**Областная учебно-исследовательская конференция**

**«Юность Поморья»**

Направление **Программирование и информационные технологии**

**Новый элемент экипировки**

**Исследовательская работа**

Выполнена учеником 11 класса муниципального бюджетного общеобразовательного учреждения «Междуреченская СШ №6» Абрамовым Русланом Леонидовичем

Научный руководитель – учитель муниципального бюджетного общеобразовательного учреждения «Междуреченская СШ №6» Игнатьев Павел Алексеевич

**г. Архангельск, 2023**

**Оглавление**

[Введение 3](#_Toc118279880)

[Модель нового элемента экипировки 4](#_Toc118279881)

[Программа для работы модели 5](#_Toc118279882)

[Заключение 6](#_Toc118279883)

[Библиографический список 7](#_Toc118279884)

[Приложение 1.](#_Toc118279885)

[Программа для управления моделью 8](#_Toc118279886)

[Приложение 2.](#_Toc118279887)

[Основные элементы модели 9](#_Toc118279888)

[Приложение 3.](#_Toc118279889)

[Программа для радиомаяка 10](#_Toc118279890)

# Введение

На поле боя одной из основных причин летальных исходов является получение смертельных травм (69,1%), кровопотеря (14,3%), а также кровопотеря и шок (7,3%), шок (2,8%), асфиксия (1,5%).[3] Нередко солдат не может оказать само-и-взаимопомощь в силу разных причин ( болевой шок и пр.) Если вовремя не оказать помощь при артериальном кровотечении человек может погибнуть через 2,5-3 минуты.[9] Согласитесь, это довольно мало. Именно поэтому артериальное кровотечение необходимо купировать быстро. Представители некоторых военных специальностей встречаются с данной проблемой наиболее часто (например: саперы). Предположим, что можно создать новый элемент экипировки, решающий данную проблему.

Цель – создать новый элемент экипировки.

Объект исследования – программа для управления элемента экипировки.

Предмет исследования – элемент экипировки.

В соответствии с данной целью были поставлены следующие задачи:

1. Рассмотреть принцип действия и создать модель нового элемента экипировки.
2. Написать программу для работы модели.

Методы исследования: теоретическое и практическое моделирование, эксперименты.

Гипотеза – возможно создать новый элемент экипировки.

# Модель нового элемента экипировки

Главной проблемой является определение массивной кровопотери. Известно, что кровь человека обладает определёнными показателями: солёности, цветности, плотности и др. Концентрация соли в плазме человека составляет 0,9 % NaCl.[8] При массивной кровопотери кровь неизбежно попадает на обмундирование, где и можно разместить датчики. В модели устройства мы будем использовать датчик концентрации солей.[5] Датчик измеряет концентрацию солей, методом измерения электропроводности. Датчик состоит из измерительного щупа и платы управления. Датчик работает так : на один из электродов щупа подаётся небольшое переменное напряжение, а со второго снимается сигнал, который затем обрабатывается и передается в сигнальный пин платы управления. Так же такие датчики называют TDS-метрами (Total Dissolved Solids), что в переводе означает: общее количество растворенных твердых веществ. В нашей модели датчик будет подключен к питанию 5 вольт, земле и нулевому аналоговому входу микроконтроллера Arduino Uno. Если датчик определит наличие массивной кровопотери, то будут предприняты действия по остановке кровотечения, а именно надут жгут и включен аварийный радиомаяк. В нашей модели датчик будет представлять собой один или несколько расположенных параллельно друг другу проводников, находящихся отдельно или вплетённых в волокна одежды. Реле, подключенное к 6 аналоговому выходу, включает воздушный насос для накачивания манжеты, являющейся жгутом. Десяти секунд работы насоса достаточно для того, что бы прекратить кровообращение в конечности. Далее насос выключается и начинает работу радиомаяк. Модель радиомаяка работает на синтезаторе частоты Si5351.[10][11] Наш телеграфный маяк будет передавать сигнал. Управление синтезатором происходит по шине I2C, для того что бы синтезатор начал выдавать нужные нам частоты нужно записать различные значения в его внутренние регистры, предварительно эти значения нужно рассчитать на основе необходимой нам частоты и данных об использовании опорном кварцевом генераторе.[6] Плата управляется с помощью Arduino, для которой уже существуют готовые библиотеки. Платы радиомаяка и Arduino соединены с помощью четырёх контактов: SCL – 5 аналоговый вход, SDA – 4 аналоговый вход, земля – земля, VIN – к 5-вольтовому входу. Сигнал будет передаваться кодом Морзе.[4] К выходу синтезатора частоты подключена стандартная антенна и всё помещено в корпус. Воздушный насос на 7 вольт, который накачивает манжету, управляется с помощью реле. Питание на насос подаётся отдельно, от связки никель-кадмиевых батарей, в связи с его большим энергопотреблением.

