



Управление образования города Пензы
МБОУ лицей №73 г. Пензы
«Лицей информационных систем и технологий»

Секция: Computer science

Разработка программного обеспечения для функционирования Автоматизированного Заменителя Масок (АЗМ)

Работу выполнили:
Шмаранов Алексей, ученик 6Б

Научный руководитель:
учитель технологии
Пеганов Станислав Юрьевич

Пенза 2021

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
Цель:	4
Задачи:	4
Актуальность	4
I. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ	5
ПРИНЦИП РАБОТЫ УЛЬТРАФИОЛЕТОВЫХ БАКТЕРИЦИДНЫХ РЕЦИРКУЛЯТОРОВ ВОЗДУХА	5
ПРОГРАММИРОВАНИЕ МИКРОКОНТРОЛЛЕРОВ СЕМЕЙСТВА ARDUINO	6
II. ПРАКТИЧЕСКАЯ ЧАСТЬ	8
НАЗНАЧЕНИЕ	8
СОСТАВ УСТРОЙСТВА	8
ОПИСАНИЕ УСТРОЙСТВА	9
ФУНКЦИОНАЛЬНАЯ СХЕМА	10
АЛГОРИТМ РАБОТЫ УСТРОЙСТВА	11
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ РАБОТЫ МИКРОКОНТРОЛЛЕРА	12
НАЗНАЧЕНИЕ ПОЛЬЗОВАТЕЛЬСКИХ ФУНКЦИЙ	12
ТЕСТИРОВАНИЕ УСТРОЙСТВА	15
ЗАКЛЮЧЕНИЕ	15
ЛИТЕРАТУРА	15
Приложение №1	16

ВВЕДЕНИЕ

Миллионы людей во всем мире ежегодно заболевают вирусными заболеваниями. Наиболее тяжелые формы наблюдаются, в первую очередь, у стариков и детей до года. Эта инфекция наносит огромный ущерб здоровью населения и иногда приводит к серьезным осложнениям. От гриппа, COVID-19 и других, а также сопутствующих им осложнений ежегодно умирают сотни людей. Ежегодные эпидемические вспышки вирусных заболеваний объясняются именно высоким уровнем изменчивости вируса.

Во время чихания, так же как и при кашле, изо рта больного человека вылетают мельчайшие частицы слюны и мокроты, в которых вирусы содержатся в огромных количествах. Поэтому основной механизм передачи гриппа так и называется — воздушно-капельный.

Одним способов защиты от распространения таких инфекций, является применение медицинских масок.

Применение медицинских масок имеет не только положительные стороны, которых несомненно больше, но и отрицательные. Как показали медицинские исследования, неправильное и длительное использование медицинских масок может привести к возникновению таких болезней как: как эмфизема легких, бронхиальная астма, грибковые, аллергические, бактериальные заболевания.

Я предлагаю заменить ношение медицинской маски, применением носимого автономного ультрафиолетового бактерицидного рециркулятора-облучателя.

Обеззараживание воздуха происходит в процессе принудительной рециркуляции его с помощью вентилятора через корпус облучателя, внутри которого размещены ультрафиолетовые светодиоды с длиной волны 230-300 нм, при этой длине волны ультрафиолет наиболее эффективно уничтожает РНК и ДНК вирусов и бактерий.

Цель:

1. Разработать модель носимого автономного бактерицидного рециркулятора-облучателя (Автономный Заменитель Масок (АЗМ)) для обеспечения потока обеззараженного воздуха перед лицом человека.
2. Создать программное обеспечение для автоматизированного управления и функционирования Автономного Заменителя Масок (АЗМ), с максимально удобным управлением режимами работы и индикации.

Задачи:

1. Изучить принципы работы бактерицидных рециркуляторов-облучателей.
2. Разработать программное обеспечение управления и индикации.
3. Собрать экспериментальный образец.
4. Протестировать данное устройство.

Актуальность

Обеззараживание воздуха ультрафиолетовыми бактерицидными рециркуляторами является одним из самых действенных методов профилактики заражения инфекционными заболеваниями распространяющихся воздушным путем.

В связи с масштабным распространением COVID-19 и широким применением медицинских масок возникает очень много недовольства их применением. Я, предлагаю таким людям совершенно новое устройство Автономный Заменитель Масок. Анализ информации в сети Интернет не дал никаких сведений о наличии и разработки таких устройств.

Созданный прибор является аналогом бактерицидного очистителя воздуха, только носимого постоянно с собой. Для создания такого прибор не потребовалось больших финансовых затрат, себестоимость стоимость устройства составляет менее 2000 руб.

