализации данной дипломной работы уместно использовать такие аппаратные средства как:

- микроконтроллер;
- дисплей, программируемый 20x4;
- реле;
- цифровая кнопка;
- модуль часов
- светодиод

Контроллер – слово, образовавшееся от инфинитивной формы английского глагола «to control» – повелевать, управлять. Контроллеры

разделяются по группам и, в зависимости от принципа работы, используются в конструировании механических или электронных устройств.

Существующий рынок микроконтроллеров заполнен различными моделями такого вида устройств. Большинство производителей выпускают мини-компьютеры, в функционал которых заложена работа микроконтроллеров.

Топ-5 популярных микроконтроллеров:

– Iskra JS — это флагманская плата. В микропроцессор включен интерпретатор на языке JavaScript. Продукт создан на основе платформы Espruino. Подходит для совместной работы с Arduino. Техническое приспособление максимально совместимо с платами расширения и сенсорными инструментами. Стоимость такого микроконтроллера варьируется от тысячи четырехсот рублей.

– Mbed компании ARM. Компания ARM занимается созданием программно-аппаратных платформ и операционных систем для электронных девайсов с 32-разрядными микроконтроллерами из группы ARM Cortex-M. Аппаратная часть состоит из: платы ARM; платформы Mbed и FRGM, производитель последней - NXP Semiconductors; Nucleo аппаратуры STMicroelectronics и др.

– Wemos. В линейку разработок этой компании вошли такие известные модели, как wemos d1 mini, lolin esp32 oled wemos, wemos lolin32, bme wemod d1. Последнее устройство выпущено сравнительно недавно, встроен usb-порт и разъем под батарею, главная особенность esp-wroom-32 модуль с 4 Мб памяти.

– Arduino. Популярный микропроцессор среди начинающих электронщиков. Благодаря множеству библиотек и вспомогательных модулей на любой вкус, будь то датчик движения или wi-fi адаптер, плата стала любимцем конструкторов. Документация для этого микроконтроллера на русском языке легкодоступна в сети интернет. Для новичка Arduino легок для

освоения: достаточно создать код, загрузить его в платформу и подключить периферийные устройства.

– Raspberry. Наиболее популярная модель от этой компании – Raspberry Pi Zero. Плата построена по принципу Model A+, только гораздо уменьшена в размерах. На микрокомпьютере с таким микропроцессором легко запускается любой дистрибутив Linux. Платформа загружает и запускает Raspbian или подобную ОС. Встроенная память составляет 500Мбт. Этот мощный девайс служит для решения задач повышенной сложности: обрабатывание информации и визуализация.

– Дисплей. Символьный дисплей построен на базе ЖК дисплея типа STN (Super Twisted Nematic) под управлением контроллера HD44780 и имеет синхронный параллельный 8-битный интерфейс. Дисплей оснащён светодиодной подсветкой синего цвета и способен одновременно отображать до 80 символов (20 столбцов, 04 строки) от чего и произошло название дисплея: LCD2004. Контроллер HD44780 имеет ПЗУ в которой хранятся цифры, символы латиницы и некоторые иероглифы японского языка, для их отображения на дисплее. Отсутствующие символы, в т.ч. и символы кириллицы, можно загружать в память ОЗУ контроллера, для вывода на дисплей надписей на Русском языке или нестандартных символов (например, «смайликов»).

– Реле. Релейный модуль DFRobot-это стандартное реле, используемое с платой контроллера для сопряжения внешних электрических цепей или модулей. Некоторые из наиболее распространенных приложений с Arduino являются: Домашнее освещение, электрические приборы высокой мощности и другое оборудование. Модульная конструкция позволяет легко интегрировать его с платой контроллера. Реле интегрирует СИД состояния, для того чтобы подтвердить визуально состояние ВКЛЮЧЕНО-ВЫКЛЮЧЕНО. Он может управляться через цифровой порт ввода-вывода, с простой высокой или низкой на Arduino совместимых плат. Другие платы будут работать, требуя того же уровня напряжения. Другие виды

использования включают управление электромагнитными клапанами, лампами, двигателями и другими сильноточными или высоковольтными устройствами.

– Цифровая кнопка. Кнопка (или кнопочный переключатель) – самый простой и доступный из всех видов датчиков. Нажав на нее, вы подаете контроллеру сигнал, который затем приводит к каким-то действиям: включаются светодиоды, издаются звуки, запускаются моторы. В своей жизни мы часто встречаемся с разными выключателями и хорошо знакомы с этим устройством.

– Модуль часов. Если вы создаете устройство, которому нужно знать точное время, вам пригодится модуль часов реального времени RTC (Real Time Clock). Данные модули отсчитывают точное время и могут сохранять его даже при отключении основного питания при использовании резервного питания (батарейка CR2032 или литий-ионный аккумулятор LIR2032-3,6 В), которого хватит на несколько лет. Еще совсем недавно основным модулем RTC в среде Ардуинчиков являлся модуль на микросхеме DS1307. В этом модуле использовался внешний кварцевый генератор частотой 32кГц, при изменении температуры менялась частота кварца, что приводило к погрешности в подсчете времени.

– **Светодиод** или **светоизлучающий диод** — полупроводниковый прибор с электронно-дырочным переходом, создающий оптическое излучение при пропускании через него электрического тока в прямом направлении.

Излучаемый светодиодом свет лежит в узком диапазоне спектра, т. е. светодиод изначально излучает практически монохроматический свет (если речь идёт о СД видимого диапазона) — в отличие от лампы, излучающей более широкий спектр, от которой определённый цвет свечения можно получить лишь применением светофильтра. Спектральный диапазон излучения светодиода в основном зависит от типа и химического состава использованных полупроводников и ширины запрещённой зоны.

