Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ
«МИФИ»

ИНСТИТУТ ЛАЗЕРНЫХ И ПЛАЗМЕННЫХ ТЕХНОЛОГИЙ
КАФЕДРА №31 ПРИКЛАДНАЯ МАТЕМАТИКА

Чернявский Андрей Дмитриевич

Отчет

по проектной практике на тему:

ТЕМА ПРОЕКТА

Разработка бинарных часов с индикацией, имеющих синхронизацию времени по сети

Руководитель проекта

_____ М.А. Чмыхов

«___» _____ 2021 г.

Москва, 2021 г.

Содержание

1 Используемые инструменты **4**

1.1 История создания Arduino 4

1.2 Возможности Arduino 5

1.3 Аппаратная часть Arduino 5

1.4 Программная часть Arduino 5

2 Принципиальная схема часов с синхронизацией по сети **6**

2.1 Подход к решению задачи 6

2.2 Генератор импульсов 6

2.3 Синхронизация с сервером точного времени 6

2.4 Индикация 6

3 Реализация проекта **7**

3.1 Генератор импульсов 7

3.2 Выбор платы Ардуино - Uno или Nano 9

3.3 Индикаторы на светодиодах 10

3.4 Принципиальная схема 11

3.5 Макетная (монтажная) плата 12

4 Программирование **12**

4.1 Общие положения 12

4.2 Ссылка на репозиторий GitHub 13

4.3 Обработка сигналов часов реального времени 13

4.4 Формирование сигналов для бинарной индикации 13

5 Работающий прототип часов **16**

6 Заключение **18**

Аннотация

Цель проекта – описать процесс создания прототипа часов на платформе Arduino с бинарной индикацией и синхронизацией по сети.

Для достижения цели проекта ставим следующие задачи:

- подготовить сравнительную характеристику плат Arduino Uno и Arduino Nano, выяснить, какая из них лучше подходит для использования в проекте;
- определить остальные компоненты, с использованием которых можно реализовать часы с синхронизацией;
- составить принципиальную схему и создать часы
- сделать возможным установку времени на часах и выбрать способ реализации - с помощью кнопок или через COM-порт;
- разработать программный код для работы компонентов схемы.

1 Используемые инструменты

1.1 История создания Arduino

Arduino представляет собой комбинацию аппаратной и программной частей для простой разработки электроники.

Arduino создавалось преподавателями для обучения школьников и студентов электротехнике, программированию, радиоэлектронике, системам автоматизации. Идея имела успех и проект пошел дальше. Благодаря открытой архитектуре любой желающий может производить микроконтроллеры, дополнять модельный ряд, писать программы. Все схемы и исходный код программ есть в открытом доступе.

После того как Arduino получило более широкое распространение, многие производители электроники начали выпускать собственные платы на базе микроконтроллеров Arduino.

Название конструктора отсылает нас к средневековому королю Италии Ардуину, поскольку начинала проект команда из пяти итальянцев.

1.2 Возможности Arduino

С технической точки зрения, Arduino умеет принимать и отправлять сигналы в соответствии с инструкциями в прошивке. Это позволяет получать и обрабатывать информацию с сенсоров и передавать команды исполнительным механизмам или другим устройствам. Например, микроконтроллер может получать данные с датчиков температуры, давления, влажности и выводить сводную информацию на дисплей и управлять светом, моторами, приводами.

В сети часто встречаются проекты умного дома на базе данного конструктора.

1.3 Аппаратная часть Arduino

Аппаратная часть включает в себя большое количество видов плат Arduino со встроенными программируемыми микроконтроллерами, а также дополнительные модули.

Классические платы спроектированы для монтажа в стопки через штыревые разъёмы, то есть без пайки контактов. Порты ввода-вывода микроконтроллеров оформлены в виде штыревых линеек. Это главный плюс для начинающих - не нужно использовать паяльник и ничего паять. На платах сделаны удобные контакты, которые можно соединять удобными перемычками с любыми сторонними модулями, дисплеями, сенсорами и др.

Микроконтроллеры отличаются наличием предварительно прошитого в них загрузчика. С помощью этого загрузчика пользователь загружает свою программу в микроконтроллер без использования традиционных отдельных аппаратных программаторов. Загрузчик соединяется с компьютером через интерфейс USB или с помощью отдельного переходника **UART-USB**.

Существует целая линейка микроконтроллеров. Поскольку наша задача не является сложной в вычислительном плане, мы воспользуемся недорогой платой **Arduino Nano v3.0**.

1.4 Программная часть Arduino

Программная часть состоит из бесплатной оболочки (IDE – Integrated development environment). В этой среде удобно писать скетчи и загружать их на микроконтроллер. В среде разработки уже предустановлено большое количество примеров и дополнительных библиотек.

Язык программирования называется Arduino C и представляет собой язык C++.

Среда разработки включает текстовый редактор, транслятор, средства автоматизации сборки, отладчик.