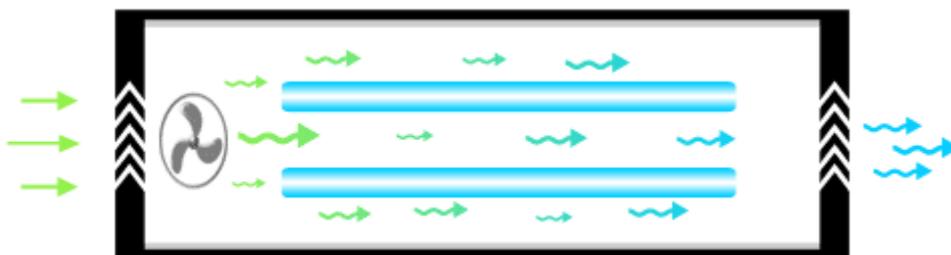
I. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.

ПРИНЦИП РАБОТЫ УЛЬТРАФИОЛЕТОВЫХ БАКТЕРИЦИДНЫХ РЕЦИРКУЛЯТОРОВ ВОЗДУХА

Ультрафиолетовое (УФ) бактерицидное облучение воздушной среды помещений — традиционное и наиболее распространенное санитарно-противоэпидемическое (профилактическое) мероприятие, направленное на снижение количества микроорганизмов в воздухе медицинских организаций и профилактику инфекционных заболеваний.

Спектральный состав УФ-излучения, вызывающего бактерицидное действие, лежит в интервале длин волн 205–315 нм. Вирусы и бактерии в вегетативной форме более чувствительны к воздействию УФ-излучения, чем плесневые и дрожжевые грибы, споровые формы бактерий.

Принцип работы бактерицидного рециркулятора закрытого типа достаточно прост. Внутри корпуса рециркулятора установлены вентиляторы и безозоновые бактерицидные лампы. Вентиляторы нагнетают поток воздуха, который поступает внутрь рециркулятора через вентиляционные отверстия. Далее воздух, облучаясь бактерицидными лампами, обеззараживается и выбрасывается наружу через аналогичные вентиляционные отверстия.



В своем устройстве, я заменил безозоновые бактерицидные лампы, которые требуют напряжения питания 220 В, на ультрафиолетовые светодиоды. Они также являются источниками УФ излучения в диапазоне от 100 до 400 нм. И, могут эффективно использоваться для стерилизации в медицине, в косметологии, при проведении судебно-медицинских экспертиз, изменения состояния материалов и дезинфекции воды.

ПРОГРАММИРОВАНИЕ МИКРОКОНТРОЛЛЕРОВ СЕМЕЙСТВА ARDUINO

Основой моего устройства является плата Arduino Nano, сердцем которого является микроконтроллер ATmega328.

В основном создание программ для данных микроконтроллеров производится в оболочке Arduino IDE.

Arduino IDE - интегрированная среда разработки для Windows, MacOS и Linux, предназначенная для создания и загрузки программ на Arduino-совместимые платы. Язык программирования, который используется для создания программ, несколько упрощенный Си и C++.

Структура программы в Arduino IDE строго формализована и состоит из функций.

Есть 2 обязательные функции, которые присутствуют в любой программе для Arduino: `setup()` и `loop()`.

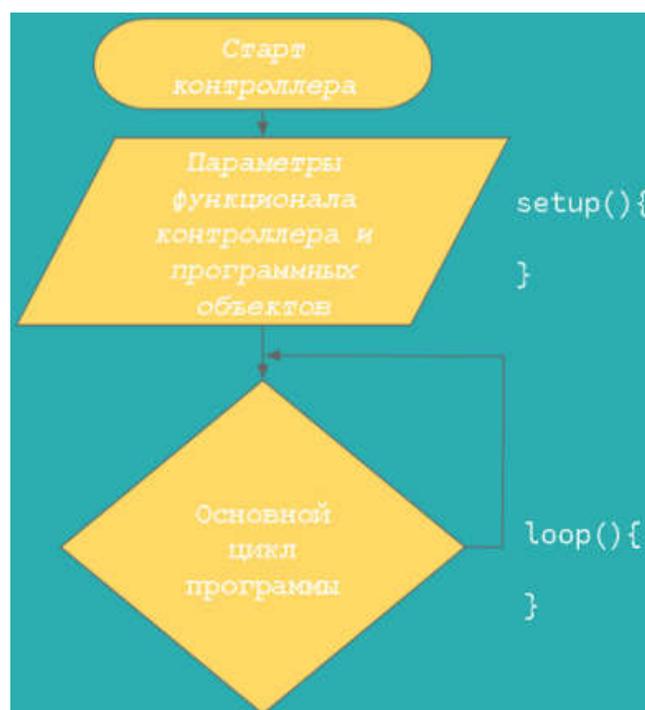
Программист может создавать собственные функции, которые могут использоваться как внутри функции `loop()`, так и внутри других функций.

