Государственное бюджетное общеобразовательное учреждение средняя общеобразовательная школа №422

Кронштадтского района Санкт-Петербурга

Умная лампа

Автор проекта:

**Михайленко Тимофей Николаевич**

Ученик 9 «Б»

Руководитель проекта: **Ненахова Ирина Викторовна,** учитель информатики

2024г.

Оглавление

[Введение 3](#_Toc157846429)

[История появления ламп 4](#_Toc157846430)

[Устройство лампы накаливания 5](#_Toc157846431)

[Поиск аналога 6](#_Toc157846432)

[Подготовительный этап проекта 7](#_Toc157846433)

[Моделирование и сборка устройства 7](#_Toc157846434)

[Заключение 9](#_Toc157846435)

[Список литературы 9](#_Toc157846436)

[Приложение 1. 9](#_Toc157846437)

# Введение

Актуальность темы:

Новые технологии заставляют нас жить по-новому. Плохо это или хорошо пока не понятно. Но многие человеческие действия сейчас могут быть автоматизированы, возможно для того чтобы человек занялся именно тем, что присуще только ему, например, искусством или креативным творчеством. Идея лампы актуальна и популярна как никогда – роботы-пылесосы, роботы, помогающие медикам, космические роботы… Человек всегда мечтает облегчить и украсить себе жизнь. Оригинальность моего проекта – учёт индивидуальных потребностей каждого человека.

Цель:

Создать необычную лампу с доступными составляющими

Реализация проекта потребовала решение следующих задач:

1. Изучение аналогов
2. Изучение аппаратного и программного обеспечения
3. Решение вопросов электропитания устройства
4. Систематизация функционала лампы

# История появления ламп

Первая лампа накаливания изобретена русским электротехником Александром Николаевичем Лодыгиным. В качестве нити накаливания он применил угольный стержень, который поместил в вакуумный сосуд. На свое изобретение летом 1874 года Лодыгин получил патент. Но на этом он не остановился. Александр Николаевич продолжил свои исследования, работая над тем, чтобы использовать тугоплавкие металлы в качестве нити накаливания.

Спустя год, Василию Федоровичу Дидрихсону удалось усовершенствовать лампу Лодыгина, тем самым продлив срок ее службы. Он предложил откачивать воздух из сосуда, а также использовать не одну, а несколько нитей накаливания.

Параллельно с Лодыгиным работу в этом же направлении вел и известный американский изобретатель Томас Эдисон. В своей лампе он использовал платиновую нить накаливания, а в 1879 году запатентовал свое изобретение. Однако такое изделие стоило очень дорого, поэтому не получило широкого распространения. Вернувшись к работе с угольным стержнем, Томас через год создает лампу, работающую в течение сорока часов. Именно Эдисон придумал цоколь и патрон, а спустя некоторое время наладил производство лампочек по цене два с половиной доллара за штуку.

Лодыгин, продолжая работать с тугоплавкими металлами, создает лампочку с вольфрамовой нитью накаливания. В 1906 году компания «*GeneralElectric*» покупает у него патент на это изобретение. Спустя три года сотрудник компании ИрвингЛенгмюр добился увеличения времени работы вольфрамовой нити путем наполнения лампы аргоном. Чуть позже американский физик Уильям Дэвид Кулидж смог усовершенствовать метод изготовления вольфрамовой нити. Все эти изобретения в комплексе позволили лампе с вольфрамовой нитью накаливания постепенно завоевать весь рынок и вытеснить конкурентов.

Устройство лампы накаливания



*Строение и схема лампы накаливания выглядят так:*

* стеклянная колба грушевидной или округлой формы;
* тело накала (вольфрамовая или угольная нить), расположенное в ней на двух держателях-крючках;
* два электрода;
* предохранитель;
* ножка;
* цоколь (корпус) с изолятором;
* его контакт (донышко).

*Рассмотрев, из чего состоит лампочка, важно понять и принцип её работы:*

* При включении света через донышко цоколя к телу накала проходит ток.
* Вольфрамовая нить сильно разогревается после замыкания электрической цепи, что приводит к её свечению.
* На этот момент температура нити достигает 570 градусов.
* Таким образом спектр свечения лампочек сдвинут в сторону теплых температур.

*Для справки:* чем ниже градус вольфрамовой/угольной нити, тем ниже будет доля энергии, которая подходит к телу накала и провоцирует его видимое излучение. Ретро-лампы тем и отличаются, что медленнее и слабее прогревают спираль.