2 Принципиальная схема часов с синхронизацией по сети

2.1 Подход к решению задачи

Для реализации часов с синхронизацией нам необходимы следующие ключевые компоненты:

- Генератор импульсов с более-менее приемлемой точностью, к примеру, +/- 1 секунда в час, то есть в относительном выражении 10^{-4};
- Способ синхронизации, в качестве которого мы выбираем сигнал навигационной системы;
- Устройство отображения, представляющее каждый разряд (часы, минуты, секунды) в двоичном виде, то есть имеющее минимум два состояния;
- Схема управления.

2.2 Генератор импульсов

В качестве генератора импульсов в типовых решениях используется кварцевый генератор, отличающийся приемлемой стабильностью тактовых колебаний, особенно при условии стабилизации температуры резонатора.

2.3 Синхронизация с сервером точного времени

Синхронизация с NTP была рассмотрена во 2-ом семестре, поэтому сегодня мы исключаем этот раздел.

2.4 Индикация

При двоичном представлении используются только два состояния индикатора – 0 (ноль) и 1 (единица). В качестве устройство отображения используем светодиоды.

Мы будем представлять дату в двоичном формате.

Всего мы насчитали, что необходимо 10 светодиодов:

- необходимо 2 (два) ряда светодиодов – по одному ряду на часы и минуты;
- для отображения минут необходимо 6 светодиодов ($2^6 = 64$, что достаточно для представления состояния 60 минут);
- для отображения часов необходимо 4 светодиода ($2^4 = 16$, что достаточно для представления состояния 12 часов).

В приведенном примере включенный светодиод означает единицу, выключенный – ноль. Каждая горизонтальная линия отображает степень двойки снизу-вверх, от нулевой до шестой. Поэтому при чтении информации с часов надо просуммировать значения колонок с включенными светодиодами.

3 Реализация проекта

3.1 Генератор импульсов

В качестве источника импульсов используем популярную микросхему DS3231 от Dallas Semiconductors. Точность часов зависит от точности кварцевого резонатора и точности соответствия между ёмкостной нагрузкой схемы тактового генератора и внутренней ёмкостью кварцевого резонатора. Дополнительная погрешность будет вноситься дрейфом частоты кварцевого резонатора, происходящим из-за температурных перепадов. Вообще говоря, данный модуль считается не считается точным – есть и более совершенные модели, в которых сильно снижены температурные колебания за счет термостата.

Как указано в описании модуля, «Часы реального времени с последовательным интерфейсом DS3231 – это малопотребляющие полные двоично-десятичные часы-календарь, включающие 56 байтов энергонезависимой статической ОЗУ. Адреса и данные передаются последовательно по двухпроводной двунаправленной шине.

Часы-календарь отсчитывают секунды, минуты, часы, день, дату, месяц и год. Последняя дата месяца автоматически корректируется для месяцев с количеством дней меньше 31, включая коррекцию високосного года. Часы работают как в 24-часовом, так и в 12-часовом режимах с индикатором AM/PM»

