

RoboCup Junior 2023

***КОМАНДА:
МТКП МГТУ им. Н.Э.
Баумана
ДИСЦИПЛИНА:
RoboCupJunior Rescue
Line***

Abstract

Одним из основных требований в современной робототехнике является переход к автономным робототехническим комплексам. Сегодня роботы стали приоритетным направлением любой отрасли, в том числе и спасательных операций.

Цель нашего проекта - создание макета робота-спасателя с функциями эвакуации пострадавших из аварийных зон. Такой проект с новыми идеями может позволить в будущем реализовать полномасштабный прототип робота-спасателя, который непосредственно будет оказывать помощь людям в экстренных ситуациях. Миссия проекта является привлечение большего количества обучающихся к области робототехники на службе человечеству.

RoboCup Russia Open - RoboCupJunior 2023

Team Description Paper

RoboCupJunior League: Rescue Line

Team Name: "Tilt"

Participants Name:

- **Кайгородцев Глеб, glekay@yandex.ru**
- **Шапилов Максим, shapilov.maksim@mail.ru**
- **Николаев Александр, niklexgg@yandex.ru**
- **Пилипенко Александр, 2004_sasha96@mail.ru**
- **Краснолобов Николай, krasnobovnikolay2004@gmail.com**

Mentor: Бабак Ольга Александровна, babak@bmstu.ru

Institution: [МТКП МГТУ им. Н.Э. Баумана](#)

Country (страна): Россия

Date (дата): 11.05.2023

Кайгородцев Глеб



Шапилов Максим



Пилипенко Александр



Николаев Александр



Краснолобов Николай



Бабак Ольга Александровна



Team Information

Team Name: “Tilt”

Country/Region: РФ, Москва

The Participants Name and their Technical Role:

Member 1: [Кайгородцев Глеб](#) – Ведущий конструктор-проектировщик проекта.

Member 2: [Шапилов Максим](#) – Конструктор-проектировщик захватывающего устройства.

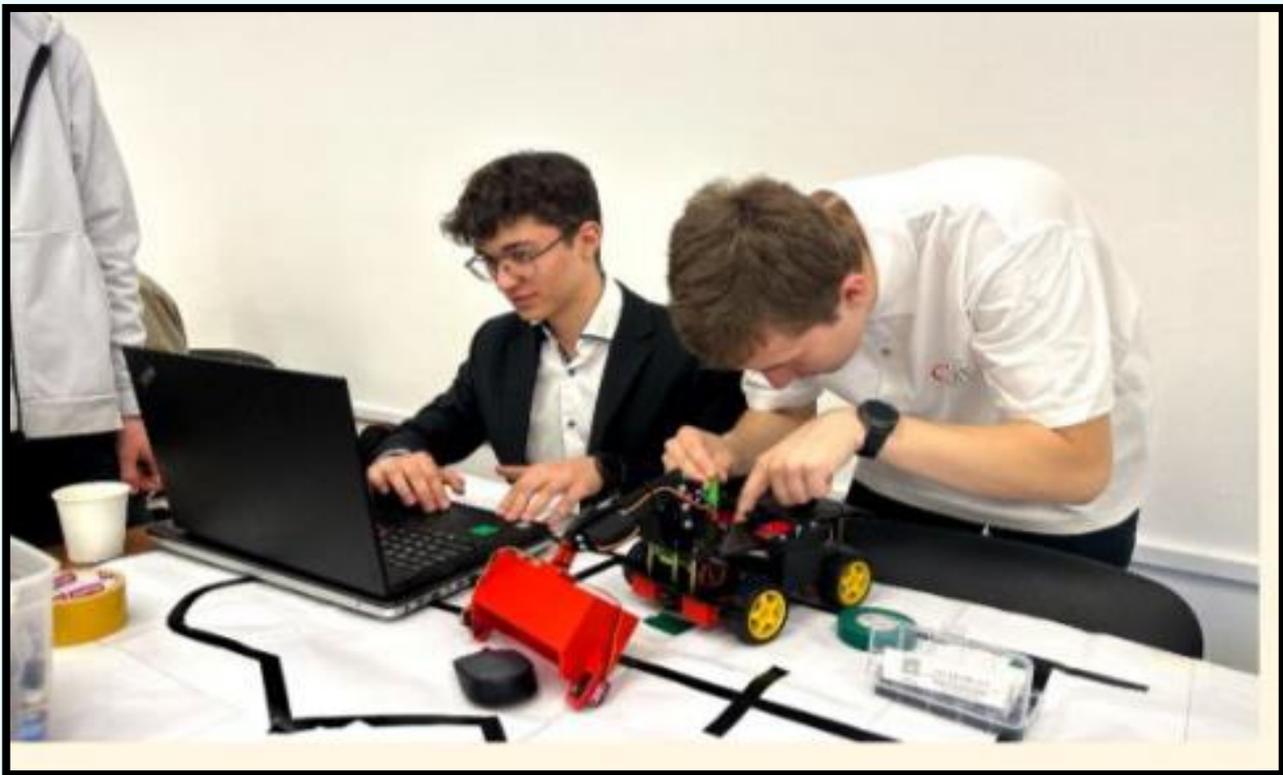
Member 3: [Пилипенко Александр](#) – Ведущий программист проекта, проектировщик электронной части.