# Пр**ограмма для работы модели**

Опишем программу считывающую данные с датчика соли, управляющую воздушным насосом и радиомаяком. Так как мы используем интерфейс I2C TWI в первой строке программы «constexpr auto pinSensor = A0;» мы используем любой пин с поддержкой АЦП. Следующей строкой «#define PIN\_RELAY 6» мы подключаем реле к 6 аналоговому выходу. Строкой «#define PIN\_RELAY1 7» мы подключаем второе реле к 7 аналоговому выходу. Для получения данных с датчика концентрации соли мы будем использовать монитор порта, для этого задаём скорость обмена строкой «Serial.begin(9600);».[7] Начинаем цикл строкой «void loop()». Наш цикл будет выполняться единожды. Поэтому в конце цикла мы указали строку «while(1) ;».[2] Строкой «int valueSensor = analogRead(pinSensor);» считываем данные с датчика концентрации солей. Следующей строкой «float voltageSensor = valueSensor \* 5 / 1024.0;» мы переводим данные с датчика в напряжение. Строкой «float tdsSensor = (133.42 \* pow(voltageSensor, 3) - 255.86 \* pow(voltageSensor, 2) + 857.39 \* voltageSensor) \* 0.5;» конвертируем напряжение в концентрацию. На нашем датчике собственного изготовления мы провели ряд экспериментов, которые заключались в воздействии на датчик как обычной воды, так и жидкости, с концентрацией солей такой же, как и в крови человека. В программе переменная «tdsSensor» принимает значения больше 150, на 10 секунд включается реле, двигатель воздушного насоса, далее он выключается, и включается радиомаяк, так же с помощью другого реле строкой «digitalWrite(PIN\_RELAY1, HIGH);». Включаем модуль радиомаяка на 10 минут (можем указать любое другое время) строкой «delay(100000);». Данные переменной выводятся на монитор порта строкой «Serial.print(tdsSensor);» . Рассмотрим кратко программу для радиомаяка. Строкой «#define RF\_FREQ 28080.0» мы установили частоту радиочастотного выхода 28 МГц. Мы сделали текст для передачи сообщений строкой «cwTx("VVV");». Для проверки передачи использовалась программа Airspy SDR с USB SDR приёмником и антенной. Сигнал принимался в модуляции CW(телеграфный ключ). [1]

# Заключение

В ходе проделанной работы был изучен вопрос создания нового элемента экипировки и программы для него. Созданная модель имеет практическое применение. Разрешена проблема использования контроля за состоянием с помощью датчика. Созданная модель решает поставленные задачи: надувает жгут в случае необходимости, отправляет сигнал о помощи.

# Библиографический список

1. Airspy [Электронный ресурс] URL: https://airspy.com/
2. While [Электронный ресурс] URL: <https://www.arduino.cc/reference/en/language/structure/control-structure/while/>
3. Анализ летальных исходов, наступивших на поле боя [Электронный ресурс] URL: https://mydocx.ru/4-41906.html
4. Генератор кода Морзе на Arduino Uno [Электронный ресурс] URL: https://microkontroller.ru/arduino-projects/generator-koda-morze-na-arduino-uno/?ysclid=l9zbfziw8713178440
5. Датчик концентрации солей [Электронный ресурс] URL: http://wiki.amperka.ru/products:troyka-tds-sensor
6. Интерфейсная шина IIC (I2C) [Электронный ресурс] URL: http://easyelectronics.ru/interface-bus-iic-i2c.html?ysclid=l9zbcx00aw570338488
7. Монитор порта [Электронный ресурс] URL: <https://роботехника18.рф/монитор-порта-ардуино/#:~:text=М>
8. Плазма крови [Электронный ресурс] URL: https://ege-study.ru/ru/ege/materialy/biologiya/plazma-krovi
9. ПРОБОЛЕЗНИ [Электронный ресурс] URL: https://probolezny.ru/povrezhdenie-magistralnyh-krovenosnyh-sosudov/
10. Простейший радиомаяк на Si5351 и Arduino. Самодельный CW Beacon с синтезатором частоты. [Электронный ресурс] URL: https://youtu.be/1aasO60XZ5o
11. Универсальный синтезатор на микросхеме Si5351 [Электронный ресурс] URL: http://rfanat.qrz.ru/s25/dds\_sio5351\_pro-min.html?ysclid=l9zbaplqdd683020777