2 СПЕЦИАЛЬНАЯ ЧАСТЬ

2.1 Разработка принципиальной схемы проекта и реализация проекта

В ходе реализации данной дипломной работы были использованы такие аппаратные средства как:

- микроконтроллеры Arduino Uno r3;
- Цифровой Релейный Модуль 5А;
- Цифровая кнопка;
- I2C 20x4 Arduino ЖК-дисплей Модуль.
- Модуль часов DS1302.
- Цифровой Синий светодиодный Световой модуль

Плата **Arduino Uno** – центр большой империи Arduino, самое популярное и самое доступное устройство. В ее основе лежит чип ATmega – в последней ревизии Ардуино Уно R3 – это ATmega328 (хотя на рынке можно еще встретить варианты платы UNO с ATmega168). Большинство ардуинщиков начинают именно с платы UNO. В этой статье мы рассмотрим основные особенности, характеристики и устройство платы Arduino Uno ревизии R3, требования к питанию, возможности подключения внешних устройств, отличия от других плат (Mega, Nano) Смотрите распиновку в приложение Б. (таблица 2).

Таблица 2 - Характеристики Arduino Uno

Микроконтроллер	ATmega328
Рабочее напряжение	5В
Напряжение питания (рекомендуемое)	7-12В
Напряжение питания (предельное)	6-20В
Цифровые входы/выходы	14 (из них 6 могут использоваться в качестве ШИМ-выходов)

Аналоговые входы	6
Максимальный ток одного вывода	40 мА
Максимальный выходной ток вывода 3.3V	50 мА
Flash-память	32 КБ (АТmega328) из которых 0.5 КБ используются загрузчиком
SRAM	2 КБ (АТmega328)
EEPROM	1 КБ (АТmega328)
Тактовая частота	16 МГц

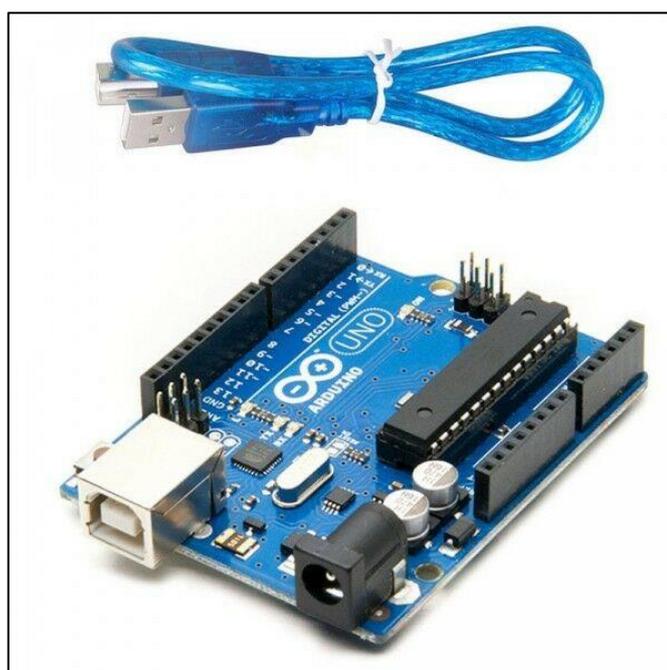


Рисунок 2 – Arduino uno r3

Цифровой Релейный Модуль 5А - чтобы управлять электроприборами с помощью микроконтроллера существует специальный тип выключателей — электромагнитные реле. Электромагнитное реле замыкает цепь нагрузки с помощью электромагнита. При подаче напряжения на катушку, в ней возникает поле, которое притягивает металлическую лапку, которая замыкает контакты нагрузки. Для работы с Arduino подойдут реле, которые управляются напряжением 5В. Такие реле способны коммутировать нагрузку

до 10А 30В постоянного тока или до 10А 250В переменного тока. Но напрямую к выводу Arduino подключить реле не получится, т.к. пины Arduino не могут обеспечить мощность, необходимую для работы катушки реле. Поэтому в схему управления реле включается транзистор по схеме с общим эмиттером для усиления управляющего тока. Схема управления реле с вывода Arduino с использованием npn-транзистора. Необходима установка ограничительного резистора между выводом Arduino и базой транзистора.

Технические характеристики релейного модуля:

- Рабочий ток одного канала – 15 – 20мА;
- Управляющее напряжение – 5В;
- Коммутируемая нагрузка – АС250V 10А, DC30V 10А;
- Светодиодная индикация состояния;
- Управление – n-канальное;



Рисунок 3 – Модуль реле

Цифровая кнопка. Нажмите, чтобы выпустить удовольствие! Наши переработанные цифровые кнопки поставляются со светодиодной подсветкой, погружной золотой поверхностью и яркими цветными шляпами. Надпись D в левом углу указывает, что это цифровой датчик. Иллюстрация кнопки, нарисованной на спине, уведомляющая о ее функции. Нажимать на нее очень весело, особенно с пластиковой крышкой. Длительный срок службы также обеспечивает его долговечность.

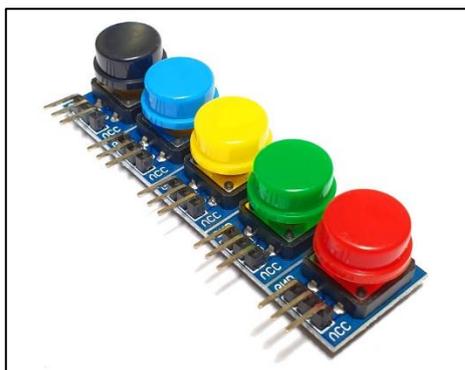


Рисунок 4 – Кнопки

I2C 20x4 Arduino ЖК-дисплей Модуль. ЖК-дисплей с подсветкой и установленным I2C/SPI конвертером. В отличие от обычного дисплея общается с контроллером Arduino по средствам 2-х проводной связи, это поможет сэкономить цифровые пины контроллера для подключения дополнительной периферии. Так же на I2C/SPI конвертере установлен потенциометр для регулировки яркости подсветки.