`Setup` означает настройка. Прямое назначение функции `setup()` состоит в том, чтобы в ней задавались первоначальные параметры работы микроконтроллера и программных объектов, используемых в остальных частях программы.

Слово `loop` означает повторяющийся цикл – закичивание выполнения набора выражений, находящихся в этой функции. Назначение функции `loop()` в том, чтобы хранить выражения основного цикла программы контроллера.

Код написанный внутри функции `setup()` выполняется один раз, при старте работы микроконтроллера или при перезагрузке, код находящийся внутри функции `loop()` выполняется последовательно бесконечно.

Функции созданные программистом, могут быть нескольких видов. Созданную функцию необходимо задекларировать вне скобок любой другой функции, таким образом, она может стоять выше или ниже функции "`loop()`".



Например:

```
int filter()
{
  uint16_t sum = 0;
  for (byte i = 0; i < 16; i++)
    sum += analogRead(0);
  return (sum >> 4); // деление на 16
}
```

или:

```
void manualmode() {
  while (1) {
    if (millis() - tmr2 >= 10000) {
      battery();
      tmr2 = millis();
    }
    sleep(false); // сон на 128 мс
  }
}
```

Результатом вызова функции `filter()` в данном случае возврат подсчитанного целого числа в результате измерения величины напряжения аккумуляторной батареи Аналого-Цифровым Преобразователем платы.

Функция `manualmode()` при ее вызове последовательно запускает операторы и функции, которые прописаны внутри ее.

Разделения кода на функции имеет ряд преимуществ:

- Функции позволяют организовать программу. Очень часто помогают заранее составить концепцию программы.
- Функции кодируют одно действие в одном месте программы. Далее необходимо только отладить код функции.
- Функции сокращают шансы на появление ошибки при необходимости изменения кода.
- Функции сокращают текст скетчей и делают его компактным, т.к. некоторые секции используются много раз.
- Функции облегчают использование кода в других программах делая его модульным. В этом случае функции обладают еще одним небольшим преимуществом, делая код программы легким для чтения.

II. ПРАКТИЧЕСКАЯ ЧАСТЬ

НАЗНАЧЕНИЕ

Автономный Заменитель Масок защищает от различного рода инфекций распространяющихся воздушным путем, таких как грипп, ОРЗ, дифтерия, туберкулез, COVID-19 и многих других путем создания вертикального потока обеззараженного воздуха перед лицом человека.

Автономный Заменитель Масок обеспечивает:

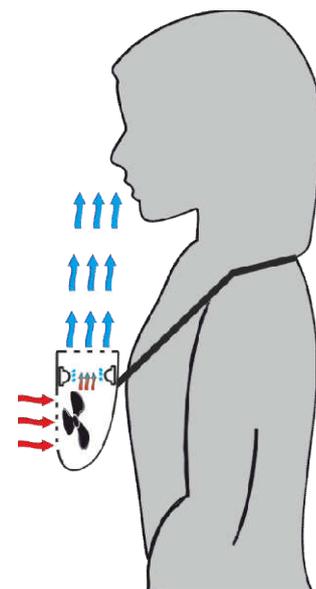
1. Бактерицидную обработку воздуха, путем его прогона через пространство, облучаемое ультрафиолетовыми светодиодами.



2. Мобильность применения.

3. Работу от аккумуляторной батареи 12 В.

4. Автоматизированный и ручной режим работы.



СОСТАВ УСТРОЙСТВА

1. Пластиковый корпус.
2. 2 модуля ультрафиолетовых светодиодов.
3. Компьютерные вентилятор (кулер) 12V.
4. Кнопочный выключатель.
5. Аккумуляторная батарея Li-Po 11,4В 1800 мАч.
6. Лазерный дальномер VL53L0X.
7. Полевой транзистор IRF3205.
8. Arduino Nano.

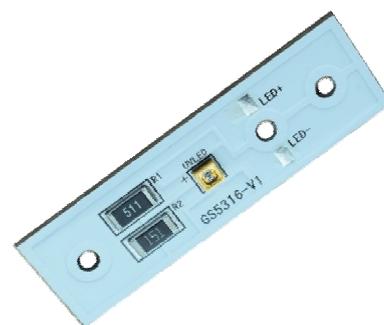
Модуль ультрафиолетовых светодиодов:

Напряжение питания 12В.

Мощность: 0,5 Вт.

Испускаемая цвет: фиолетовый UV (275-395 нм)

Сила света: 2-5 мВт



Лазерный дальномер VL53L0X.