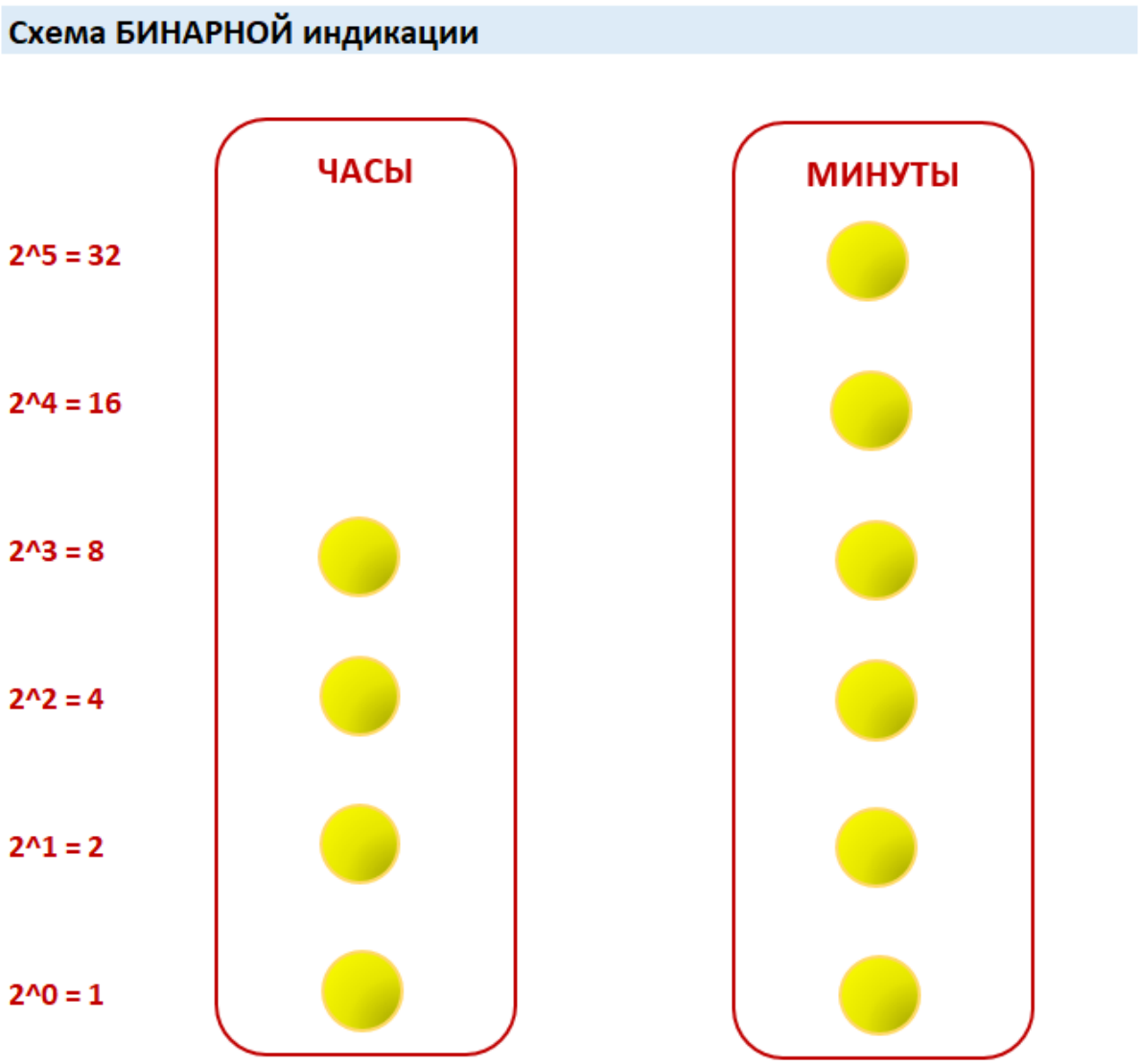

Рис. 1: Схема индикации

Литиевая батарея емкостью 4,8 А/ч способна поддерживать модуль в течение 10 лет (при комнатной температуре).

В нашей задаче нас интересуют:

- VCC и GND – питание +5В и земля соответственно;
- SCL – Serial Clock Input - вход последовательных синхроимпульсов) – используется для синхронизации данных по последовательному интерфейсу;
- SDA – Serial Data Input/Output serial - вход/выход последовательных данных) – вывод входа/выхода для двухпроводного последовательного интерфейса

Сравнительная характеристика плат.

Параметр	Arduino Uno	Arduino Nano
Микроконтроллер	ATmega328P, тактовая частота 16 МГц	ATmega328P, тактовая частота 16 МГц
Пины питания	**VIN:** Входной пин (напряжение от 7 до 12 вольт) **5V** – выходной пин, максимальный ток 1А **3V** - Выходной пин от стабилизатора напряжения, максимальный ток 150 мА Выводы GND – 3 штуки Выводы AREF и IOREF	**VIN:** Входной пин (напряжение от 7 до 12 вольт) **5V** *– выходной пин, максимальный ток 800 мА **3V** *- Выходной пин от стабилизатора напряжения, максимальный ток 50 мА *питать плату через выводы 5V и 3.3V не рекомендуется, есть риск спалить плату Выводы GND – 2 штуки Вывод AREF
Пины общего назначения	14 цифровых + 6 аналоговых = 20 всего	14 цифровых + 6 аналоговых = 20 всего
АЦП	6 (пины A0-A5), разрядность постоянна и равна 10 бит	8 (пины A0-A7), разрядность постоянна и равна 10 бит
ШИМ	6 пинов, разрядность постоянна и равна 8 бит	6 пинов, разрядность постоянна и равна 8 бит
I2C	SDA – A4, SCL – A5	SDA – A4, SCL – A5
Размеры	69 на 53 мм	18 на 45 мм
Абсолютный максимум тока через пин	40 мА	40 мА
Рекомендуемый ток через пин	20 мА	20 мА
Максимальный ток через всю плату	200 мА	200 мА

Вывод: так как для проекта будет задействовано 10 цифровых пинов, 3 пина GND, пины SCA, SDL и 5V, то плата Arduino Nano может также быть использована, если соединить два провода, ведущих к GND, в один.

Рис. 2: Сравнение плат Ардуино

3.2 Выбор платы Ардуино - Uno или Nano

Arduino Uno и Arduino Nano имеют 20 выводов общего назначения (14 цифровых + 6 аналоговых) – у обеих плат достаточно выводов для использования в нашем проекте, так как всего планируется использовать 14 выводов – 10 для светодиодов, 2 для кнопок и 2 для часов RTC.

Также обе платы имеют выводы для I2C (A4 и A5), что критически важно для проекта, так как через I2C будет осуществляться передача данных с RTC на плату.

Arduino Uno и Arduino Nano имеют одинаковый рекомендуемый ток через каждый вывод – 20 миллиампер. Также для них одинаковый максимально допустимый ток через всю плату, равный 200 миллиампер. Эти параметры важно учесть, чтобы подобрать резисторы с соответствующим сопротивлением.

Главное различие для проекта состоит в размерах плат – Uno имеет габариты 69 мм на 53 мм, а Nano – 18 мм на 45 мм. В связи с этим, а также согласно указаниям научного руководителя для проекта была выбрана плата Arduino Nano.

Рис. 3: Таблица напряжений светодиодов

3.3 Индикаторы на светодиодах

Для панели индикации можно использовать обычные китайские светодиоды на 3 V.

