

**Муниципальное бюджетное общеобразовательное учреждение  
«Яблоневская основная общеобразовательная школа»**

**Муниципальная  
проектно – исследовательская конференция школьников  
«Одиссея разума»**

Направление: физико-математическое  
Секция: техническая

**Проектная работа по теме:  
«Электронные шахматные часы  
на базе микроконтроллера Arduino»**



**Выполнил:** Выползов Илья, ученик 6«А» класса

**Руководитель:** Золотухин Константин Геннадьевич, учитель информатики

**п. Яблоневка**

## Введение

В МБОУ «Яблоневская ООШ» организована работа спортивной секции «Шахматы». Были приобретены несколько шахматных досок и шахматные часы. К сожалению, часы оказались неудобными – падение флажка легко не заметить. Руководитель секции Светлана Олеговна запланировала приобретение шахматных часов, лишенных данного недостатка.

Учитель информатики нашей школы Константин Геннадьевич Золотухин предложил мне создать приложение «Шахматные часы», которое можно будет запускать с компьютера или ноутбука. С этой задачей я успешно справился, представив на прошлой конференции «Одиссея разума» свое приложение для шахматистов.

Во время работы над тем проектом у меня родилась идея создания готового изделия – шахматных часов на базе микроконтроллера «Arduino».

**Цель:** создание электронных шахматных часов базе микроконтроллера «Arduino».

### **Задачи:**

1. Разработать принципиальную схему электронных часов;
2. Собрать часы на макетной плате;
3. Написать программу в системе программирования Arduino IDE;
4. Произвести отладку и тестирование программы;
5. Соединить элементы электрической цепи с помощью пайки;
6. Спроектировать 3D модель корпуса часов в системе КОМПАС 3D;
7. Распечатать корпус на 3D принтере;
8. Собрать часы в корпусе;
9. Протестировать работу часов в реальных условиях.

## **Практическая значимость и ценность проекта**

Благодаря реализации этого проекта, я повышу свой уровень в программировании на языке C++, 3D-моделировании и пайке. Все эти навыки мне пригодятся в моей будущей профессиональной деятельности в качестве инженера электронных систем. Кроме того, используя недорогие компоненты и материалы, моя работа сэкономит для школы около 5400 рублей, которые планировалось потратить на покупку шахматных часов. В отличие от классических механических часов, электронные часы не подвержены износу, гораздо удобнее в настройке и использовании.