Рабочее напряжение: 3,3 В / 5 В

Расстояние: от 30 до 2000 мм

Точность определения дальности: $\pm 5\%$ (режим высокой скорости), $\pm 3\%$ (режим высокой точности)

Время ранжирования (мин): 20 мс (режим высокой скорости), 200 мс (режим высокой точности)

Угол обзора: 25°

Длина волны лазера: 940 нм



Полевой транзистор IRF3205

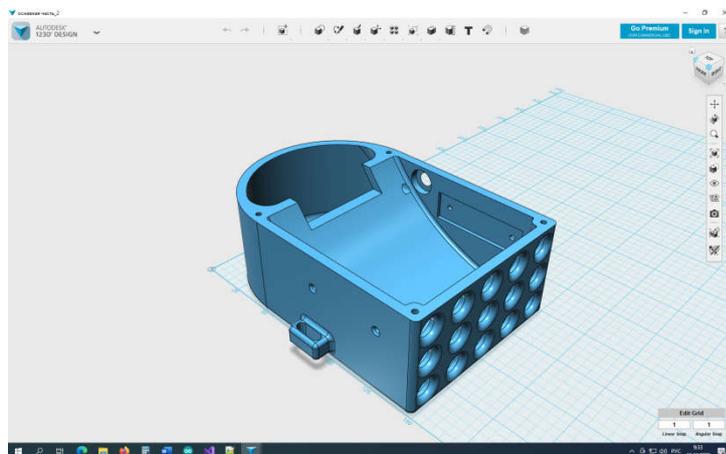
Максимальное напряжение сток-исток $U_{си}$ 55В

Максимальный ток сток-исток при 25 С $I_{си}$ макс. 110 А

Максимальное напряжение затвор-исток $U_{зи}$ макс. ± 20 В

Корпус

Корпус данного устройства был напечатан на 3D-принтере. Проектирование корпуса производилось в программе 123DDesign.



ОПИСАНИЕ УСТРОЙСТВА

Основой данного автономного устройства являются малогабаритный компьютерный вентилятор и 2 модуля ультрафиолетовых светодиодов.

Устройство обеспечивает работу в двух режимах автоматическом или ручном.

Выбор режима осуществляется при включении питания устройства. Если включить устройство с ладонью, поднесенной к лазерному дальномеру, то оно продолжит работу в ручном режиме. Если при включении, перед лазерным даль-

номером не было каких либо препятствий, то устройство включится в автоматизированном режиме.