# Поиск аналога

Для того что бы провести анализ и понять стоит ли делать данный проект я решил посмотреть сколько будет стоить купить уже готовую аналогичную лампу:

|  |  |
| --- | --- |
| Пример из интернета | Стоимость, руб. |
| Файл STL Рамка для светодиодной матрицы 12812B 16x16 🔧・3D-печатная модель  для загрузки・Cults | 3000 |
| Светодиодная матрица WS2812B 16x16 адресная SMD5050 RGB, 5В, 256  светодиодов, Черная подложка (ID#1101836243), цена: 575 ₴, купить на Prom.ua | 2500 |
| Лава лампа. Продолжение | Пикабу | 3000 |

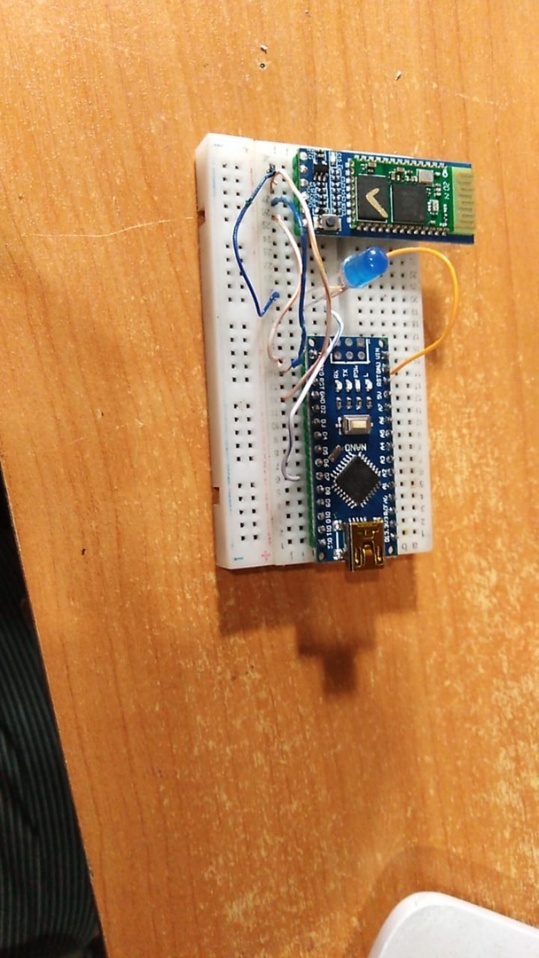
# Подготовительный этап проекта

Я сидел, делал уроки, понял, что мне не хватает освещения и декора на рабочем месте, я решил посмотреть какие-нибудь лампы и что-нибудь для декораций. Мне ничего не понравилось, и я решил сделать что-то своими руками, нашёл в интернете лампу с разными эффектами, которую можно сделать самому. Я нашёл составляющие компоненты, программу, и начал собирать, паять, получил сначала макет, но работал он немного некорректно, начал смотреть в программе, менять настройки, в итоге всё заработало, так как нужно. Потом я сделал лампу уже в том виде, в котором она есть сейчас.