Характеристики:

- Цвет подсветки: Синий;
- Количество символов в строке: 20;
- Количество строк: 4;
- Язык: по умолчанию поддерживает латиницу;
- Интерфейсы: I2C/TWI;
- Напряжение питания: 5В.



Рисунок 5 – Дисплей I2C 20x4

Модуль часов DS1302. Модуль часов реального времени на микросхеме DS1302. Предназначен для учета времени, даты, дня недели и т.д. Содержит календарь до 2100 года с компенсацией високосных годов. Выгодно отличается от других модулей DIP корпусом чипа — если модуль применяется в устройстве и закреплен без возможности замены модуля (или наличием сложностей при замене), то можно будет очень легко поменять чип в случае выхода из строя.

Характеристики:

- Рабочее напряжение от 2 до 5.5 Вольт
- Рабочий ток при 2 В меньше 300 нА
- 3-х проводной интерфейс SCLK, I/O, RST. TTL совместим с 5 Вольт
- Кварц 32.768 кГц

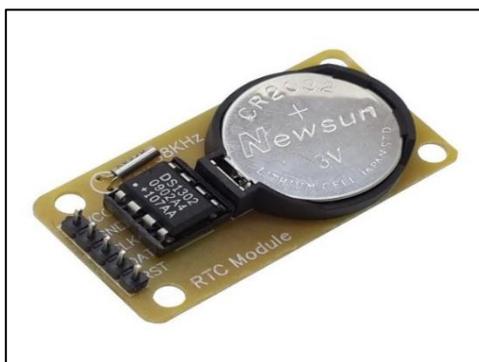


Рисунок 6 – Модуль часов DS1302

Светодиодный модуль – это система из одного или более led-кристаллов, которые помещаются в отдельный корпус. Каждый из светодиодных элементов имеет вывод для электропитания от сети.

В чем заключается отличие светодиодного модуля от отдельного светодиода как компонента электронных схем? Разница в том, что светодиодный модуль – это самостоятельное устройство, то есть:

- расположена светодиодная система в герметично закрытом корпусе, который имеет возможность крепиться на плоскостях различными способами;

- в светодиодный модуль входят резисторы, ограничивающие ток. Также напряжение питания стабильно и равняется, как правило, 12 В;
- каждый светодиодный модуль оснащен двумя парами выводов, которые соединяют несколько модулей в одну цепь при помощи кабеля.

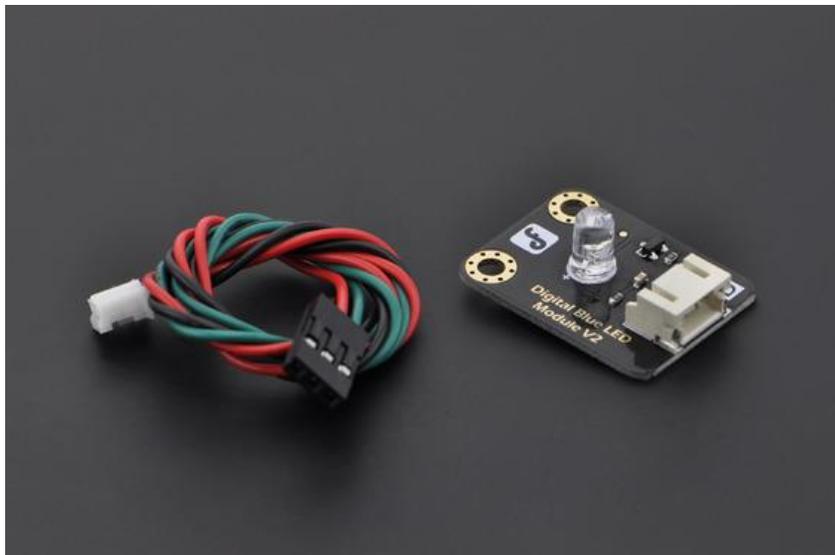


Рисунок 7 – Светодиодный модуль

Вид автоматизированной подачи звонка в соответствии с рисунком 7.



Рисунок 8 - Автоматизированной подачи звонка

Произвольную и электрическую схемы смотрите в приложения В и Г.

2.2 Программирование микроконтроллера

Интегрированная среда разработки (Integrated development environment-IDE), система программных средств, используемая программистами для разработки программного обеспечения. IDE обычно представляет собой единственную программу, в которой проводилась вся разработка. Она обычно

содержит много функций для создания, изменения, компилирования, развертывания и отладки программного обеспечения. Цель среды разработки заключается в том, чтобы абстрагировать конфигурацию, необходимую, чтобы объединить утилиты командной строки в одном модуле, который позволит уменьшить время, чтобы изучить язык, и повысить производительность разработчика.

Нужно отметить тот факт, что компании разрабатывающие микроконтроллеры так же разрабатывают IDE для своей продукции.

В разработки программного кода для портативного устройства сурдоперевода использовалась IDE «Arduino».

Arduino IDE — это бесплатная программная среда разработки, предназначенная для программирования одноименной платы. На сегодняшний день с помощью Arduino конструируют всевозможные интерактивные, обучающие, экспериментальные, развлекательные модели и устройства. Интерфейс сравнительно простой в освоении, его основой является язык C++ (<https://www.arduino.cc/en/main/software>). Изначально инструментарий поставляется на английском. И хотя команды меню довольно просты, программу легко можно перевести на нужный язык используя команды (File → Preferences → Language).

Настройка IDE:

- Соединяем плату Arduino с компьютером по средствам USB кабеля.

- Переходим (Пуск → Панель управления → Диспетчер устройств).

Находим "Порты COM и LPT" и видим к какому из COM портов она подключена.

Вид определения COM порта в соответствии с рисунком 9.