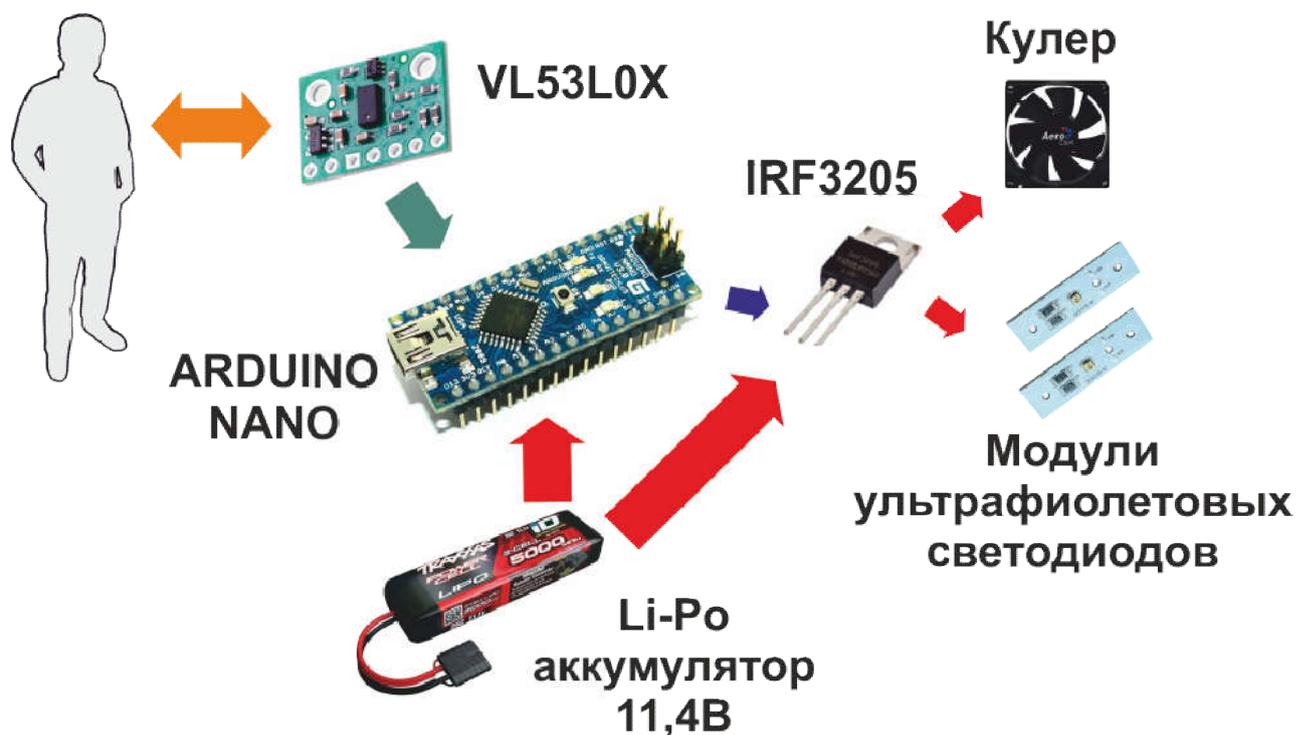
Автоматизированный режим

Данный режим в основном предназначен для использования вне помещений. При обнаружении человека впереди себя на расстоянии менее 2м, устройство автоматически включается и продолжает работать не менее 5 минут.

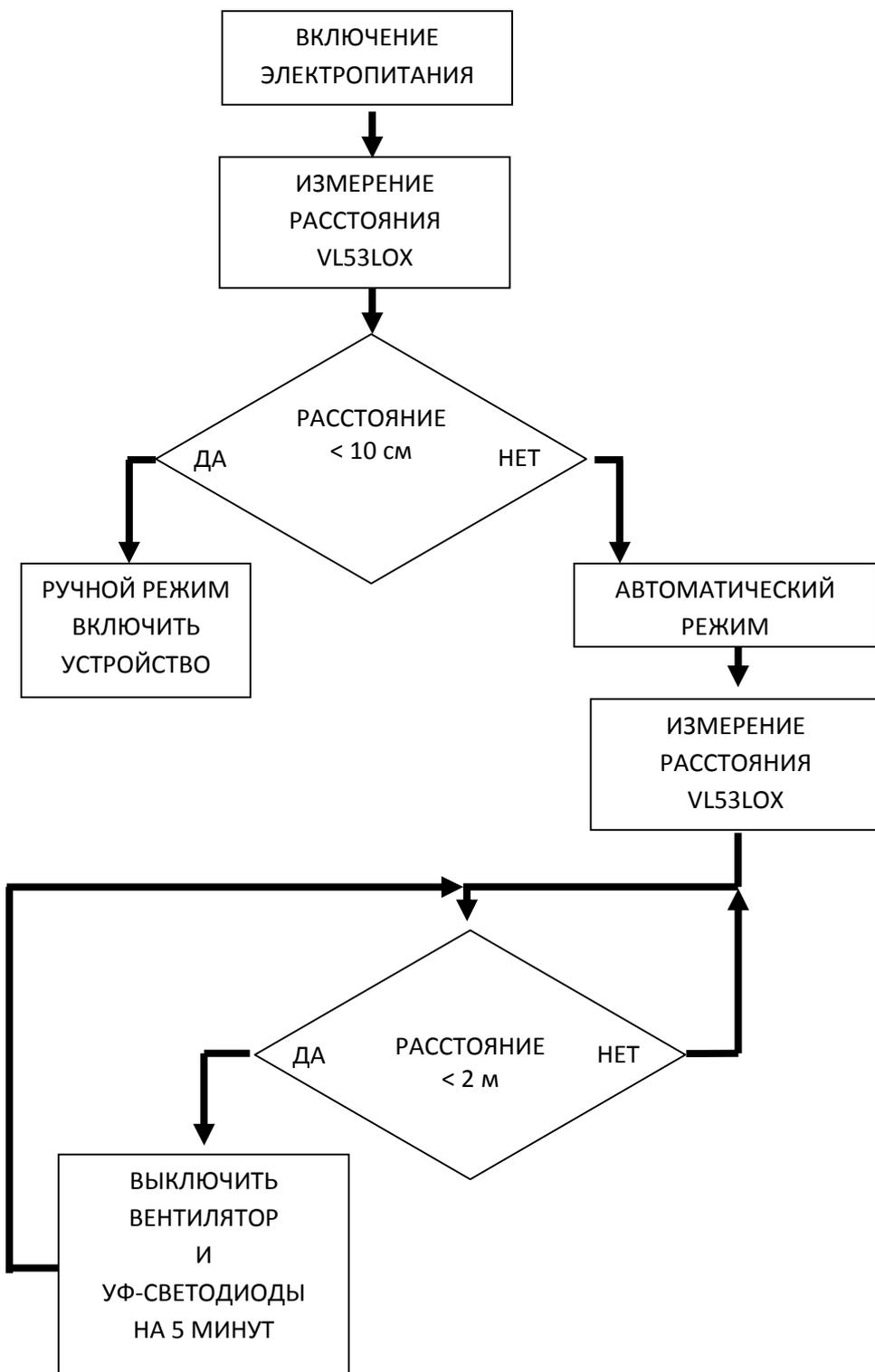
Ручной режим

Данный режим предназначен для использования внутри помещений, после перехода в данный режим устройство постоянно работает. Выключение производится нажатием на кнопку отключения питания.