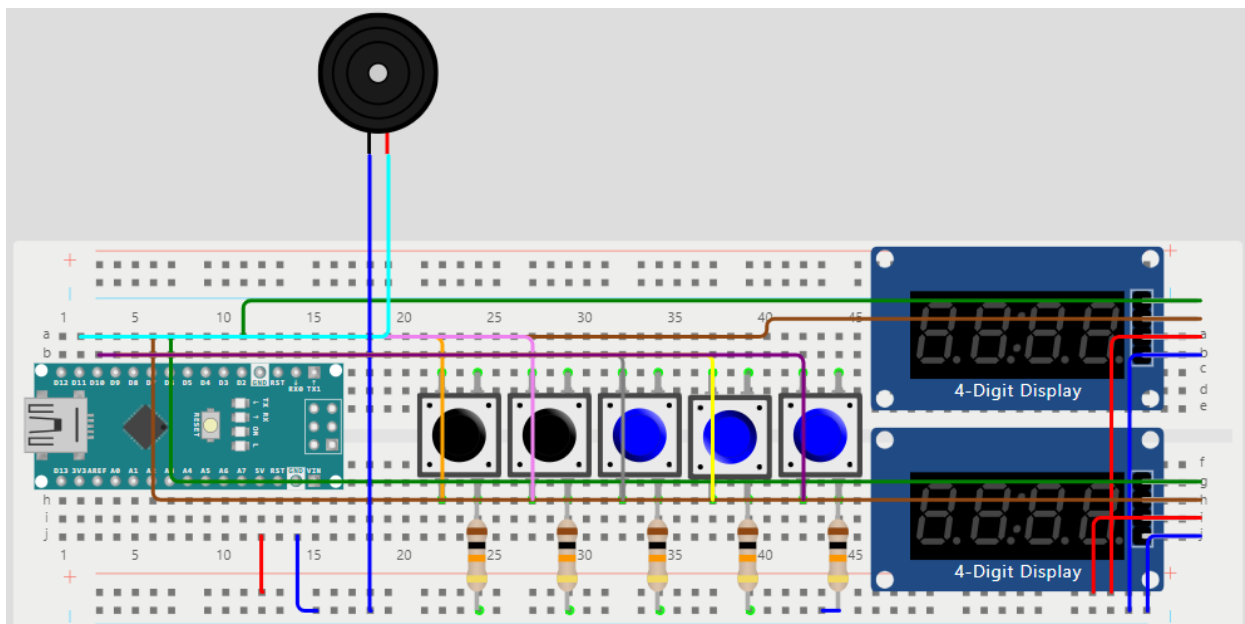
## **Основная часть**

Шахматные часы это прибор, используемый для осуществления контроля времени в шахматах. Классические шахматные часы объединяют в себе два часовых механизма, связанных между собой переключателем, блокирующим ход одного из них. Управление переключателем осуществляется с помощью двух кнопок – по одной для каждого игрока. Сделав ход, игрок нажимает ближайшую к себе кнопку. Часы, показывающие оставшееся для него время останавливаются, а часы соперника запускаются.

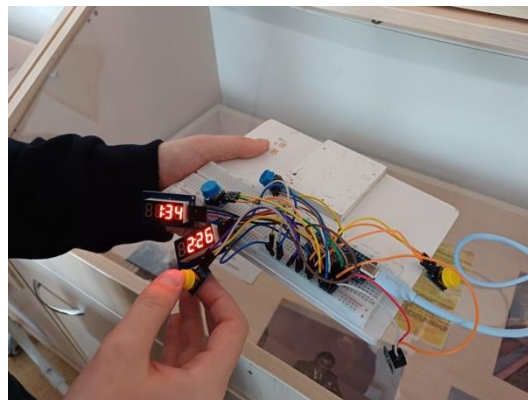
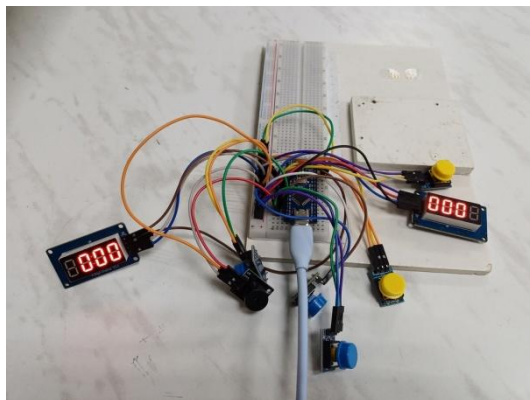
Для изготовления электронных часов понадобится:

1. Микроконтроллер Arduino – 1 шт.;
2. Цифровые дисплеи – 2 шт.;
3. Пьезоэлектрический излучатель (зуммер) – 1 шт.;
4. Кнопки управления – 5 шт.;
5. Выключатель питания – 1 шт.;
6. Резисторы – 5 шт.;
7. Соединительные провода;
8. Элементы питания;
9. Корпус из PLA-пластика для 3D принтера.

## Принципиальная схема электронных часов



## Сборка часов на макетной плате



## Программа в системе программирования Arduino IDE

```

/* V 1.0.0
// Подключение библиотеки для индикаторов
#include "GyverTM1637.h"
// Подключение контактов
#define CLK_1 2 // Подключение контактов 1 ого индикатора
#define DIO_1 3 // Подключение контактов 1 ого индикатора
#define BTN_1 4 // Подключение кнопки для переключения времени
#define BTN_2 5 // Подключение кнопки для переключения времени
#define BTN_3 9 // Подключение кнопки для настройки игры