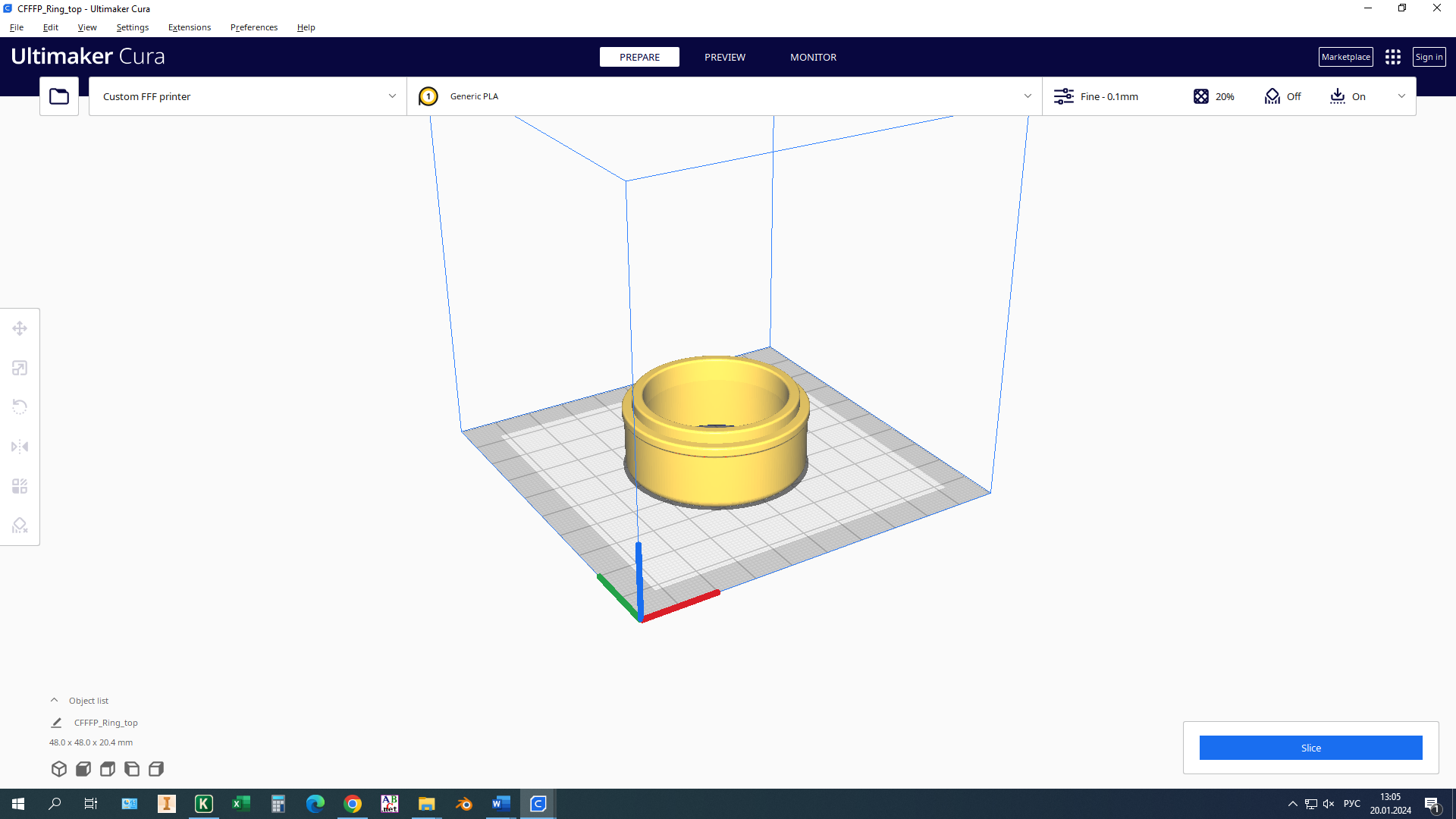
Wi-Fi - технология

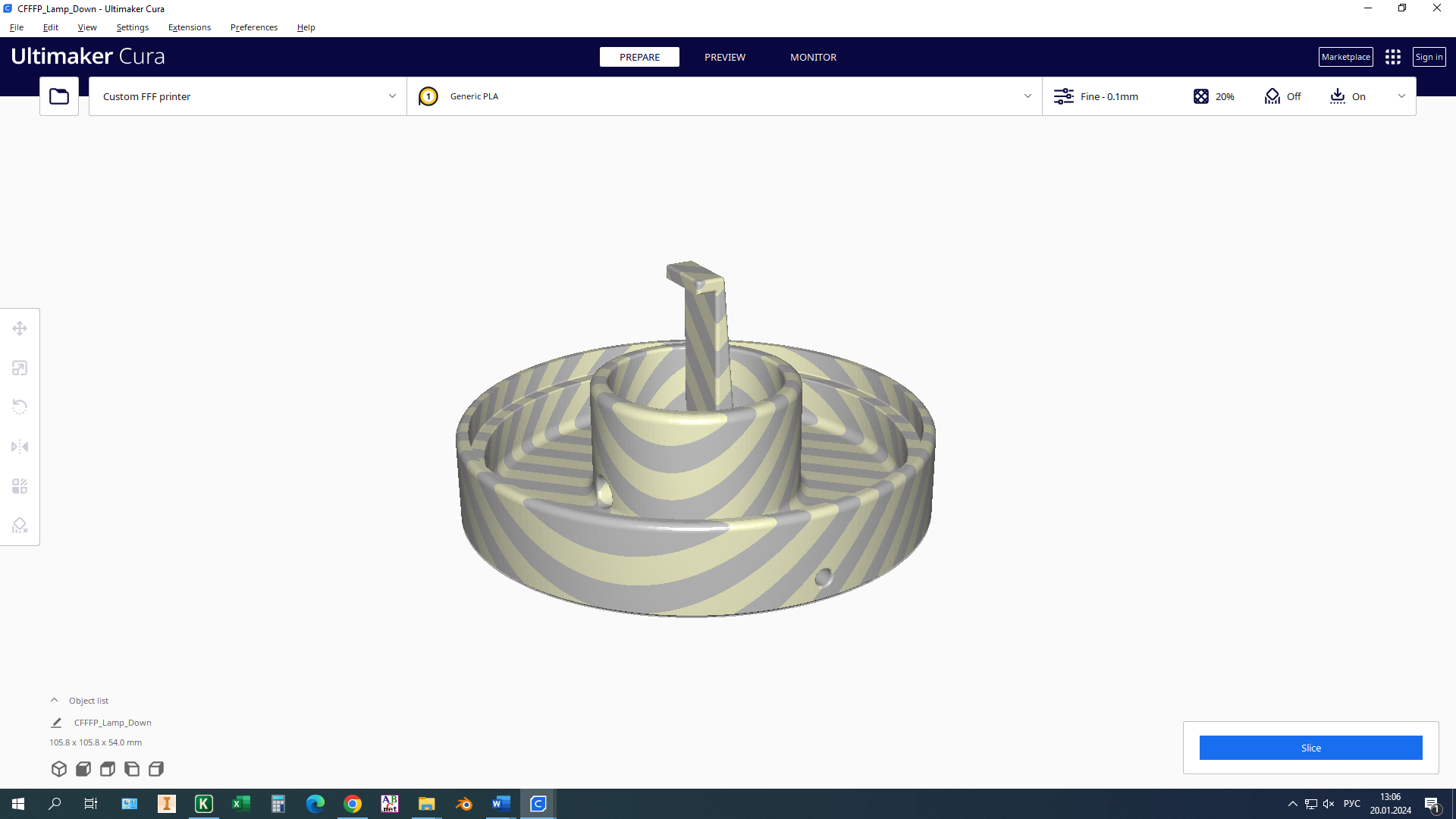
# Моделирование и сборка устройства

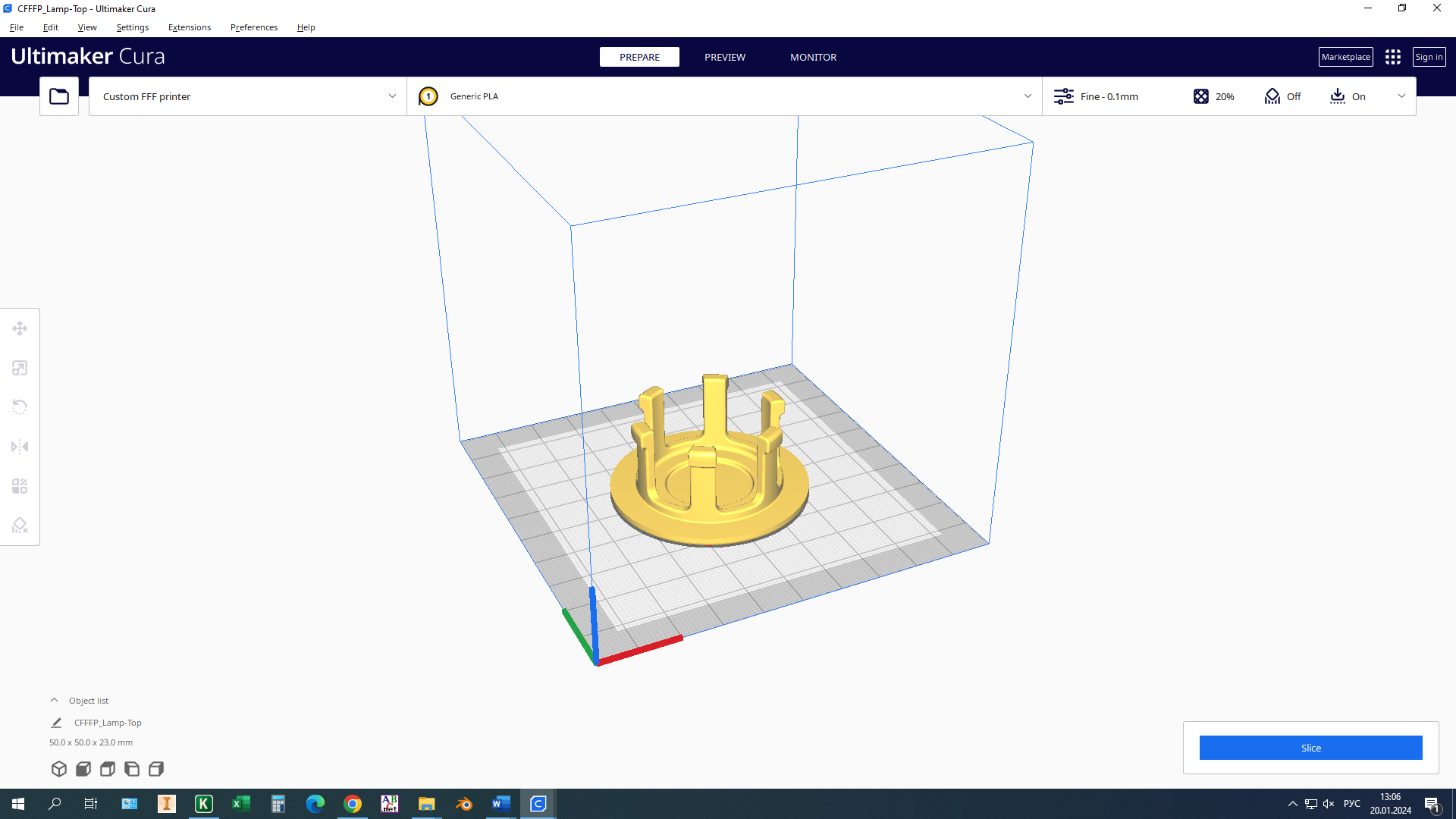
Как мозг своего проекта я решил взять плату Arduino Nano, также матрица 16 на 16 светодиодов, колба для корпуса, рассеиватели и Wi-Fi модуль для подключения к телефону.