# 

# Приложение 1.

## Программа для управления моделью

constexpr auto pinSensor = A0;

#define PIN\_RELAY 6

#define PIN\_RELAY1 7

void setup() {

Serial.begin(9600);

pinMode(PIN\_RELAY, OUTPUT);

pinMode(PIN\_RELAY1, OUTPUT);

}

void loop() {

int valueSensor = analogRead(pinSensor);

float voltageSensor = valueSensor \* 5 / 1024.0;

float tdsSensor = (133.42 \* pow(voltageSensor, 3) - 255.86 \* pow(voltageSensor, 2) + 857.39 \* voltageSensor) \* 0.5;

if (tdsSensor>150) {

Serial.println("Датчик сработал");

digitalWrite(PIN\_RELAY, HIGH);

delay(10000);

digitalWrite(PIN\_RELAY, LOW);

digitalWrite(PIN\_RELAY1, HIGH);

delay(100000);

Serial.print("TDS Value = ");

Serial.print(tdsSensor);

Serial.println(" ppm");

delay(1000);

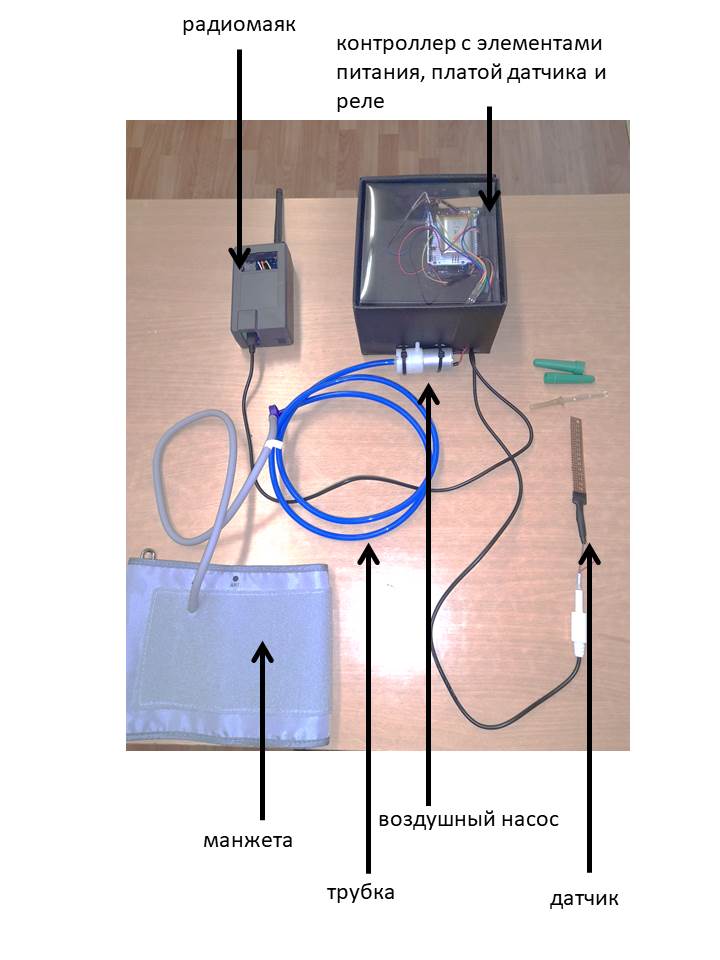
while(1) ;