ФУНКЦИОНАЛЬНАЯ СХЕМА



АЛГОРИТМ РАБОТЫ УСТРОЙСТВА



ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ РАБОТЫ МИКРОКОНТРОЛЛЕРА

Полный листинг программы приведен в Приложении №1.

Программное обеспечение АЗМ обеспечивает:

1. Выбор автоматизированного и ручного режима работы путем опроса датчика VL53L0X в момент включения.
2. Работу в автоматизированном режиме путем измерения датчиком VL53L0X расстояния до впереди расположенного человека(предмета).
3. Управления электронным ключом на основе полевого транзистора IRF3205 и включения вентилятора и светодиодов.
4. Контроля заряда встроенной аккумуляторной батареи, путем измерения ее напряжения на одном из аналоговых портов АЦП платы Arduino Nano.
5. Подачу сигнала разряда аккумуляторной батареи, путем изменения частоты вращения вентилятора, и подачи звукового сигнала самим вентилятором.

НАЗНАЧЕНИЕ ПОЛЬЗОВАТЕЛЬСКИХ ФУНКЦИЙ

ФУНКЦИЯ	НАЗНАЧЕНИЕ
<pre>void automode() { while (1) { if (sensor.readRangeSingleMillimeters() < 1500) { digitalWrite(pin_fan, HIGH); fan = true; tmr = millis(); } if (millis() - tmr >= 300000 && fan) { digitalWrite(pin_fan, LOW); fan = false; tmr = millis(); } if (millis() - tmr2 >= 10000) { battery(); tmr2 = millis(); } sleep(false); // сон на 128 мс } }</pre>	<p>Работа устройства в автоматическом режиме.</p> <p>Бесконечный цикл.</p> <p>Если расстояние до человека менее 2 м, то Включить вентилятор и светодиоды.</p> <p>Каждые 10 секунд вызов функции контроля заряда батареи.</p> <p>Проверка напряжения на аккумуляторе.</p> <p>Сброс таймера.</p> <p>Сон на 128 мс</p>

ФУНКЦИЯ	НАЗНАЧЕНИЕ
<pre>void manualmode() { while (1) { if (millis() - tmr2 >= 10000) { battery(); tmr2 = millis(); } sleep(false); // сон на 128 мс } }</pre>	<p>Работа устройства в ручном режиме. Каждые 10 секунд вызов функции контроля заряда батареи. Проверка напряжения на аккумуляторе. Сброс таймера. Сон на 128 мс.</p>
<pre>void battery() { voltage = (float)filter() * VREF * c / 1024; if (voltage < 11.5) { if (voltage < 11.1) { digitalWrite(pin_fan, LOW); tone(pin_fan, 880); delay(3000); noTone(pin_fan); digitalWrite(pin_fan, LOW); sleep(true); // бесконечный сон } if (millis() - tmr >= 150000) { digitalWrite(pin_fan, LOW); tone(pin_fan, 880); delay(500); tone(pin_fan, 880); delay(500); noTone(pin_fan); digitalWrite(pin_fan, fan); tmr = millis(); } } }</pre>	<p>Контроль заряда батареи. Вызов функции filter() для получения значения напряжения аккумуляторной батареи. Если напряжение меньше 11.5в, то. И если напряжение меньше 11.1в, то. Выключить вентилятор и светодиоды. Пищать вентилятором. Ждать 3 секунды. Прекратить пищать. Выключить вентилятор и светодиоды. Бесконечныйсон</p> <p>Каждые 2.5 минуты сигнализировать о разряде. Выключить вентилятор и светодиоды. Пищать вентилятором. Ждать 0.5 секунды. Прекратить пищать. Ждать 0.5 секунды. Пищать вентилятором. Ждать 0.5 секунды. Прекратить пищать. Если винтелятор был включен – включить. Сброс таймера.</p>
<pre>int filter() { uint16_t sum = 0; for (byte i = 0; i < 16; i++) sum += analogRead(0); return (sum >> 4); // деление на 16 }</pre>	<p>Измерение величины напряжения аккумуляторной батареи на порту А0, внутренним АЦП платы. Найти среднее арифметическое с полученной величины. Цикл на кол-во усреднений. Суммирование полученных значений. Деление на 16 (кол-во усреднений).</p>