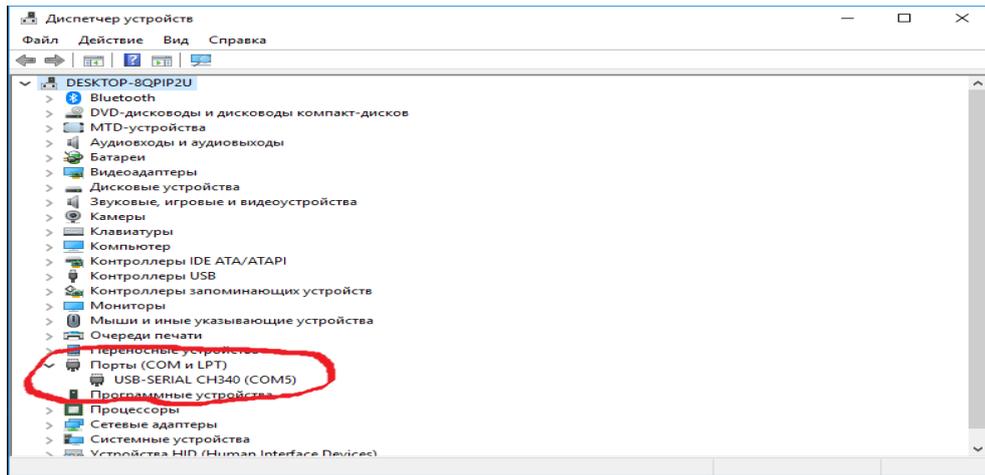


Рисунок 9-Определения COM порта

– Запускаем Arduino IDE и переходим (Инструменты → Порт).
Выбираем COM порт, определенный ранее.

Вид выбора COM порта в соответствии с рисунком 10.

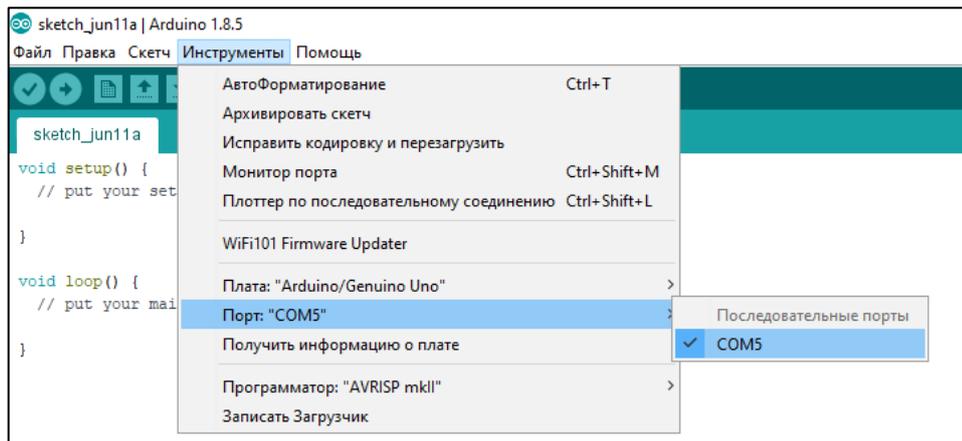


Рисунок 10-Выбор COM порта

– Следующим шагом будет выбор используемой платы.

Вид выбора используемой платы в соответствии с рисунком 11.

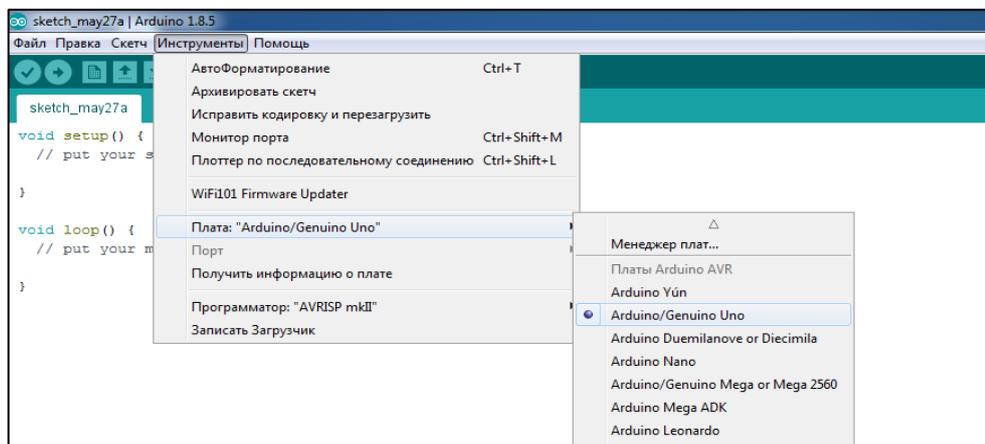


Рисунок 11-Выбор используемой платы

Так же может понадобиться выбрать процессор, используемый в архитектуре микроконтроллера.

Вид выбора процессора микроконтроллера в соответствии с рисунком 12.

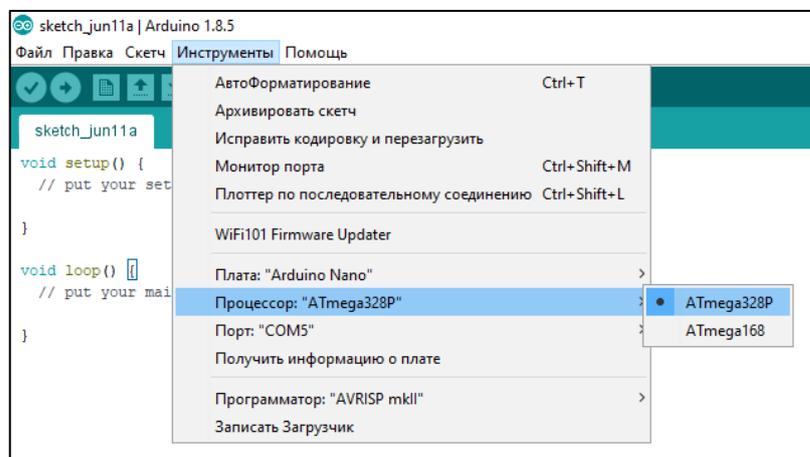


Рисунок 12-Выбор процессора микроконтроллера

Загрузка программного кода (sketch) в плату происходит с помощью нажатия кнопки «Загрузка» либо при выборе команд (Скетч→Загрузка).