```

```

#define BTN_4 8 // Подключение кнопки для переключения режима
#define CLK_2 6 // Подключение контактов 2 ого индикатора
#define DIO_2 7 // Подключение контактов 2 ого индикатора
#define zum 10 // Подключение контактов зуммера
#define BTN_5 11 // Подключение кнопки для общей паузы
// Все переменные
int MIN = 0;
int SEC = 0;
int MIN1 = 0;
int SEC1 = 0;
int finish_min = 0;
int finish_sec = 0;
int chess_moves = 0;
bool bt1 = false;
bool bt2 = false;
bool config_mode = true;
bool notBut = true;
GyverTM1637 disp1(CLK_1, DIO_1);
GyverTM1637 disp2(CLK_2, DIO_2);
unsigned long timer1 = 0;
unsigned long timer2 = 0;
// настройка
void setup(){
    pinMode(BTN_1, INPUT);
    pinMode(BTN_2, INPUT);
    pinMode(BTN_3, INPUT);
    pinMode(BTN_4, INPUT);
    pinMode(BTN_5, INPUT);
    pinMode(zum, OUTPUT);
    disp1.clear();
    disp1.brightness(7);
    disp1.clear();
    disp2.clear();
    disp2.brightness(7);
    disp2.clear();
    timer1 = millis();
    timer2 = millis();
}
// Все основные функции
void loop(){
    clockbtn(); // функция для обработки нажатий всех кнопок
    select_mode(); // функция для настройки часов
    chessClock(); // основная функция для таймера
    reset(); // функция для ресета в конце игры

```

```

}
// Функция таймера
void chessClock(){
  if (config_mode == false){
    disp1.displayClock(MIN, SEC);
    disp1.point(1);
    disp2.displayClock(MIN1, SEC1);
    disp2.point(1);
    if ((MIN1 != 0 || SEC1 != 0) && notBut == false){
      if (bt1 == true){
        if (MIN1 != 0){
          if(millis() - timer1 > 1000){
            timer1 = millis();
            SEC--;
          }
          if (SEC == -1){
            SEC = 59;
            MIN--;
          }
        }
      }
    }
    else {
      if (millis() - timer1 > 1000){
        timer1 = millis();
        SEC--;
      }
      if (MIN == 0 && SEC == 0){
        disp1.displayClock(finish_min, finish_sec);
        notBut = true;
        digitalWrite(zum, HIGH);
        delay(500);
        digitalWrite(zum, LOW);
        clockPoint2();
      }
    }
  }
  else{
    disp1.displayClock(MIN, SEC);
  }
}

if ((MIN != 0 || SEC != 0) && notBut == false){
  if (bt2 == true){
    if (MIN1 != 0){
      if(millis() - timer2 > 1000){

```

```

    timer2 = millis();
    SEC1--;
}
if (SEC1 == -1){
    SEC1 = 59;
    MIN1--;
}
}
else {
    if (millis() - timer2 > 1000){
        timer2 = millis();
        SEC1--;
    }
    if (MIN1 == 0 && SEC1 == 0){
        disp2.displayClock(finish_min, finish_sec);
        notBut = true;
        digitalWrite(zum, HIGH);
        delay(500);
        digitalWrite(zum, LOW);
        clockPoint1();
    }
}
}
else{
    disp2.displayClock(MIN1, SEC1);
}
}
}
}
// Функции мерцания точки
void clockPoint1(){
    for (int i = 0; i < 2; i++){
        disp1.point(1);
        delay(1000);
        disp1.point(0);
        delay(1000);
        disp1.point(1);
    }
}
void clockPoint2(){
    for (int i = 0; i < 2; i++){
        disp2.point(1);
        delay(1000);
        disp2.point(0);
    }
}

```

```

    delay(1000);
    disp2.point(1);
  }
}
// Функция для всех кнопок
void clockbtn() {
  if (digitalRead(BTN_5) == HIGH){
    bt1 = false;
    bt2 = false;
  }
  if (digitalRead(BTN_4) == HIGH){
    SEC = SEC + 30;
    delay(200);
  }
  if (digitalRead(BTN_3) == HIGH){
    config_mode = false;
    notBut = false;
  }
  if (digitalRead(BTN_1) == HIGH){
    bt1 = true;
    bt2 = false;
    chess_moves++;
    delay(200);
  }
  if (digitalRead(BTN_2) == HIGH){
    bt1 = false;
    bt2 = true;
    chess_moves++;
    delay(200);
  }
}
// Функция выбора времени
void select_mode(){
  disp1.displayClock(MIN, SEC);
  disp2.displayClock(MIN1, SEC1);
  if (config_mode == true) {
    if (SEC == 60){
      SEC = 0;
      MIN++;
    }
    if (MIN == 60){
      MIN = 0;
      SEC = 0;
    }
  }
}