ФУНКЦИЯ	НАЗНАЧЕНИЕ
<pre> void sleep (bool _loop) { ADCSRA &= ~(1 << ADEN); ACSR = (1 << ACD); // выключениеАЦП if (!_loop) { // настройка WDT uint8_t wdtReg = (1 << WDIE); wdtReg = (1 << WDP0) (1 << WDP1); // 0.125s cli(); WDTCR = (1 << WDCE) (1 << WDE); // разрешаем вмешательство WDTCR = wdtReg; sei(); asm ("wdr"); // сброс } SMCR = (1 << SM1) (1 << SE); // сон power-down, sleep enable MCUCR = (1 << 5); MCUCR = (1 << 6); //выключение BOD asm("sleep"); // выход из сна SMCR = 0; timer0_millis += 128; // корректировка millis ADCSRA = (1 << ADEN); ACSR &= ~(1 << ACD); // включение АЦП } </pre>	<p>Функция сна. Включение-Выключение АЦП. Настройка и включение прерываний WDT. Прерывания каждые 125мс</p>
<pre> void setup() { pinMode(pin_fan, OUTPUT); Wire.begin(); if (!sensor.init()) sleep(true); // бесконечный сон sensor.setTimeout(500); sensor.setMeasurementTimingBudget(200000); // высокая точность #ifdef (DEBUG_ENABLE) Serial.begin(9600); #endif if (sensor.readRangeSingleMillimeters() < 50) { modes = MANUAL; digitalWrite(pin_fan, HIGH); fan = true; } else { modes = AUTO; digitalWrite(pin_fan, HIGH); delay(500); digitalWrite(pin_fan, LOW); fan = false; } PRR = (1 << PRSPI) (1 << PRTIM1); // выключение SPI и 1 Timer } </pre>	<p>Инициализация платы и пинов. Инициализация шины I2C. Инициализация датчика VL53L0X. Задание режима высокой точности измерения датчика. Выбор режимов работы.</p>

ТЕСТИРОВАНИЕ УСТРОЙСТВА

Тестирование устройства производилось мною лично.

Время непрерывной работы в ручном режиме составило порядка 6-8 часов.

При работе в автоматизированном режиме вне помещения зависело от температуры окружающего воздуха:

При температуре: 0°C - -5°C – 2-3 часа.

При температуре -5°C - -10°C – 1-1,5 часа

ЗАКЛЮЧЕНИЕ

Разработанный Автономный Заменитель Масок, обеспечивает полноценную локальную противовирусную и антимикробную обработку воздуха в непосредственной близости от человека, что позволяет снизить риск заражения различными заболеваниями, в особенности в настоящее время в период увеличения распространения заражения COVID-19.

При этом отпадает необходимость в постоянном ношении маски, что для отдельных людей является достаточно большим преимуществом.

Устройство требует усовершенствования в плане снижения своей массы и усовершенствования режимов работы.

ЛИТЕРАТУРА

1. <https://ru.wikipedia.org/wiki>
2. <http://sovet-ingenera.com/vent/cond/bezlopastnoj-ventilyator.html>
3. <https://aqua-rmnt.com/ventilyaciya/kak-mozhno-sdelat-kondicioner-svoimi-rukami-v-domashnix-usloviyah.html>
4. <https://www.philohome.com/nxtspotlight/spotlight.htm>

Приложение №1

```

#include <Wire.h>
#include <VL53L0X.h>
VL53L0X sensor;

#define pin_fan 9
extern volatile unsigned long timer0_millis;

enum {
  AUTO,
  MANUAL,
} modes;

bool fan;
uint32_t tmr;
uint32_t tmr2;
#define VREF 4.98 // точное напряжение на пине 5V (в данном случае зависит от стабилизатора на
плате Arduino)
#define DIV_R1 9850 // точное значение 10 кОм резистора
#define DIV_R2 4620 // точное значение 4.7 кОм резистора
float voltage;
float c = (DIV_R1 + DIV_R2) / DIV_R2;
#define DEBUG_ENABLE 0

#if (DEBUG_ENABLE == 1)
#define DEBUG(x) Serial.println(x);
#else
#define DEBUG(x)
#endif

void setup() {
  pinMode(pin_fan, OUTPUT);
  Wire.begin();
  if (!sensor.init()) sleep(true); // бесконечный сон
  sensor.setTimeout(500);
  sensor.setMeasurementTimingBudget(200000); // высокая точность
#if (DEBUG_ENABLE)
  Serial.begin(9600);
#endif
  if (sensor.readRangeSingleMillimeters() < 50) {
    modes = MANUAL;
    digitalWrite(pin_fan, HIGH);
    fan = true;
  }
  else {
    modes = AUTO;
    digitalWrite(pin_fan, HIGH);
    delay(500);
    digitalWrite(pin_fan, LOW);
    fan = false;
  }
  PRR |= (1 << PRSPI) | (1 << PRTIM1); // выключение SPI и 1 Timer
}