Подключать светодиоды необходимо через резистор сопротивлением 510 Ом, поскольку светодиод обладает малым внутренним сопротивлением и при прямом подключении может перегореть.

В проекте используются светодиоды красного и желтого цвета, эксплуатация которых должна отвечать следующим условиям:

- чтобы светодиод не подвергся деградации, через него должен проходить ток не более 20 мА. Это значение тока одинаково для красных и желтых светодиодов;
- чтобы вывод платы Arduino Nano не испортился, через него должен проходить ток не более 20 мА.

Для красных светодиодов:

При токе через светодиод, равном 20 мА падение напряжения на красном светодиоде составляет примерно 1.8 вольт. Так как питание идет от источника с напряжением 5В, то падение напряжения на резисторе будет равно 5,0 – 1,8 = 3,2 В. В проекте

используются резисторы с сопротивлением 510 Ом, поэтому ток через резистор будет иметь значение I = 3,2 В/510 Ом = 6,27 х 0,001 А или 6,27 мА.

6,27 мА < 20 мА (значительно меньше), следовательно, красные светодиоды не имеют риска деградировать, так как использованы достаточно мощные резисторы.

Для желтых светодиодов:

При токе через светодиод, равном 20 мА падение напряжения на желтом светодиоде составляет примерно 2,1 вольт. Так как питание идет от источника с напряжением 5В, то падение напряжения на резисторе будет равно 5,0 – 2,1 = 2,9 В.

В проекте используются резисторы с сопротивлением 510 Ом, поэтому ток через резистор может иметь максимальное значение I = 2,9 В/510 Ом = 5,69 х 0,001 А или 5,69 мА.

5,69 мА < 20 мА (значительно меньше) , следовательно: Желтые светодиоды не имеют риска деградировать, так как использованы достаточно мощные резисторы.

Таким образом, через каждый вывод платы Arduino Nano течет ток значительно меньший 20 мА, следовательно, выводы платы не имеют риска испортиться.

Так как всего в проекте используется 10 светодиодов (4 желтых и 6 красных), то общий ток через плату не будет превышать предельного значения в 200 мА, потому что через каждый светодиод проходит ток меньший 20 мА.

3.4 Принципиальная схема

Принципиальная схема отражает схему соединения основных компонентов.

Для черчения под Arduino удобно использовать бесплатную программу (среду разработки) Tinkercad. С использованием данной программы можно сформировать макет реальной печатной платы. Программа включает набор готовых компонентов, в частности, макетные и монтажные платы (в том числе Arduino), целый набор аналоговых и цифровых микросхем, любые радиодетали: конденсаторы, транзисторы, резисторы, светодиоды, батарейки, кнопки.

Схема доступна для рисования, как в окне «Макетная плата», так и в окне «Принципиальная схема» простым перетаскиванием нужных компонентов на рабочее поле. При выборе окна «Печатная плата» можно приступить к разводке проводников и размещению элементов.

Результат работы экспортируется в pdf-файл или jpeg.

3.5 Макетная (монтажная) плата

Как говорилось, удобство Arduino как обучающего конструктора состоит в использовании типовых элементов, соединяемых между собой без пайки.

Концепция беспаечной макетной платы позволяет многократно использовать одни и те же компоненты обучающего конструктора.

Однако, если мы хотим долгое время использовать устройство, то нам необходимо пропаивать контакты во избежание их окисления. Тогда следует воспользоваться обычной платой с пайкой.