Изначально был создан макет, который был запущен неудачно, но после нескольких попыток изменений в коде програмы мы вышли на идеальный результат и собрали его уже, так как он представлен сейчас.







# Заключение

В заключение хочется отметить, что, как правило, построение интеллектуального дома начинается именно с функции управления освещением, поскольку одна эта функция способна значительно преобразить наш опыт пребывания в своем доме.

Проведенный опыт показал, что при должной подготовке, наличии соответствующих знаний и инструментов создание «умного» освещения не составляет большого труда.

Всего лишь за несколько десятков лет наука и техника совершили настолько большой прорыв, что то, что ранее казалось совершенно фантастичным, теперь реализуемо в короткие сроки и доступно каждому.

Работая над этим проектом, я узнал подробнее об аппаратной и программной части платформы Arduino, а также много нового о современных технологиях систем освещения.

# Список литературы

1. Игнатьева, Е. А. Умный дом – технология будущего / Е. А.
2. Игнатьева, С. А Шпаков // Современные наукоемкие технологии.-2013-№7
3. Лапина, А. П. Энергоэффективные технологии / Лапина А. П.
4. Инженерный вестник Дона -2015-№ 1-2.
5. Новый взгляд на умный дом — [Электронный Ресурс].

# Приложение 1.

Для работы лампы был написан такой код:

#include "pgmspace.h"

#include "Constants.h"

#include <FastLED.h>

#include <ESP8266WiFi.h>

#include <ESP8266WebServer.h>

#include <WiFiManager.h>

#include "CaptivePortalManager.h"

#include <WiFiUdp.h>

#include <EEPROM.h>

#include "Types.h"

#include "timerMinim.h"

#ifdef ESP\_USE\_BUTTON

#include <GyverButton.h>

#endif

#include "fonts.h"

#ifdef USE\_NTP

#include <NTPClient.h>

#include <Timezone.h>

#endif

#include <TimeLib.h>

#ifdef OTA

#include "OtaManager.h"

#endif

#if USE\_MQTT

#include "MqttManager.h"

#endif

#include "TimerManager.h"

#include "FavoritesManager.h"

#include "EepromManager.h"

CRGB leds[NUM\_LEDS];

WiFiManager wifiManager;

WiFiServer wifiServer(ESP\_HTTP\_PORT);

WiFiUDP Udp;

#ifdef USE\_NTP

WiFiUDP ntpUDP;

NTPClient timeClient(ntpUDP, NTP\_ADDRESS, 0, NTP\_INTERVAL);