```

```

void loop() {
  if (modes == MANUAL) manualmode();
  else automode();
}

void sleep (bool _loop) {
  ADCSRA &= ~(1 << ADEN);
  ACSR |= (1 << ACD); // выключение АЦП
  if (!_loop) {
    // настройка WDT
    uint8_t wdtReg = (1 << WDIE);
    wdtReg |= (1 << WDP0) | (1 << WDP1); // 0.125s
    cli();
    WDTCSR |= (1 << WDCE) | (1 << WDE); // разрешаем вмешательство
    WDTCSR = wdtReg;
    sei();
    asm ("wdr"); // сброс
  }

  SMCR |= (1 << SM1) | (1 << SE); // сон power-down, sleep enable

  MCUCR = (1 << 5);
  MCUCR = (1 << 6); //выключение BOD

  asm("sleep");

  // выход из сна
  SMCR = 0;
  timer0_millis += 128; // корректировка millis
  ADCSRA |= (1 << ADEN);
  ACSR &= ~(1 << ACD); // включение АЦП
}

ISR(WDT_vect) { // просыпаемся тут
  WDTCSR |= (1 << WDCE) | (1 << WDE); // разрешаем вмешательство
  WDTCSR = 0; // выключаем
  asm ("wdr"); // обнуляем
}

void automode() {
  while (1) {
    if (sensor.readRangeSingleMillimeters() < 1500) {
      digitalWrite(pin_fan, HIGH);
      fan = true;
      tmr = millis();
    }
    if (millis() - tmr >= 300000 && fan) {
      digitalWrite(pin_fan, LOW);
      fan = false;
      tmr = millis();
    }
  }
}

```

```

    if (millis() - tmr2 >= 10000) {
        battery();
        tmr2 = millis();
    }
    sleep(false); // сон на 128 мс
}
}
void manualmode() {
    while (1) {
        if (millis() - tmr2 >= 10000) {
            battery();
            tmr2 = millis();
        }
        sleep(false); // сон на 128 мс
    }
}

void battery() {
    voltage = (float)filter() * VREF * c / 1024;
    if (voltage < 11.5) {
        if (voltage < 11.1) {
            digitalWrite(pin_fan, LOW);
            tone(pin_fan, 880);
            delay(3000);
            noTone(pin_fan);
            digitalWrite(pin_fan, LOW);
            sleep(true); // бесконечный сон
        }
        if (millis() - tmr >= 150000) {
            digitalWrite(pin_fan, LOW);
            tone(pin_fan, 880);
            delay(500);
            tone(pin_fan, 880);
            delay(500);
            noTone(pin_fan);
            digitalWrite(pin_fan, fan);
            tmr = millis();
        }
    }
}

int filter() {
    uint16_t sum = 0;
    for (byte i = 0; i < 16; i++)
        sum += analogRead(0);
    return (sum >> 4); // деление на 16
}

```

РЕЦЕНЗИЯ
на научно-исследовательскую работу учащегося
МБОУ лицея №73
«Автоматизированный Заменитель Масок (АЗМ)»

Вышеназванная работа демонстрирует в первую очередь огромное стремление учащегося, проделавшего данную работу, к изучению таких наук как физика, информатика, технология и претворению полученных знаний в практическую плоскость повседневной жизни.

Учащимися проделана большая работа по изучению основ программирования микроконтроллеров и 3D-моделирования, несмотря на то, что он является учеником 6 класса.

Конечно же, до практических результатов данного исследования очень далеко, но оно имеет актуальность в сфере развития инновационных технологий в современных условиях в целях успешной борьбы с заболеваниями.

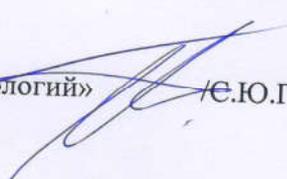
Данная работа также демонстрирует нестандартный подход к решению инженерных задач, конструктивность и нестандартность мышления ученика, большой объем изученного материала вне учебной программы.

Я, считаю, что проведенная работа заслуживает высокой оценки с точки зрения актуальности исследования, его новизны и нестандартности.

Учитель технологии

МБОУ лицей №73 г. Пензы

«Лицей информационных систем и технологий»

 /С.Ю.Пеганов/