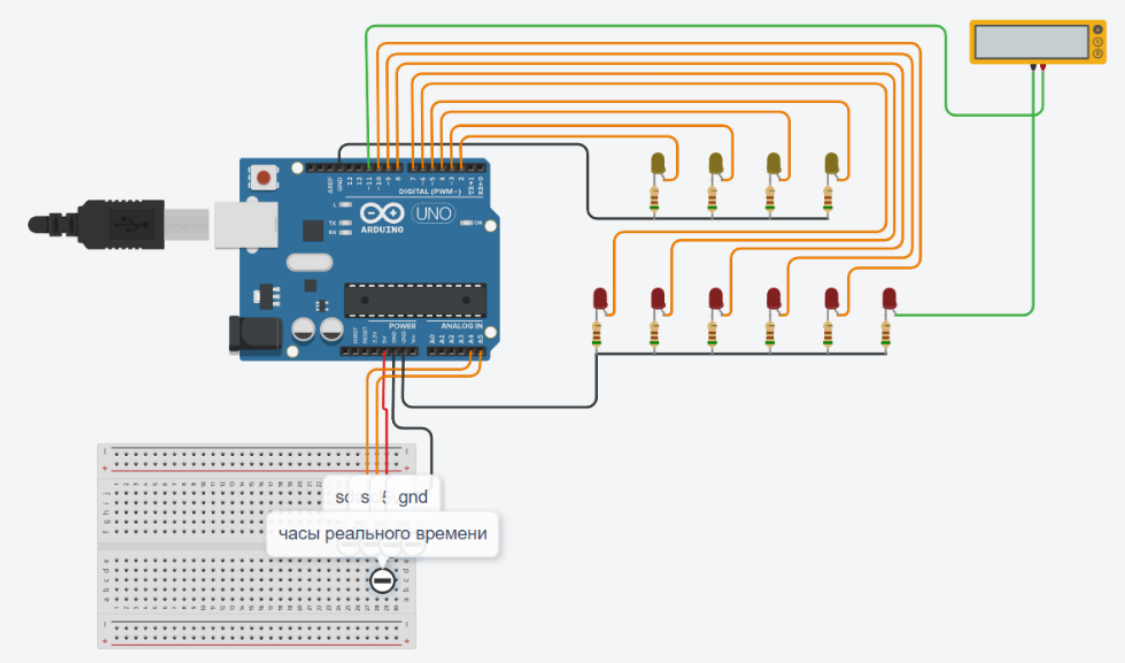

Рис. 4: Принципиальная схема

4 Программирование

4.1 Общие положения

Программирование микроконтроллеров Arduino осуществляется на языке программирования C++. Этот язык является низкоуровневым, поэтому считается сложным и имеет высокий порог вхождения. Но для программирования Arduino используется упрощенная версия C++.

Также для упрощения разработки прошивок существует множество функций, классов, методов и библиотек, благодаря чему работать с этими микроконтроллерами достаточно удобно.

Рис. 5: Код установки часов ч.1

4.2 Ссылка на репозиторий GitHub

Программный код, использованный при выполнении проекта, размещен в репозитории GitHub по адресу: `https://github.com/Andrey621/clock_with_binary_indication`

4.3 Обработка сигналов часов реального времени

Для работы с модулем часов реального времени нам потребуются библиотеки DS1307RTS Time. Наша несложная задача заключается в получении информации о времени из модуля RTC.

4.4 Формирование сигналов для бинарной индикации

Последняя задача будет сложнее предыдущей, потому что потребуется преобразование десятичного представления в бинарное, то есть число часов требуется преобразовать в двоичное для представления 4-мя индикаторами (светодиодами), число минут - для представления 6-ю светодиодами.

Задача будет решаться использованием bitRead() и созданием массивов (для светодиодов, отображающих количество часов и минут).