  #ifdef SUMMER\_WINTER\_TIME

  TimeChangeRule summerTime = { SUMMER\_TIMEZONE\_NAME, SUMMER\_WEEK\_NUM, SUMMER\_WEEKDAY, SUMMER\_MONTH, SUMMER\_HOUR, SUMMER\_OFFSET };

  TimeChangeRule winterTime = { WINTER\_TIMEZONE\_NAME, WINTER\_WEEK\_NUM, WINTER\_WEEKDAY, WINTER\_MONTH, WINTER\_HOUR, WINTER\_OFFSET };

  Timezone localTimeZone(summerTime, winterTime);

  #else

  TimeChangeRule localTime = { LOCAL\_TIMEZONE\_NAME, LOCAL\_WEEK\_NUM, LOCAL\_WEEKDAY, LOCAL\_MONTH, LOCAL\_HOUR, LOCAL\_OFFSET };

  Timezone localTimeZone(localTime);

  #endif

#endif

timerMinim timeTimer(3000);

bool ntpServerAddressResolved = false;

bool timeSynched = false;

uint32\_t lastTimePrinted = 0U;

#ifdef ESP\_USE\_BUTTON

GButton touch(BTN\_PIN, LOW\_PULL, NORM\_OPEN);

#endif

#ifdef OTA

OtaManager otaManager(&showWarning);

OtaPhase OtaManager::OtaFlag = OtaPhase::None;

#endif

#if USE\_MQTT

AsyncMqttClient\* mqttClient = NULL;

AsyncMqttClient\* MqttManager::mqttClient = NULL;

char\* MqttManager::mqttServer = NULL;

char\* MqttManager::mqttUser = NULL;

char\* MqttManager::mqttPassword = NULL;

char\* MqttManager::clientId = NULL;

char\* MqttManager::lampInputBuffer = NULL;

char\* MqttManager::topicInput = NULL;

char\* MqttManager::topicOutput = NULL;

bool MqttManager::needToPublish = false;

char MqttManager::mqttBuffer[] = {};

uint32\_t MqttManager::mqttLastConnectingAttempt = 0;

SendCurrentDelegate MqttManager::sendCurrentDelegate = NULL;

#endif

uint16\_t localPort = ESP\_UDP\_PORT;

char packetBuffer[MAX\_UDP\_BUFFER\_SIZE];

char inputBuffer[MAX\_UDP\_BUFFER\_SIZE];

static const uint8\_t maxDim = max(WIDTH, HEIGHT);

ModeType modes[MODE\_AMOUNT];

AlarmType alarms[7];

static const uint8\_t dawnOffsets[] PROGMEM = {5, 10, 15, 20, 25, 30, 40, 50, 60};

uint8\_t dawnMode;

bool dawnFlag = false;

uint32\_t thisTime;

bool manualOff = false;

int8\_t currentMode = 0;

bool loadingFlag = true;

bool ONflag = false;

uint32\_t eepromTimeout;

bool settChanged = false;

bool buttonEnabled = true;

unsigned char matrixValue[8][16];

bool TimerManager::TimerRunning = false;

bool TimerManager::TimerHasFired = false;

uint8\_t TimerManager::TimerOption = 1U;

uint64\_t TimerManager::TimeToFire = 0ULL;

uint8\_t FavoritesManager::FavoritesRunning = 0;

uint16\_t FavoritesManager::Interval = DEFAULT\_FAVORITES\_INTERVAL;

uint16\_t FavoritesManager::Dispersion = DEFAULT\_FAVORITES\_DISPERSION;

uint8\_t FavoritesManager::UseSavedFavoritesRunning = 0;

uint8\_t FavoritesManager::FavoriteModes[MODE\_AMOUNT] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

uint32\_t FavoritesManager::nextModeAt = 0UL;

bool CaptivePortalManager::captivePortalCalled = false;

void setup()

{

  Serial.begin(115200);

  Serial.println();

  ESP.wdtEnable(WDTO\_8S);

  #ifdef MOSFET\_PIN

  pinMode(MOSFET\_PIN, OUTPUT);

  #ifdef MOSFET\_LEVEL

  digitalWrite(MOSFET\_PIN, !MOSFET\_LEVEL);

  #endif

  #endif

  #ifdef ALARM\_PIN

  pinMode(ALARM\_PIN, OUTPUT);

  #ifdef ALARM\_LEVEL

  digitalWrite(ALARM\_PIN, !ALARM\_LEVEL);