Member 4: [Николаев Александр](#) – Специалист по технологии 3Д-печати, монтажник.

Member 5: [Краснолобов Николай](#) – программист проекта, проектировщик электронной части.

Team Background

Команда принимала участие в отборочном Московском этапе RoboCupOpen Russia2023, где заняла 2 место в лиге Rescue Line и выиграла в номинации “Лучший инженерный журнал”.



Technical information

Наша команда представляет проект под аббревиатурным названием **"В.А.Л.Е.Р.А"**. Все его преимущества и функции описаны в этой расшифровке:

- **В** - **вездеходный**, это обеспечивается за счет устойчивой колесной базы и повышенного сцепления на колесах робота.
- **А** - **автоматический**, **"В.А.Л.Е.Р.А"** является полностью автономной машиной для выполнения задач, непосильных человеку.
- **Л** - **легко маневрирующий**, в силу своих сервомоторов с крутящим моментом в 10 кг*см робот легко проезжает любые препятствия и сложные участки трассы.
- **Е** - **ёмкий**, в силу своих небольших размеров, **"В.А.Л.Е.Р.А"** имеет очень насыщенную начинку внутри ([читать подробнее](#)).
- **Р** - **робот**
- **А** - **аптекарь**, из-за того, что робот может доставлять медицинский пакет с помощью захватывающего устройства.

Главными задачами проекта являются:

- 1. Выбор элементной базы под проектируемые задачи**
- 2. Схемы подключения электроники проекта**
 - **Преобразователи напряжения DC-DC, их функции и назначение**
- 3. Спроектировать 3Д-модель основания робота**
- 4. Разработать 3Д-модель захватывающего устройства (ковша)**
- 5. Обоснованно выбрать вид пластика для печати**
- 6. Подготовить 3Д-модели к печати на 3Д-принтере**
- 7. Сборка всех частей робота**
 - **Сборка корпуса**
 - **Подключение электроники и датчиков**
- 8. Этап программирования робота**
- 9. Отладка программы**
- 10. Исправление выявленных ошибок**

График работы над проектом:

Дата	Реализуемая задача
28.03.2023	Появление идеи участия в соревнованиях по робототехнике
29.03.2023 – 05.04.2023	Ознакомление с регламентом RoboCupJunior и окончательный выбор лиги Rescue Line League
06.04.2023 – 09.04.2023	Создание набросков и идей реализации проекта
10.04.2023 – 12.04.2023	Распределение ролей и определение обязанностей
11.04.2023 – 15.04.2023	Выбор элементной базы
14.04.2023 – 26.04.2023	Печать деталей конструкции корпуса
13.04.2023 – 16.04.2023	Создание пластин и ходовой части робота
13.04.2023 – 19.04.2023	Первичная сборка робота (сборка корпуса и монтаж элементной базы) и начальная отладка кода
20.04.2023 – 24.04.2023	Разработка и создание захватывающего устройства
24.04.2023 – 25.04.2023	Полная сборка и отладка робота
24.04.2023 – 29.04.2023	Подготовка и участие в Московском этапе.
30.04.2023-04.05.2023	Устранение недостатков робота в корпусе и электронной части.
05.05.2023 – 11.05.2023	Программирование и отладка программы.