Вид загрузки скетча в плату в соответствии с рисунком 13.

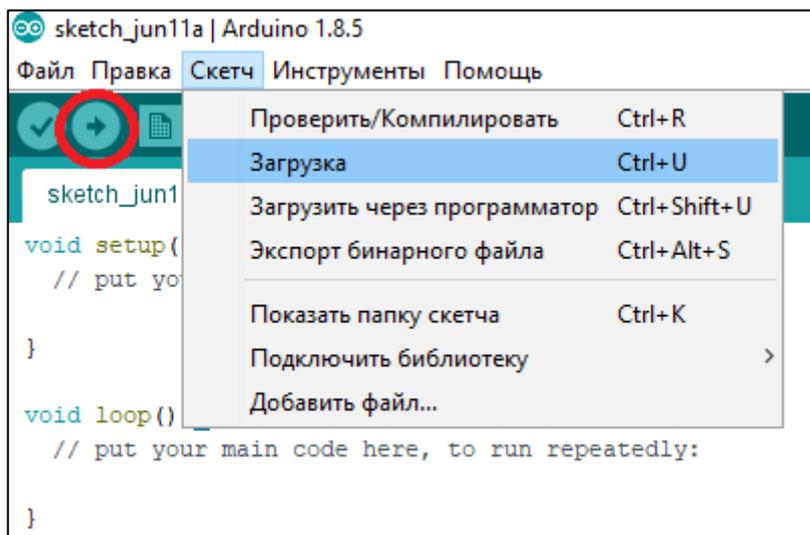


Рисунок 13-Загрузка скетча в плату

После вызова функции `setup()` (установка), которая инициализирует и устанавливает первоначальные значения, функция `loop()` (цикл) позволяет программе совершать вычисления и реагировать на них. Таким образом, плата Arduino постоянно получает показания последовательных портов.

Для заполнения памяти времени в соответствии с листингом 1

```

#include <EEPROM.h>
void setup()
{
Serial.begin(9600);
EEPROM.write(0, 8);
EEPROM.write(1, 30);
EEPROM.write(2, 9);
EEPROM.write(3, 20);
EEPROM.write(4, 10);
EEPROM.write(5, 15);
EEPROM.write(6, 11);
EEPROM.write(7, 05);
EEPROM.write(8, 12);
EEPROM.write(9, 00);
EEPROM.write(10, 12);
EEPROM.write(11, 45);
EEPROM.write(12, 13);
EEPROM.write(13, 35);
EEPROM.write(14, 14);
EEPROM.write(15, 30);
Serial.println("Готово");
}
void loop() {}

```

С программным кодом разработанного для устройства можно ознакомиться в «Приложении А».

2.3 Руководство пользователя

Для использования автоматизированной подачи звонка нужно подключить usb к ноутбуку и подождать пока включиться дисплей.

Вид устройства подключенное в соответствие с рисунком 14.

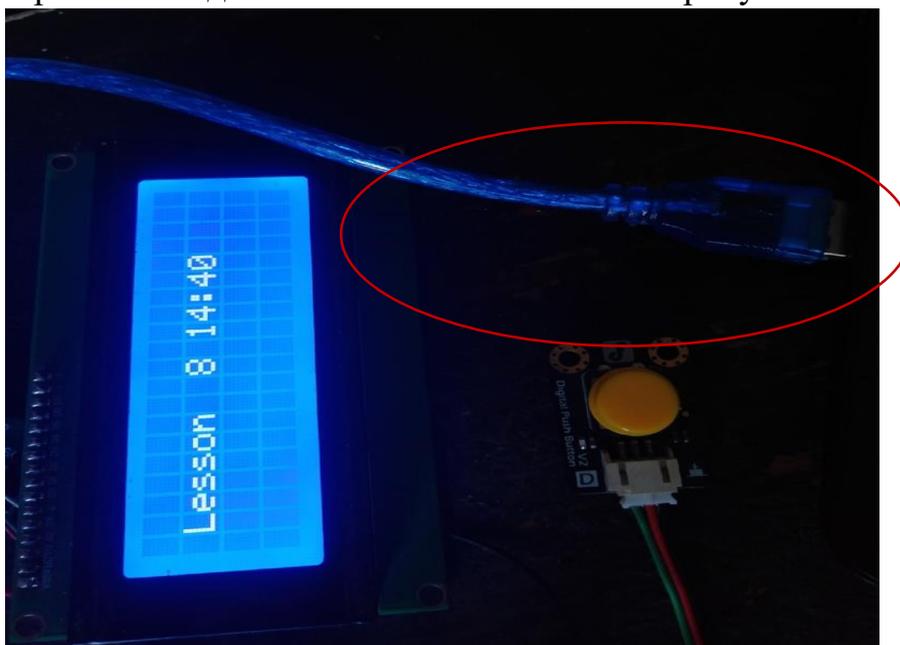


Рисунок 14 – Подключение к ноутбуку

Чтобы посмотреть сколько уроков находится в настройках или поменять время нажимаем кнопку 5 указанной в программном коде.

Вид переключения в настройки в соответствии с рисунком 15.



Рисунок 15 – С помощью кнопки нажимаем вперед

Чтобы вернуться на главный экран нажимаем кнопку 6 указанной в программном коде.

Вид переключения в настройки в соответствии с рисунком 16.