}

}

# Приложение 2.

## Основные элементы модели



# Приложение 3.

## Программа для радиомаяка

#include "si5351.h"

#include "Wire.h"

#define RF\_FREQ 28080.0

#define CW\_DOT\_LEN 120

Si5351 si5351;

const uint8\_t cwSymbTab[][5] = {

{1, 2}, // 0 A

{2, 1, 1, 1}, // 1 B

{2, 1, 2, 1}, // 2 C

{2, 1, 1}, // 3 D

{1}, // 4 E

{1, 1, 2, 1}, // 5 F

{2, 2, 1}, // 6 G

{1, 1, 1, 1}, // 7 H

{1, 1}, // 8 I

{1, 2, 2, 2}, // 9 J

{2, 1, 2}, // 10 K

{1, 2, 1, 1}, // 11 L

{2, 2}, // 12 M

{2, 1}, // 13 N

{2, 2, 2}, // 14 O

{1, 2, 2, 1}, // 15 P

{2, 2, 1, 2}, // 16 Q

{1, 2, 1}, // 17 R

{1, 1, 1}, // 18 S

{2}, // 19 T

{1, 1, 2}, // 20 U

{1, 1, 1, 2}, // 21 V

{1, 2, 2}, // 22 W

{2, 1, 1, 2}, // 23 X

{2, 1, 2, 2}, // 24 Y

{2, 2, 1, 1}, // 25 Z

{2, 2, 2, 2, 2}, // 26 0

{1, 2, 2, 2, 2}, // 27 1

{1, 1, 2, 2, 2}, // 28 2

{1, 1, 1, 2, 2}, // 29 3

{1, 1, 1, 1, 2}, // 30 4

{1, 1, 1, 1, 1}, // 31 5

{2, 1, 1, 1, 1}, // 32 6

{2, 2, 1, 1, 1}, // 33 7

{2, 2, 2, 1, 1}, // 34 8

{2, 2, 2, 2, 1} // 35 9

};

void cwSendSym(uint16\_t len)

{

si5351.output\_enable(SI5351\_CLK0, 1);

digitalWrite(LED\_BUILTIN,HIGH);

delay(len);

si5351.output\_enable(SI5351\_CLK0, 0);

digitalWrite(LED\_BUILTIN,LOW);

}

void cwTxChar(char ch)

{

uint8\_t cwSym;

uint8\_t tabIndex;

tabIndex = 255;

if ((ch >= 65) && (ch <= 90)) tabIndex = ch - 65;

if ((ch >= 97) && (ch <= 122)) tabIndex = ch - 97;

if ((ch >= 48) && (ch <= 57)) tabIndex = ch - 22;

if (tabIndex == 255)

{

delay(CW\_DOT\_LEN \* 3);

return;

}

for(byte i = 0; i < 5; i++)

{

cwSym = cwSymbTab[tabIndex][i];

if (cwSym == 1) cwSendSym(CW\_DOT\_LEN);

else if (cwSym == 2) cwSendSym(CW\_DOT\_LEN \* 3);

else continue;

delay(CW\_DOT\_LEN);

}

delay(CW\_DOT\_LEN \* 2);

}

void cwTx(char\* msg)

{

for(byte i = 0; i < strlen(msg); i++) cwTxChar(msg[i]);

}

void setup()

{

bool i2c\_found;

pinMode(LED\_BUILTIN,OUTPUT);

Serial.begin(115200);

i2c\_found = si5351.init(SI5351\_CRYSTAL\_LOAD\_8PF, 0, 0);

if(!i2c\_found)

{

Serial.println("si5351 not found on I2C bus!");

while(1)

{

digitalWrite(LED\_BUILTIN,HIGH);

delay(200);

digitalWrite(LED\_BUILTIN,LOW);

delay(100);

}

}

si5351.update\_status();

si5351.drive\_strength(SI5351\_CLK0, SI5351\_DRIVE\_8MA);

si5351.set\_correction(80000, SI5351\_PLL\_INPUT\_XO);

si5351.set\_freq(RF\_FREQ \* 100000ULL, SI5351\_CLK0);

si5351.output\_enable(SI5351\_CLK0, 0);

}

void loop()

{

cwTx("VVV");

delay(2 \* 1000);

}