Ход выполнения задач проекта

1. Выбор элементной базы под проектируемые задачи

Основой нашего проекта стала платформа **ArduinoMega 2560** на микроконтроллере **ATmega2560**. Выбрана она в силу достаточно большого количества своих входов/выходов, необходимых для подключения выводов всех датчиков.



Плата Arduino	Микроконтроллер	Рабочее напряжение (В)	Цифровые входы/выходы	Выходы с ШИМ	Аналоговые входы/выходы
Uno	Atmega328	5	14	6	6
Leonardo	ATmega32u4	5	20	7	12
Nano	ATmega328	5	14	6	8
Mega	ATmega2560	5	54	14	16

Таблица сравнения входов/выходов Arduino

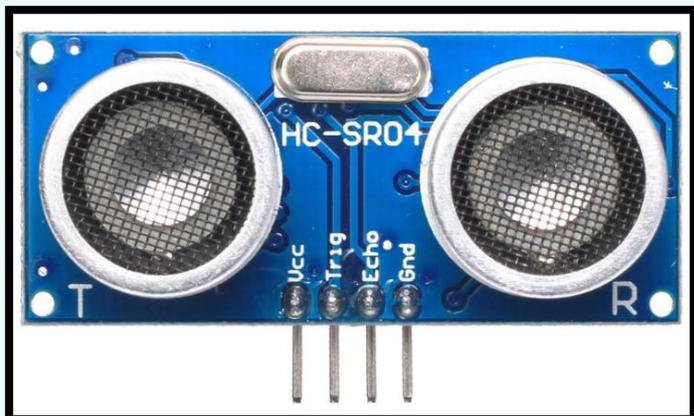
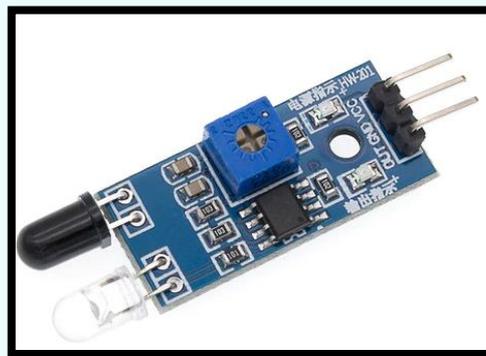
В качестве моторов были выбраны сервоприводы **MG995** с крутящим моментом в **10 кг*см** с шестеренками из металла



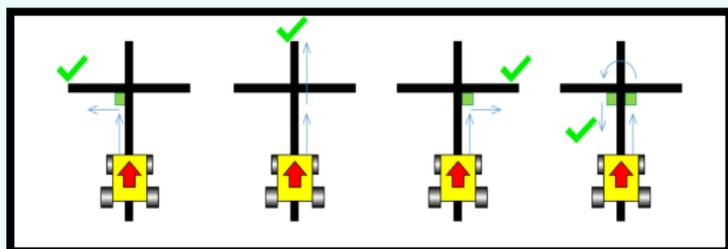
Выбор датчиков основывался на основных задачах, которые должен выполнять робот для участия на соревнованиях **RoboCupRussia Open 2023**:

- **Езда по линии**
- **Объезд препятствий**
- **Определение направления пути на перекрестках**
- **Эвакуация пострадавших из зоны эвакуации**

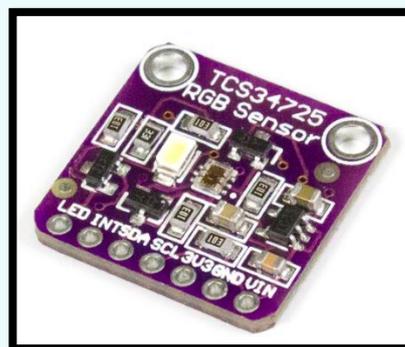
Для реализации езды по линии были выбраны **ИК-датчики** с регулированием чувствительности определения отражения. Принцип работы очень простой и надежный – это и есть главные достоинства данного элемента.