Рисунок 16 – С помощью нажимаем назад

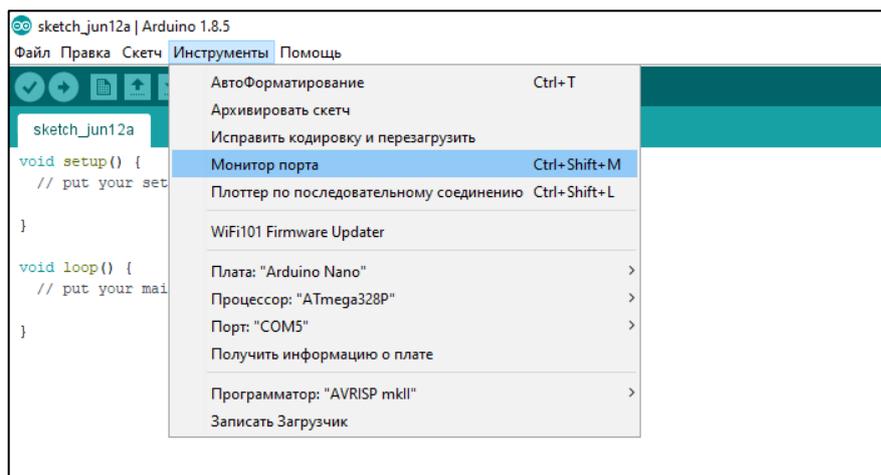


Рисунок 17- Запуск инструмента «Монитор порта» в IDE «Arduino»

ЗАКЛЮЧЕНИЕ

В результате проделанной научной работы было разработано электронное устройство, автоматизирующую подачи звонков. В результате разработки была создана схема функциональная, а также разработана программа управления микроконтроллера. В ходе проекта была реализована расписания школьных звонков. В программе управления реализованы часы реального времени.

В рамках данной цели были выполнены следующие задачи:

- произведен анализ существующих прототипов устройств;
- выбраны средства реализации проекта;
- реализован проект;
- написано руководство пользователя;
- рассчитана себестоимость проекта.

В ходе разработке данного научного проекта были рассмотрены и проанализированы существующие прототипы.

Результаты, достигнутые при выполнении дипломной работы:

- проанализирована радиотехническая литература;
- разработана структурная схема автомата подачи звонков;
- на основе структурной схемы спроектирована принципиальная электрическая схема автомата;
- разработана и написана программа на языке программирования C/C++ для автомата подачи звонков;
- при испытании автомата были выявлены некоторые недостатки, которые были устранены;

Цель проектной работы была достигнута: разработан и создан автомат подачи звонков, он будет использоваться в образовательных учреждениях.

Данный автомат подачи звонков может программно дорабатываться и совершенствоваться.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Горшенков А.А. Основы технологии и проектирования радиоэлектронной аппаратуры: учеб. Пособие // А.А. Горшенков, 2018. — 273 с.;
2. МакРобертс, М. Начала Arduino / М. МакРобертс - London: CUP, 2019. - 459 с.
3. Массимо, Б. Arduino для начинающих волшебников / Б. Массимо - М.: VSD, 2015. - 128 с.
4. Петин В.А. Проекты с использованием контроллера Arduino / Петин В.А. – БХВ Петербург [2019 с.239-242];
5. Радионов А.А. Электрооборудование и электроавтоматика [Текст] / А.А. Радионов. – Магнитогорск, 2019. – 126 с.;
6. Шуберт Ф.Е., «Светодиоды» , Москва, ФИЗМАТЛИТ, 2017 год;
7. Эванс, Б. Arduino блокнот программиста / Б. Эванс - London: CUP, 2017. - 40 с.
8. Сажнев, А. М. Цифровые устройства и микропроцессоры: учебное пособие для вузов/А. М. Сажнев. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2021. — 139 с
9. Огнева, М. В. Программирование на языке C++: практический курс: учебное пособие для вузов / М. В. Огнева, Е. В. Кудрина. — Москва: Издательство Юрайт, 2021. — 335 с
10. Гринберг, А.С. Информационные технологии моделирования процессов управления экономикой: Учебное пособие [Текст]/А.С. Гринберг, В.М. Шестаков. - М.: Юнити-Дана, 2018. - 399 с.
11. Методическое пособие-практикум по курсу "Основы экономики и управления производством" [Текст]/ ред. И.И. Сигов. - М.: Экономика, 2017. - 223 с.
12. Королев, А. В. Экономико-математические методы и моделирование: учебник и практикум для вузов/ А. В. Королев. — Москва: Издательство Юрайт, 2021. — 280 с

ПРИЛОЖЕНИЯ

Листинг «расписание звонков»

```

#include <Wire.h>
#include <EEPROM.h>
#include <LiquidCrystal_I2C.h>
#include "GyverButton.h"
#include <iarduino_RTC.h>

#define LESSONTIME 45

#define BTN1 5
#define BTN2 6
#define LED 8

LiquidCrystal_I2C lcd(0x27, 20 , 4);
GButton butt1(BTN1);
GButton butt2(BTN2);
iarduino_RTC time(RTC_DS1302, 4, 2, 3);

int a, currentTime;
int  timeMass[8];

byte menuLevel, currentLesson;

void setup() {
  Serial.begin(9600);
  pinMode(LED, OUTPUT);

  butt1.setType(LOW_PULL);      // установка типа кнопки
  (HIGH_PULL - подтянута к питанию, LOW_PULL - к gnd)
  butt2.setType(LOW_PULL);      // установка типа кнопки
  (HIGH_PULL - подтянута к питанию, LOW_PULL - к gnd)

  for (byte i = 0; i < 16; i += 2)
    timeMass[i / 2] = int(EEPROM.read(i)) * 60 + EEPROM.read(i +
1);

  lcd.init();
  lcd.backlight();
  lcd.clear();

  time.begin();
  // Ниже у нас прописываются временные значения в виде секунд,
  минут, часов, даты, включая месяц и год, и также можете указать
  день недели
  //time.settime(10, 14, 12, 4, 10, 17, 3); // 10 сек, 10 мин,
  10 час, 4 , октября, 2017 года, среда
  time.getTimeUnix();
}

void loop() {
  butt1.tick(); // Назад