  #endif

  #endif

  #if defined(GENERAL\_DEBUG) && GENERAL\_DEBUG\_TELNET

  telnetServer.begin();

  for (uint8\_t i = 0; i < 100; i++)

  {

    handleTelnetClient();

    delay(100);

    ESP.wdtFeed();

  }

  #endif

  #if defined(ESP\_USE\_BUTTON)

  touch.setStepTimeout(BUTTON\_STEP\_TIMEOUT);

  touch.setClickTimeout(BUTTON\_CLICK\_TIMEOUT);

    #if ESP\_RESET\_ON\_START

    delay(1000);

    if (digitalRead(BTN\_PIN))

    {

      wifiManager.resetSettings();

      LOG.println(F("Настройки WiFiManager сброшены"));

    }

    buttonEnabled = true;

    EepromManager::SaveButtonEnabled(&buttonEnabled);

    ESP.wdtFeed();

    #endif

  #endif

  FastLED.addLeds<WS2812B, LED\_PIN, COLOR\_ORDER>(leds, NUM\_LEDS);

  FastLED.setBrightness(BRIGHTNESS);

  if (CURRENT\_LIMIT > 0)

  {

    FastLED.setMaxPowerInVoltsAndMilliamps(5, CURRENT\_LIMIT);

  }

  FastLED.clear();

  FastLED.show();

  EepromManager::InitEepromSettings(

    modes, alarms, &espMode, &ONflag, &dawnMode, &currentMode, &buttonEnabled,

    &(FavoritesManager::ReadFavoritesFromEeprom),

    &(FavoritesManager::SaveFavoritesToEeprom));

  LOG.printf\_P(PSTR("Рабочий режим лампы: ESP\_MODE = %d\n"), espMode);

  wifiManager.setDebugOutput(WIFIMAN\_DEBUG);

  CaptivePortalManager \*captivePortalManager = new CaptivePortalManager(&wifiManager);

  if (espMode == 0U)

  {

    if (sizeof(AP\_STATIC\_IP))

    {

      LOG.println(F("Используется статический IP адрес WiFi точки доступа"));

      wifiManager.setAPStaticIPConfig(

        IPAddress(AP\_STATIC\_IP[0], AP\_STATIC\_IP[1], AP\_STATIC\_IP[2], AP\_STATIC\_IP[3]),

        IPAddress(AP\_STATIC\_IP[0], AP\_STATIC\_IP[1], AP\_STATIC\_IP[2], 1),

        IPAddress(255, 255, 255, 0));

    }

    WiFi.softAP(AP\_NAME, AP\_PASS);

    LOG.println(F("Старт в режиме WiFi точки доступа"));

    LOG.print(F("IP адрес: "));

    LOG.println(WiFi.softAPIP());

    wifiServer.begin();

  }

  else

  {

    LOG.println(F("Старт в режиме WiFi клиента (подключение к роутеру)"));

    if (WiFi.SSID().length())

    {

      LOG.printf\_P(PSTR("Подключение к WiFi сети: %s\n"), WiFi.SSID().c\_str());

      if (sizeof(STA\_STATIC\_IP))

      {

        LOG.print(F("Сконфигурирован статический IP адрес: "));

        LOG.printf\_P(PSTR("%u.%u.%u.%u\n"), STA\_STATIC\_IP[0], STA\_STATIC\_IP[1], STA\_STATIC\_IP[2], STA\_STATIC\_IP[3]);

        wifiManager.setSTAStaticIPConfig(

          IPAddress(STA\_STATIC\_IP[0], STA\_STATIC\_IP[1], STA\_STATIC\_IP[2], STA\_STATIC\_IP[3]),

          IPAddress(STA\_STATIC\_IP[0], STA\_STATIC\_IP[1], STA\_STATIC\_IP[2], 1),

          IPAddress(255, 255, 255, 0));

      }

    }

    else

    {

      LOG.println(F("WiFi сеть не определена, запуск WiFi точки доступа для настройки параметров подключения к WiFi сети..."));

      CaptivePortalManager::captivePortalCalled = true;

      wifiManager.setBreakAfterConfig(true);

      showWarning(CRGB::Yellow, 1000U, 500U);

    }

    wifiManager.setConnectTimeout(ESP\_CONN\_TIMEOUT);

    wifiManager.setConfigPortalTimeout(ESP\_CONF\_TIMEOUT);

    wifiManager.autoConnect(AP\_NAME, AP\_PASS);

    delete captivePortalManager;

    captivePortalManager = NULL;

    if (WiFi.status() != WL\_CONNECTED)