```



```

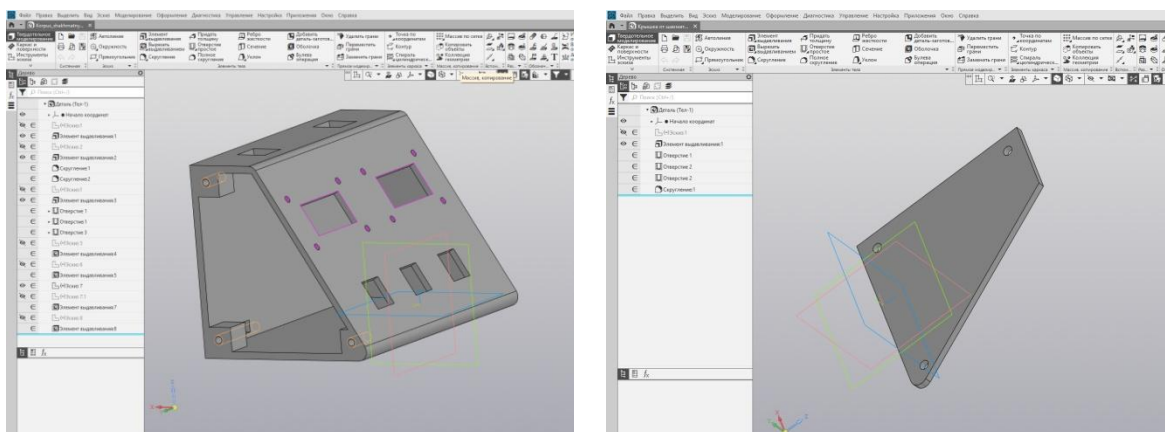
    MIN1 = MIN + 0;
    SEC1 = SEC + 0;
  }
}
// Функция перезагрузки
void reset(){
  if (notBut == true){
    if(digitalRead(BTN_1) == HIGH || digitalRead(BTN_2) == HIGH){
      disp1.clear();
      disp2.clear();
      disp1.point(0);
      disp2.point(0);
      disp1.displayInt(chess_moves);
      disp2.displayInt(chess_moves);
      delay(3000);
      MIN = 0;
      SEC = 0;
      MIN1 = 0;
      SEC1 = 0;
      config_mode = true;
      bt1 = false;
      bt2 = false;
      chess_moves = 0;

    }
  }
}

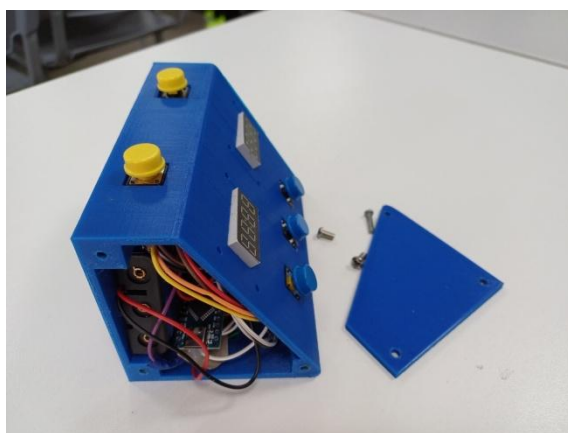
```

### **Создание 3D модели корпуса часов в системе КОМПАС 3D**

Компьютерная модель корпуса часов, состоящая из двух деталей, выполнена в системе автоматизированного проектирования КОМПАС 3D v23 (учебная версия). В процессе моделирования применялись такие операции как: выдавливание, отверстие, скругление. Файлы с 3D моделями были импортированы в формат STL для печати на 3D принтере.



## Сборка часов в корпусе



Для монтажа деталей в корпусе применялись винты и клей, подходящий для пластмассовых деталей.

## Тестирование

Для тестирования часов была выбрана игра в блиц с контролем времени по 3 минуты на игрока. Использование часов не выявило каких-либо проблем и сбоев. Устройство удобно в обращении и имеет привлекательный дизайн.



Проект шахматных часов был опубликован мною на ресурсе github.  
Ссылка: <https://github.com/kommunist448/chess-clock>