```

```

butt2.tick(); // Вперёд

if (butt2.isHelded()) {
  if (menuLevel == 2) {
    menuLevel = 3;
    lcd.setCursor(12, 2);
    lcd.print("  -");
  }
  if (menuLevel == 1) {
    menuLevel = 2;
    lcd.setCursor(12, 2);
    lcd.print("-");
  }
  if (menuLevel == 0) {
    menuLevel = 1;
    currentLesson = 0;
    lcd.clear();
    lcd.setCursor(1, 0);
    lcd.print("Settings:");
    updSetTime();
  }
}

if (butt1.isHelded()) {
  if (menuLevel == 1) {
    menuLevel = 0;
    lcd.clear();
  }
  if (menuLevel == 2) {
    menuLevel = 1;
    lcd.setCursor(12, 2);
    lcd.print(" ");
    for (byte i = 0; i < 7; i++)
      if (timeMass[i] + LESSONTIME > timeMass[i + 1])
        timeMass[i + 1] = timeMass[i] + LESSONTIME + 5;
    if (timeMass[6] + LESSONTIME > timeMass[7])
      timeMass[7] = timeMass[6] + LESSONTIME + 5;
    for (byte i = 0; i < 8; i++) {
      EEPROM.write(i * 2, timeMass[i] / 60);
      EEPROM.write(i * 2 + 1, timeMass[i] % 60);
    }
  }
  if (menuLevel == 3) {
    menuLevel = 2;
    lcd.setCursor(12, 2);
    lcd.print("-  ");
  }
}

if (butt2.isClick()) {
  if (menuLevel == 1) {
    currentLesson += 1;
    currentLesson = currentLesson % 8;
  }
}

```

```

    }
    if (menuLevel == 2) {
        if (timeMass[currentLesson] < 1440 - 60 * (9 -
currentLesson))
            timeMass[currentLesson] += 60;
    }
    if (menuLevel == 3) {
        timeMass[currentLesson] += 5;
        if (timeMass[currentLesson] % 60 == 0)
timeMass[currentLesson] -= 60;
    }
    if (menuLevel) updSetTime();
}

if (butt1.isClick()) {
    if (menuLevel == 3) {
        timeMass[currentLesson] -= 5;
        if (timeMass[currentLesson] % 60 == 55)
timeMass[currentLesson] += 60;
    }
    if (menuLevel == 2) {
        if (timeMass[currentLesson] > 60 * currentLesson + 2)
            timeMass[currentLesson] -= 60;
    }
    if (menuLevel == 1) {
        currentLesson -= 1;
        if (currentLesson == 255) currentLesson = 7;
    }
    if (menuLevel) updSetTime();
}
time.getTimeUnix();
a = time.Hours * 60 + time.minutes;
if (a != currentTime)
    currentTime = a;
if (menuLevel == 0)
    updMainScreen();
delay(900);
}

void updMainScreen() {
    for (byte i = 0; i < 8; i++) {
        if (timeMass[i] <= currentTime && currentTime < timeMass[i]
+ LESSONTIME ) {
            if (currentTime - timeMass[i] < 10) digitalWrite(LED,
HIGH);
            else digitalWrite(LED, LOW);
            lcd.setCursor(1, 1);
            lcd.print("Lesson ");
            lcd.print(i + 1);
            break;
        }
        if (i < 7)

```

```

        if (timeMass[i] + LESSONTIME <= currentTime && currentTime
< timeMass[i + 1]) {
            lcd.setCursor(1, 1);
            lcd.print("Breaks  ");
            lcd.print(i + 1);
            break;
        }
        if (!(timeMass[0] <= currentTime && currentTime <
timeMass[7] + LESSONTIME) || !time.weekday) {
            lcd.setCursor(1, 1);
            lcd.print("Free time");
            break;
        }
    }
    lcd.setCursor(11, 1);
    lcd.print(time.Hours / 10);
    lcd.print(time.Hours % 10);
    lcd.print(":");
    lcd.print(time.minutes / 10);
    lcd.print(time.minutes % 10);
}
void updSetTime() {
    lcd.setCursor(1, 1);
    lcd.print("Lesson ");
    lcd.print(currentLesson + 1);
    lcd.setCursor(11, 1);
    lcd.print(timeMass[currentLesson] / 600);
    lcd.print(timeMass[currentLesson] / 60 % 10);
    lcd.print(":");
    lcd.print(timeMass[currentLesson] % 60 / 10);
    lcd.print(timeMass[currentLesson] % 10);
}