    {

      if (CaptivePortalManager::captivePortalCalled)

      {

        if (millis() < (ESP\_CONN\_TIMEOUT + ESP\_CONF\_TIMEOUT) \* 1000U)

        {

          LOG.println(F("Не удалось подключиться к WiFi сети\nУбедитесь в корректности имени WiFi сети и пароля\nРестарт для запроса нового имени WiFi сети и пароля...\n"));

          wifiManager.resetSettings();

        }

        else

        {

          LOG.println(F("Время ожидания ввода SSID и пароля от WiFi сети или подключения к WiFi сети превышено\nЛампа будет перезагружена в режиме WiFi точки доступа!\n"));

          espMode = (espMode == 0U) ? 1U : 0U;

          EepromManager::SaveEspMode(&espMode);

          LOG.printf\_P(PSTR("Рабочий режим лампы изменён и сохранён в энергонезависимую память\nНовый рабочий режим: ESP\_MODE = %d, %s\nРестарт...\n"),

            espMode, espMode == 0U ? F("WiFi точка доступа") : F("WiFi клиент (подключение к роутеру)"));

        }

      }

      else

      {

        LOG.println(F("Не удалось подключиться к WiFi сети\nВозможно, заданная WiFi сеть больше не доступна\nРестарт для запроса нового имени WiFi сети и пароля...\n"));

        wifiManager.resetSettings();

      }

      showWarning(CRGB::Red, 1000U, 500U);

      ESP.restart();

    }

    if (CaptivePortalManager::captivePortalCalled &&

        sizeof(STA\_STATIC\_IP) &&

        WiFi.localIP() != IPAddress(STA\_STATIC\_IP[0], STA\_STATIC\_IP[1], STA\_STATIC\_IP[2], STA\_STATIC\_IP[3]))

    {

      LOG.println(F("Рестарт для применения заданного статического IP адреса..."));

      delay(100);

      ESP.restart();

    }

    LOG.print(F("IP адрес: "));

    LOG.println(WiFi.localIP());

  }

  ESP.wdtFeed();

  LOG.printf\_P(PSTR("Порт UDP сервера: %u\n"), localPort);

  Udp.begin(localPort);

  #ifdef USE\_NTP

  timeClient.begin();

  ESP.wdtFeed();

  #endif

  #if (USE\_MQTT)

  if (espMode == 1U)

  {

    mqttClient = new AsyncMqttClient();

    MqttManager::setupMqtt(mqttClient, inputBuffer, &sendCurrent);

  }

  ESP.wdtFeed();

  #endif

  memset(matrixValue, 0, sizeof(matrixValue));

  randomSeed(micros());

  changePower();

  loadingFlag = true;

}

void loop()

{

  parseUDP();

  effectsTick();

  EepromManager::HandleEepromTick(&settChanged, &eepromTimeout, &ONflag,

    &currentMode, modes, &(FavoritesManager::SaveFavoritesToEeprom));

  #ifdef USE\_NTP

  timeTick();

  #endif

  #ifdef ESP\_USE\_BUTTON

  if (buttonEnabled)

  {

    buttonTick();

  }

  #endif

  #ifdef OTA

  otaManager.HandleOtaUpdate();

  #endif

  TimerManager::HandleTimer(&ONflag, &settChanged,

    &eepromTimeout, &changePower);

  if (FavoritesManager::HandleFavorites(

      &ONflag,

      &currentMode,

      &loadingFlag

      #ifdef USE\_NTP

      , &dawnFlag

      #endif

      ))

  {

    FastLED.setBrightness(modes[currentMode].Brightness);

    FastLED.clear();

    delay(1);

  }

  #if USE\_MQTT

  if (espMode == 1U && mqttClient && WiFi.isConnected() && !mqttClient->connected())

  {

    MqttManager::mqttConnect();

    MqttManager::needToPublish = true;

  }

  if (MqttManager::needToPublish)

  {

    if (strlen(inputBuffer) > 0)

    {

      processInputBuffer(inputBuffer, MqttManager::mqttBuffer, true);

    }

    MqttManager::publishState();

  }

  #endif

  #if defined(GENERAL\_DEBUG) && GENERAL\_DEBUG\_TELNET

  handleTelnetClient();

  #endif

  ESP.wdtFeed();

}