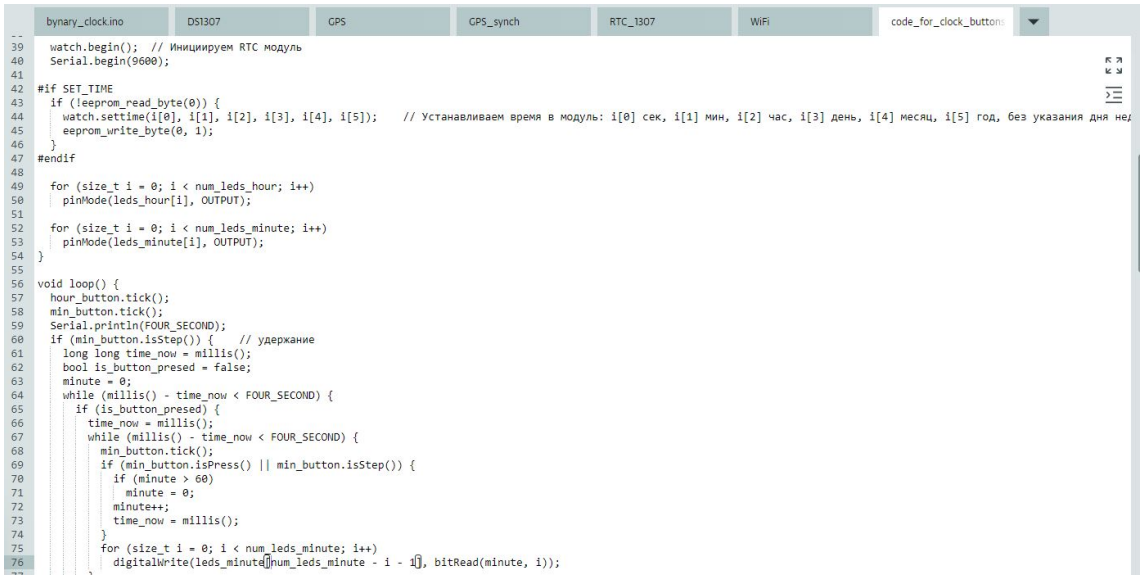

Рис. 6: Код установки часов ч.2

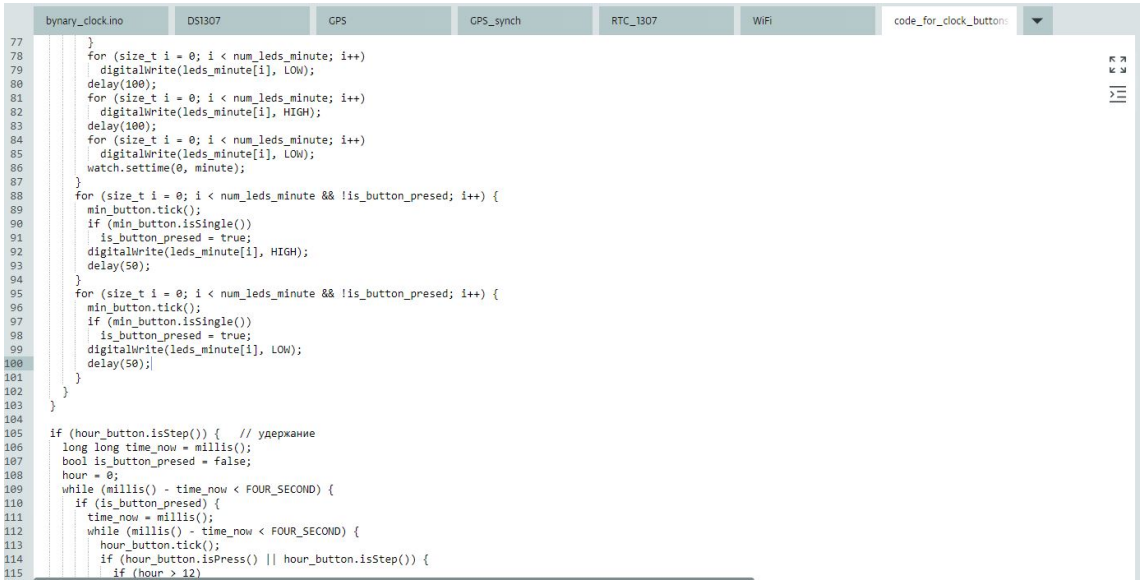

Рис. 7: Код установки часов ч.3

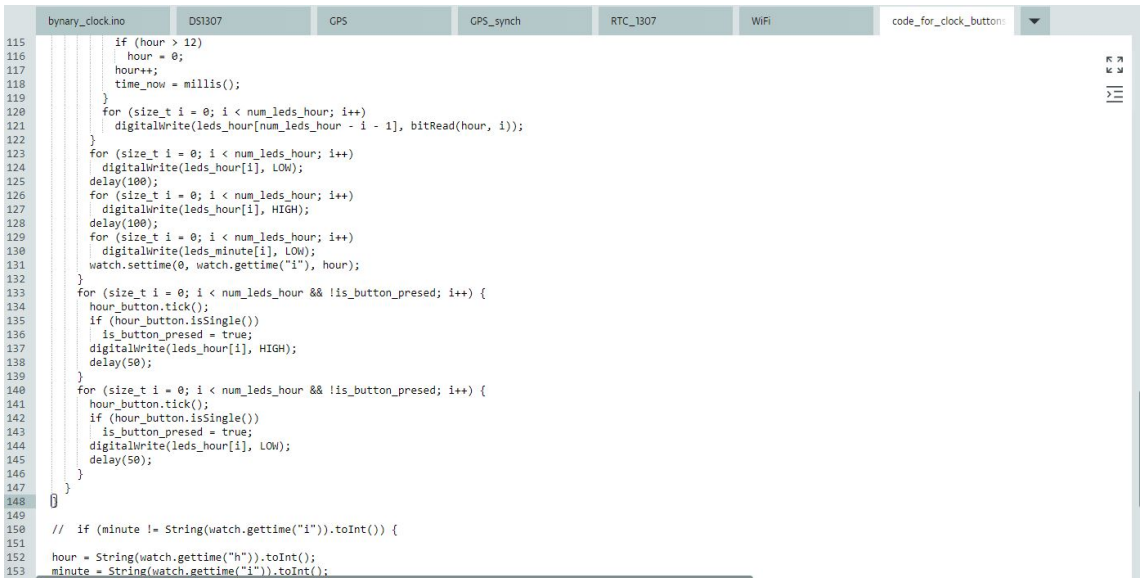

Рис. 8: Код установки часов ч.4