Система объезда препятствий основана на ультразвуковом датчике **HC-SR04**, который определяет расстояние до объектов по отраженному звуковому сигналу.



Возможные



TCS34725

Поворот на перекрестках по регламенту Лиги осуществляется благодаря зеленым квадратам на углах линии. Для решения этой задачи были выбраны датчики цвета **TCS34725**

Так же для работы системы захватывающего устройства будут использоваться 2 типа сервоприводов **MG90S** и **MG996R** с металлическими шестеренками.

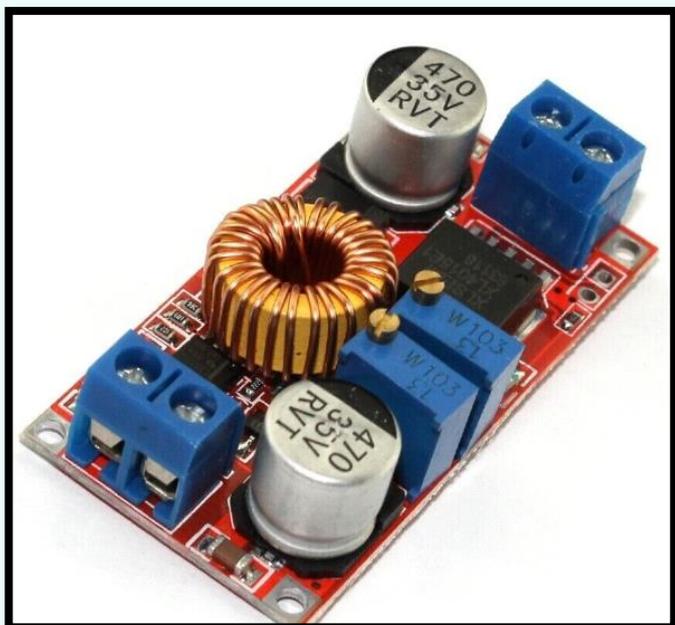


MG90S

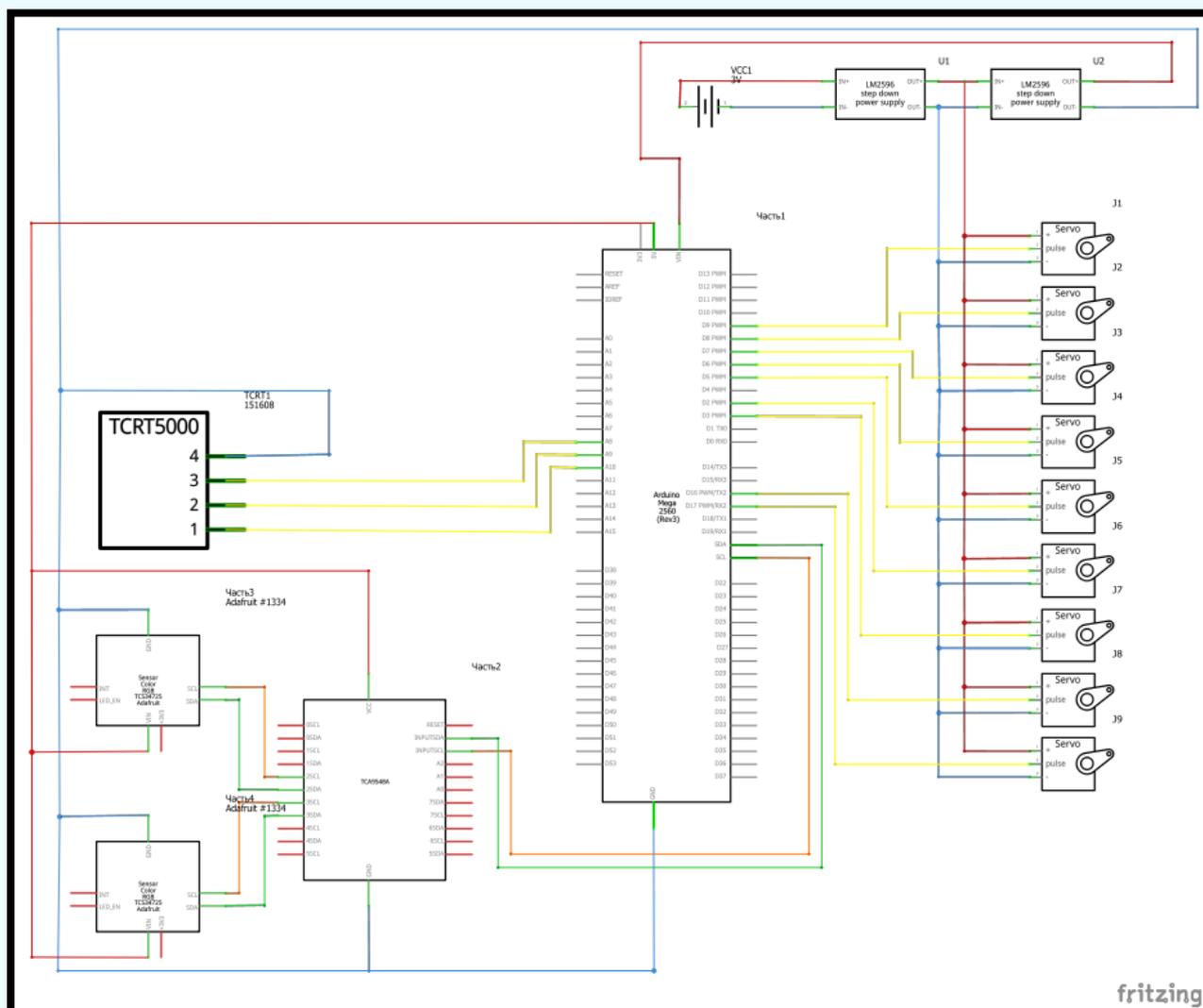


MG996R

В схеме питания платы и датчиков используется два преобразователя напряжения **DC-DC**.



Итоговая схема подключения с изменениями выглядит так:



Программирование робота

Код программы езды по линии:

Алгоритм действия ПИД регулятора в системе сервоприводов.