```

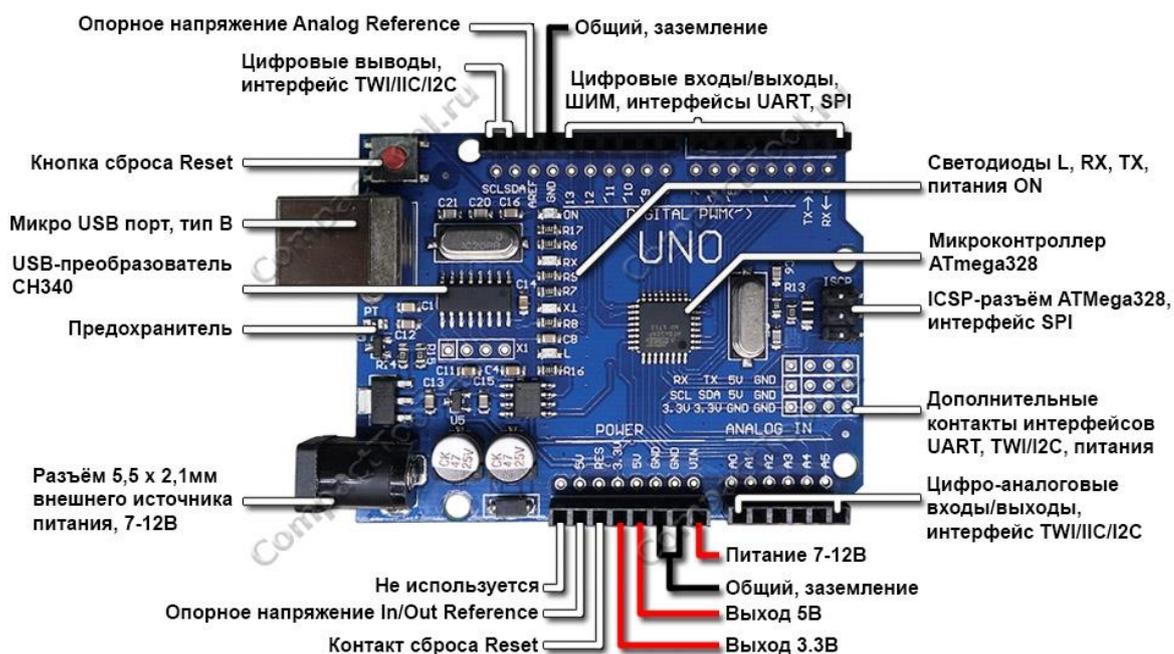


Рисунок 18 – Плата Arduino распиновка

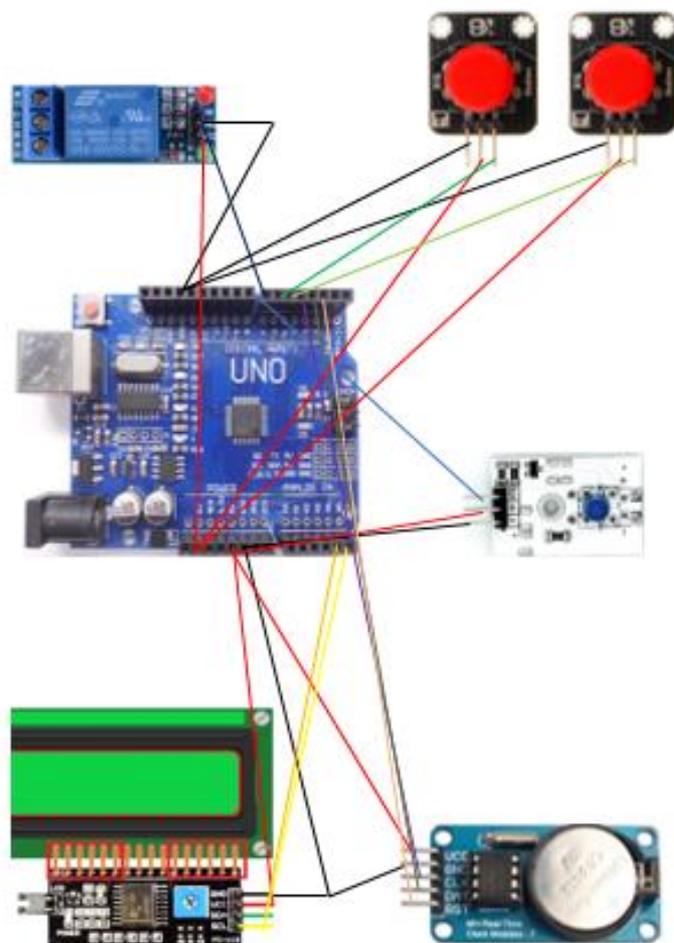


Рисунок 19 – Произвольная схема автоматизированной подачи звонка

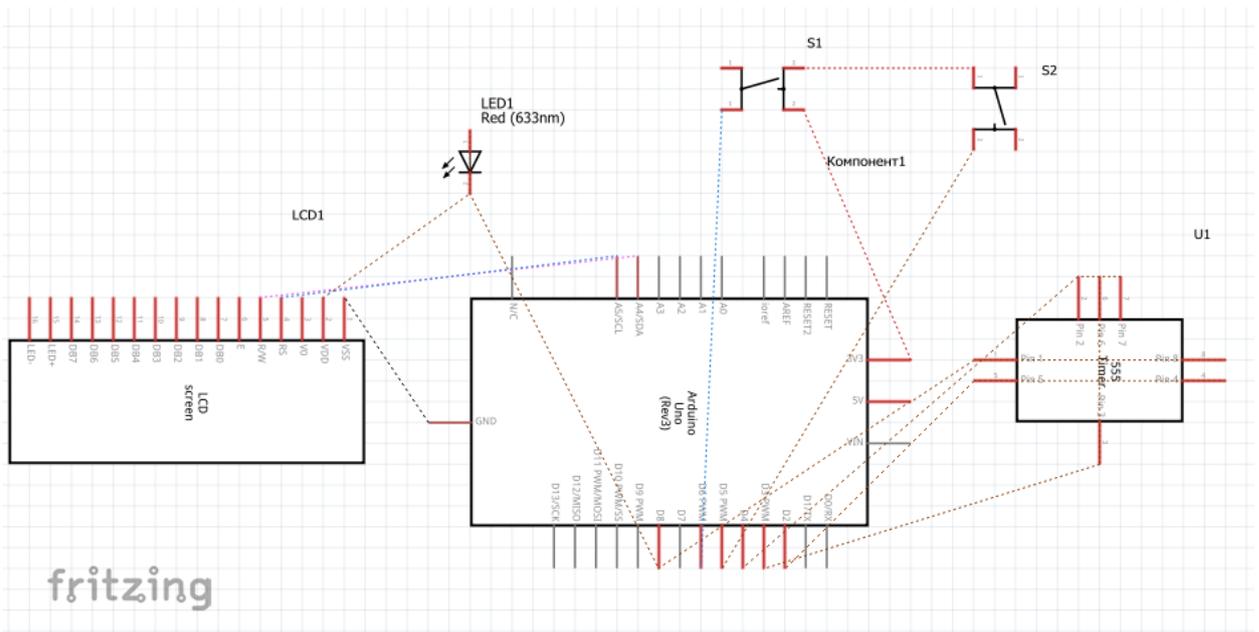


Рисунок 20 - Электрическая схема автоматизированной подачи звонка