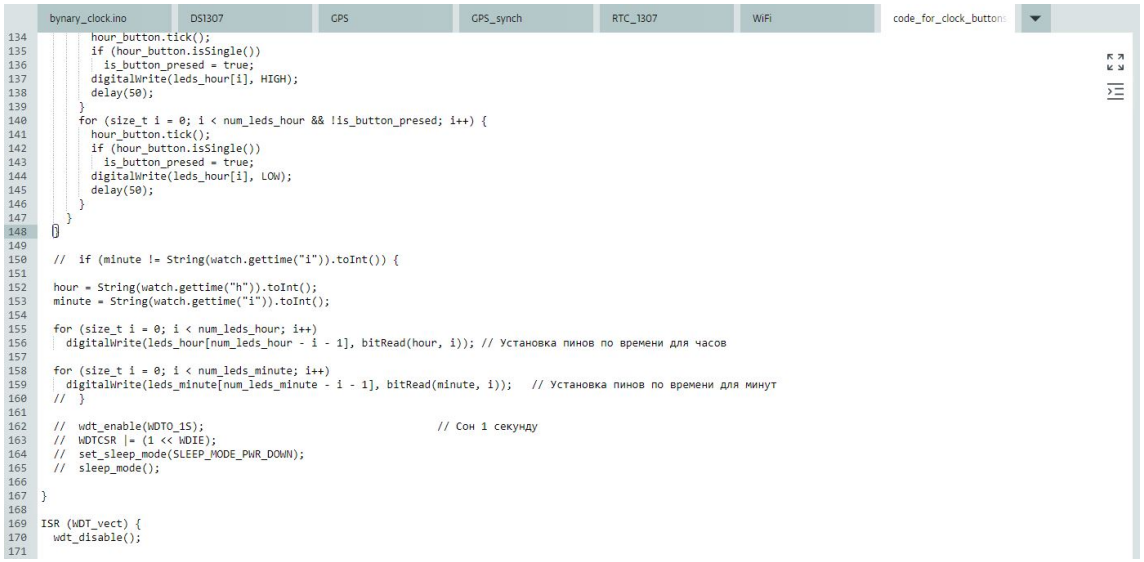

Рис. 9: Код установки часов ч.5

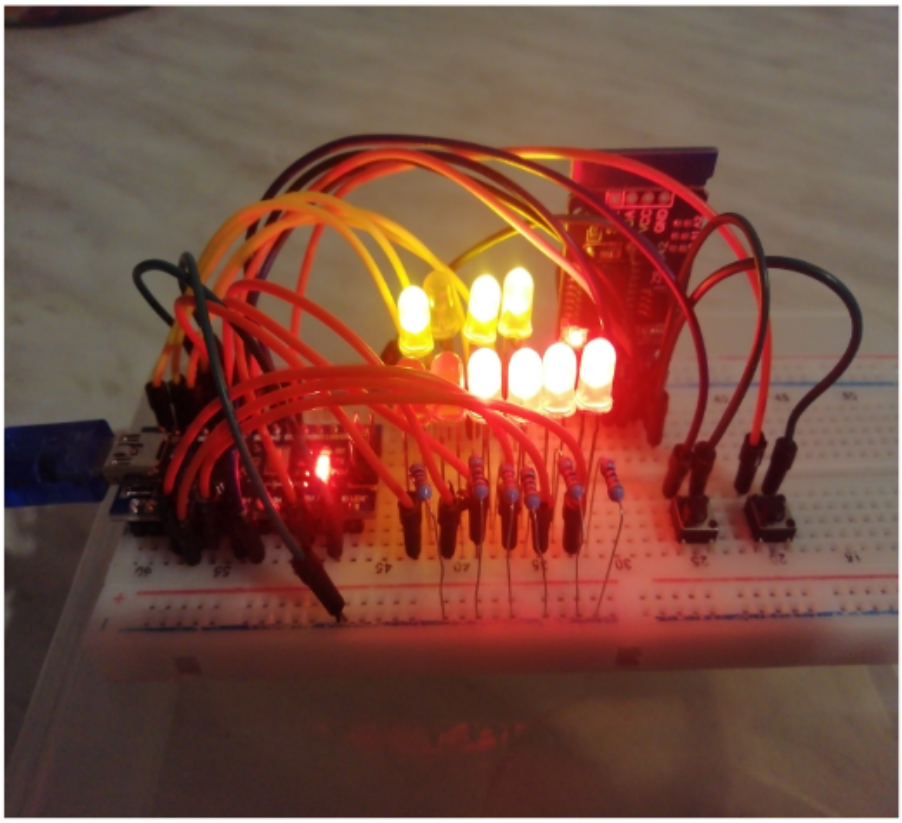

Рис. 10: Бинарные часы 1

5 Работающий прототип часов

В развитие проекта в 3-ем семестре проект был реализован "в железе"(см фото ниже):