1. Считывание сигнала с датчиков и запись в массив
2. Расчет ошибки исходя из показаний датчиков
3. Расчет управляющего сигнала ПИД-регулятора
4. Расчет скорости моторов.

```
//-----  
//Расчет ошибки  
//-----  
LFSensor[0] = analogRead(sensorL2);  
LFSensor[1] = analogRead(sensorL1);  
LFSensor[2] = analogRead(sensorM);  
LFSensor[3] = analogRead(sensorR1);  
LFSensor[4] = analogRead(sensorR2);  
  
if(( LFSensor[0]>=950 )&&(LFSensor[1]>=950 )&&(LFSensor[2]>=950)&&(LFSensor[3]>=950 )&&(LFSensor[4]<910 )) {error = -4;}  
else if((LFSensor[0]>=950 )&&(LFSensor[1]>=950 )&&(LFSensor[2]>=950 )&&(LFSensor[3]<900 )&&(LFSensor[4]<910 )) {error = -3;}  
else if((LFSensor[0]>=950 )&&(LFSensor[1]>=950 )&&(LFSensor[2]>=950 )&&(LFSensor[3]<910 )&&(LFSensor[4]>=950 )) {error = -2;}  
else if((LFSensor[0]>=950 )&&(LFSensor[1]>=950 )&&(LFSensor[2]<910 )&&(LFSensor[3]<910 )&&(LFSensor[4]>=950 )) {error = -1;}  
else if((LFSensor[0]>=950)&&(LFSensor[1]>=950 )&&(LFSensor[2]<910 )&&(LFSensor[3]>=950 )&&(LFSensor[4]>=950 )) {error = 0;}  
else if((LFSensor[0]>=950 )&&(LFSensor[1]<910 )&&(LFSensor[2]<910 )&&(LFSensor[3]>=950 )&&(LFSensor[4]>=950 )) {error =1;}  
else if((LFSensor[0]>=950 )&&(LFSensor[1]<910 )&&(LFSensor[2]>=950 )&&(LFSensor[3]>=950 )&&(LFSensor[4]>=950 )) {error = 2;}  
else if((LFSensor[0]<910 )&&(LFSensor[1]<910 )&&(LFSensor[2]>=950 )&&(LFSensor[3]>=950 )&&(LFSensor[4]>=950 )) {error = 3;}  
else if((LFSensor[0]<910 )&&(LFSensor[1]>=950 )&&(LFSensor[2]>=950 )&&(LFSensor[3]>=950 )&&(LFSensor[4]>=950 )) {error = 4;}  
  
P = error;  
I = I + error;  
D = error-previousError;  
PIDvalue = (Kp*P) + (Ki*I) + (Kd*D);  
previousError = error;  
  
int leftFMotorSpeed = 1500 + iniMotorPower - PIDvalue;  
int rightFMotorSpeed = 1500 - iniMotorPower*adj - PIDvalue;  
int leftBMotorSpeed = 1500 + iniMotorPower - PIDvalue;  
int rightBMotorSpeed = 1500 - iniMotorPower*adj - PIDvalue;  
  
constrain(leftFMotorSpeed, 1400, 1600);  
constrain(rightFMotorSpeed, 1400, 1600);  
constrain(leftBMotorSpeed, 1400, 1600);  
constrain(rightBMotorSpeed, 1400, 1600);  
  
leftFServo.writeMicroseconds(leftFMotorSpeed);  
rightFServo.writeMicroseconds(rightFMotorSpeed);  
leftBServo.writeMicroseconds(leftBMotorSpeed);  
rightBServo.writeMicroseconds(rightBMotorSpeed);
```

Прохождение пунктирной линии и поворотов на 90 градусов

```
// Прохождение пунктира
//-----
if ((error == 0) && (analogRead(sensorM) >= 950)) {
    leftFServo.write(120);
    rightFServo.write(60);
    rightBServo.write(60);
    leftBServo.write(120);

    delay(100);
}

if(flag == 0){
    if(((LFSensor[2]<900)&&(LFSensor[3]<900)&&(LFSensor[4]<900))|((LFSensor[2]<900)&&(LFSensor[3]<900))){
        leftFServo.write(120);
        rightFServo.write(120);
        rightBServo.write(120);
        leftBServo.write(120);
        delay(400);
    }

    if(((LFSensor[0]<900)&&(LFSensor[1]<900)&&(LFSensor[2]<900))|((LFSensor[0]<900)&&(LFSensor[1]<900))){
        leftFServo.write(60);
        rightFServo.write(60);
        rightBServo.write(60);
        leftBServo.write(60);
        delay(400);
    }
}
```

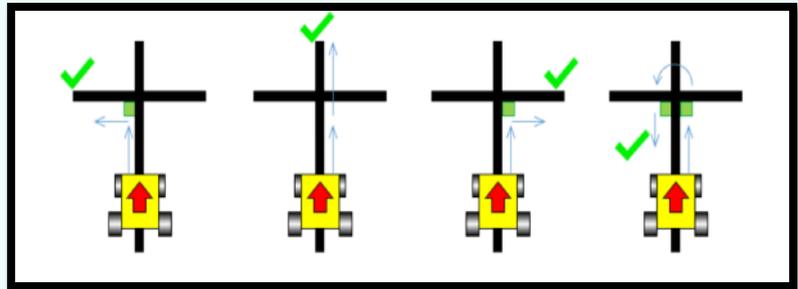
Условия прописаны с флагами, чтобы не сбивать работу других алгоритмов кода.

Детектирование зеленых меток на перекрестках.

Алгоритм основывается на различении зеленого и красного цвета из диапазона

Регламентом лиги предусматриваются следующие взаимодействия с зелеными маркерами:

Для этого были прописаны следующие условия: поворота налево, проезд прямо, поворот направо и разворот на 360 градусов.



Пример кода:

```
//-----  
//Зеленая метка  
//-----  
  
tcaselect(2);  
tcs.getRGB(&red, &green, &blue); // запрашиваем у датчика цветные компоненты  
delay(20);  
tcaselect(3);  
tcs.getRGB(&red1, &green1, &blue1);  
delay(20);  
  
if((green>95)&&(green<135)&&(red>40)&&(red<70)){  
    flag = 1;  
    leftFServo.write(90);  
    rightFServo.write(90);  
    rightBServo.write(90);  
    leftBServo.write(90);  
    delay(1000);  
  
    leftFServo.write(120);  
    rightFServo.write(60);  
    rightBServo.write(60);  
    leftBServo.write(120);  
  
    delay(200);  
  
    leftFServo.write(120);  
    rightFServo.write(120);  
    rightBServo.write(120);  
    leftBServo.write(120);  
    delay(1000);  
    flag = 0;  
}  
  
if((green1>95)&&(green1<135)&&(red1>40)&&(red1<70)){  
    flag = 1;  
    leftFServo.write(90);  
    rightFServo.write(90);  
    rightBServo.write(90);  
    leftBServo.write(90);  
    delay(1000);  
}
```

Детектирование датчиками цвета зеленых маркеров происходило следующим образом:

Сначала идет запрос сигнала с датчика:

```
tcselect(2);  
tcs.getRGB(&red, &green, &blue); // запрашиваем у датчика цветные компоненты  
delay(20);  
tcselect(3);  
tcs.getRGB(&red1, &green1, &blue1);  
delay(20);
```

Датчик TCS34725 выдает 3 значения: содержание **красного (R), **зеленого (G)**, и **синего (B)** цвета в диапазоне от 0 до 255.**

```
Red = 148; Green = 210; Blue = 163  
Red = 89; Green = 205; Blue = 160  
Red = 87; Green = 205; Blue = 160  
Red = 87;
```

Autoscroll Show timestamp

Исходя из этих значений можно прописать условия, при которых датчик будет распознавать зеленую метку.

Эвакуация пострадавших из зоны.

```
void evacuation(){
    leftFServo.write(90);
    rightFServo.write(90);
    rightBServo.write(90);
    leftBServo.write(90);

    for (int pos = 0; pos <= 180; pos += 1) {
        mainscoop.write(pos);
        delay(25);
    }

    clashRight.write(180);
    delay(300);

    for (int pos1 = 180; pos1 >= 150; pos1 -= 1) {
        secscoop.write(pos1);
        delay(35);
    }

    delay(5000);

    for (int pos2 = 180; pos2 >= 15; pos2 -= 1) {
        clashRight.write(pos2);
        delay(20);
    }

    //поднятие наверх
    for (int pos = 180; pos > 0; pos -= 1) {
        mainscoop.write(pos);
        delay(25);
    }
    for (int pos1 = 150; pos1 <= 180; pos1 += 1) {
        secscoop.write(pos1);
        delay(35);
    }
}
```

Алгоритм сделан на плавном управлении сервоприводами с помощью цикла `for()`

Discussion and Conclusion

Во время подготовки к основному этапу RoboCup 2023, команда вспомнила и закрепила знания и умения во всех областях создания роботов таких как

- *проектирование в 3Д-системах*
- *подготовка модели к 3Д-печати*
- *Сборка робота*
- *Программирование робота*
- *Отладка программы*

В будущих работах планируется перейти на усовершенствованную систему на плате Raspberry Pi 3B и попытаться изучить ComputerVision.