Рис. 11: Бинарные часы 2

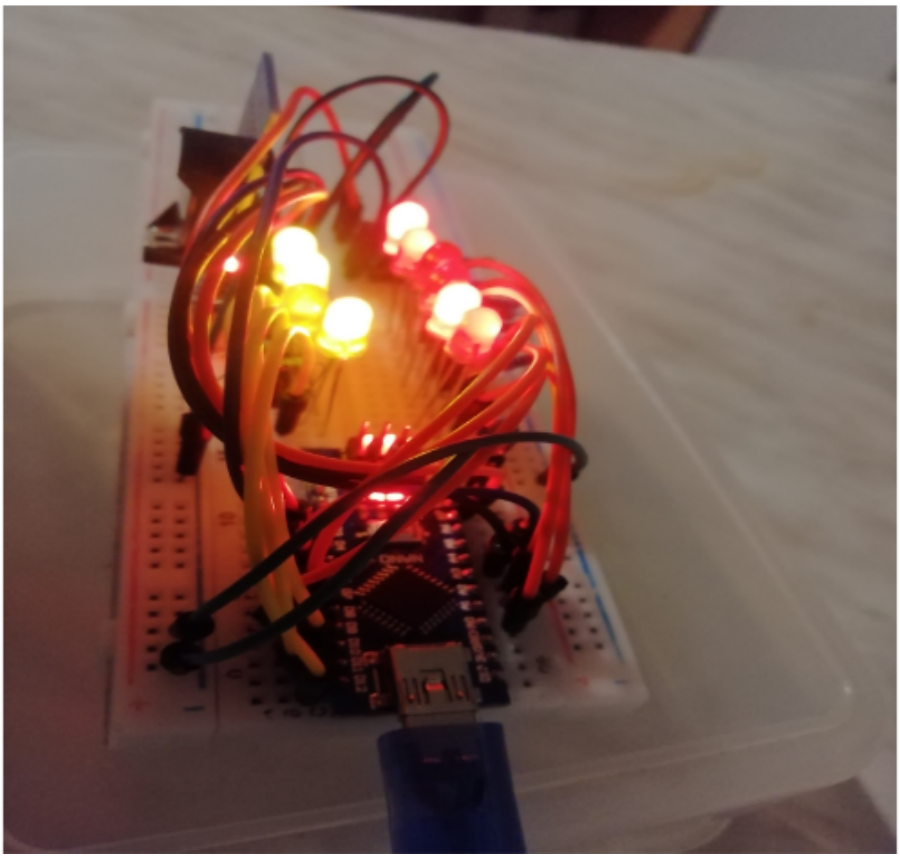

Рис. 12: Бинарные часы 3

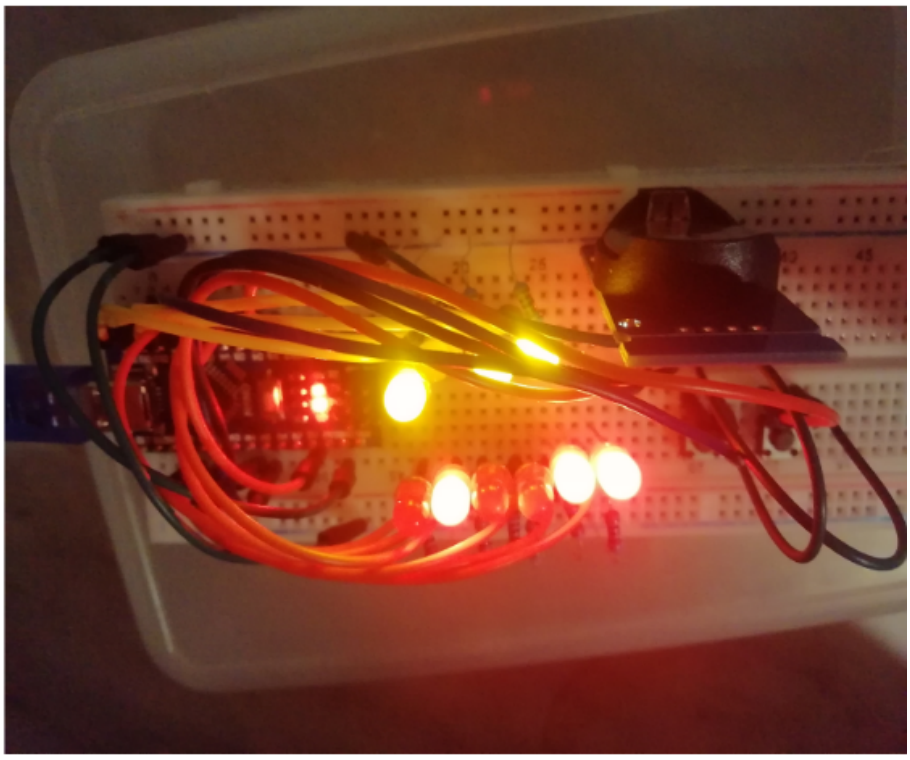

Рис. 13: Бинарные часы 4

6 Заключение

В результате исполнения проекта были решены следующие задачи:

- созданы часы на платформе Arduino;
- подготовлена сравнительная характеристика плат Arduino Uno и Arduino Nano. Выяснено, что для проекта лучше подходит плата Arduino Nano;
- определены остальные компоненты (светодиоды, резисторы, кнопки, макетная плата и RTC) для создания часов;
- разработана принципиальная схема с использованием ПО Tinkercad;
- отработана программная часть:

  получение данных с часов реального времени RTC;

  преобразование времени с модуля RTC в удобный для индикации формат;

  управление работой светодиодов для бинарной индикации времени;

  сделана возможной установка времени с помощью двух кнопок.

- создан видеоролик, отражающий работу часов (смену состояний светодиодов) с 6 часов 59 минут по 7 часов 00 минут;

- создан видеоролик, на котором демонстрируется установка времени (часов и минут) на часах с помощью кнопок с 9-39 до 6-19;

Список литературы

При работе над проектом были использованы следующие источники информации:

1. http://wiki.amperka.ru - изменяемый пользователями раздел сайта, посвященный элементной базе

2. http://alexgyver.ru - Фото и видео самоделок, Arduino проекты

3. Виктор Петин. Проекты с использованием контроллера Arduino. СПб: БХВ-Петербург, 2015.

4. Улли Соммер. Программирование микроконтроллерных плат Arduino Freeduino. СПб: БХВ-Петербург, 2012.

5. Марк Геддес. 25 крутых проектов с Arduino. Москва: Эксмо, 2019.

6. Виктор Петин. 77 проектов для Arduino. Litres, 2019.

7. Блум Джереми. Изучаем Arduino: инструменты и методы технического волшебства. СПб: БХВ-Петербург, 2015

8. Саймон Монк. Программируем Arduino. СПб: Питер, 2017

9. Салахова А. А., Феоктистова О. А., Александрова Н. А., Храмова М. В. Arduino. Полный учебный курс. От игры к инженерному проекту. Лаборатория знаний, 2020