ИНЖЕНЕРНАЯ ТЕТРАДЬ

[Наша Команда 2](#_Toc1)

[Конструкция Робота 3](#_Toc2)

[Программный Код 4](#_Toc3)

[Тестирование и Проблемы 6](#_Toc4)

# Наша Команда

Иван Бунчук – Капитан команды. Занимается робототехникой 8 год. В команде отвечает за создание проекта: придумал, проработал и описал его. Также пишет код для езды робота.

Тимур Степанов – Является главным программистом в команде. Он пишет сложные программы для робота и всё время занят.

Егор Тараканов – Занимается робототехникой 7 лет. В команде отвечает за составление проекта и остальных текстов выполняет главные задачи с поиском проблемы и решений в проекте.

Владислав Ачох – Занимается робототехникой второй год. В команде отвечает за создание робота: занимается его сборкой и пайкой.

Михаил Мороз – Занимается робототехникой 7 лет. В команде выполняет роль блогера: фотографирует всё и выпускает посты. Также он помогает другим участникам команды.

# **Конструкция Робота**

Наш робот представляет собой колёсную платформу с двумя ведущими колёсами и небольшой шаровой опорой. Управление осуществляется при помощи микроконтроллера WeMos D1 Mini на базе платы Esp 8266, которая помимо основных функций процессора, также имеет модули Wi-Fi и Bluetooth. Контроль над моторами осуществляется с помощью драйвера моторов. Связь между платами, моторами и источником питания осуществляется через контактные провода и макетную плату.

# **Программный Код**

Общее

Благодаря усилиям главного программиста, все программы находятся в открытом доступе на сайте GitHub, а также для упрощения разработки использовалась система контроля версий Git. В итоге мы отточили (либо обучились) навыкам работы с этими инструментами.

Программа для робота и программа для распознавания видео с камеры находятся в разных Git проектах на аккаунте GitHub главного программиста команды. Ссылки на проекты:

Робот (ССЫЛКА)

Камера (ССЫЛКА)

Робот

Специально для соревнований мы изучили С++ и его использование Arduino-подобных проектах.

В интернете мы нашли примеры работы с библиотекой ESP8266WiFi для передачи информации по WiFi посредством HTTP-запросов. С помощью полученных знаний мы расширили функционал, добавив необходимые функции, запускающиеся при HTTP-запросах на плату. Для примера: запрос «http:/192.168.0.4/f» запустил бы функцию езды вперёд (от англ. Forward).

Само перемещение мы реализовали с помощью библиотеки GyverMotors, информацию о котором нашли на официальном сайте разработчика (ВСТАВИТЬ ССЫЛКУ)

Основная Программа

После получения рабочего прототипа робота и его программы мы перешли к разработке Основной Программы – программы, написанной на ЯП Python с использованием библиотеки компьютерного зрения OpenCV2.

Основная Программа находит робота, линию, цилиндры с показаний камеры; Следуя необходимой стратегии выбирает подходящее действие для робота, и выполняет его на нём, используя библиотеку requests – встроенную библиотеку для работы с HTTP-запросами.

В программе реализованы принципы ООП (как и программа для робота, но намного в больших масштабах) для создания поддерживаемого и читаемого кода.

# **Тестирование и Проблемы**

Общее

Во время всего соревнования мы сталкивались со множеством проблем, но всегда мы находили их решение, причём не всегда с той стороны, с которой мы ожидали. Зачастую решение находил кто-нибудь из членов команды, кто ранее не был задействован для решения этой проблемы, также нам помогали советами наставники СЮТ г. Туапсе и другие её ученики. Также мы активно контактировали с участниками других команд. Наш программист зачастую помогал одной из команд, при этом мы сами обсуждали наши проблемы с представителями других команд – и часто получали помощь и поддержку, особенно на товарищеских встречах.

Конструкционные проблемы

Для реализации только координированного движения робота уже требовалось решить множество проблем.

Сначала нам необходимо было найти подходящие компоненты для робота. На это потребовалось значительное время в начале разработки, но у нас получилось с первого раза заказать нужную электронику.

Получив заказанные компоненты, мы поняли что микроконтроллер Wemos D1 mini поставляется с не припаянными контактами, а значит тестировать его было просто невозможно до момента пайки. Также проблема заключалась в некоторых отличиях начала работы с Arduino и с Wemos, так что изначально мы решили что контроллер и вовсе сломан, потому что не смогли установить и запустить на нём тестовый скетч. («Скетч» – название основной программы на микроконтроллере, которая всегда запускается при включении платы).

При пайке с нами делились инструментами и вспомогательными ресурсами участники других команд или другие ученики СЮТа, но оказалось не все из них знали о том, как паять электронику, а мы тогда были также неопытны и следовали всем советам что слышали. Именно поэтому изначально плату пытались спаять при невероятно высокой температуре, что привело ко множеству проблем в будущем.

Программные проблемы

Код для робота был сделан относительно без проблем, но даже тут не обошлось без багов из-за простой невнимательности. («Баг» – ошибка в программе) Однажды была забыта всего одна строчка, но из-за этого разработка встала на много дней. Нам не помогли даже на специализированном форуме. В итоге проблема обнаружилась при очередном перечитывании кода.

С программой компьютерного зрения всё намного сложнее. Сложность разработки заключается в тестировании – проводить его можно только в определённое время, когда есть возможность повесить камеру, развернуть поле. Это значительно усиливает и так накопительный эффект от того, что проект крайне быстро расширился, поэтому решать какие то задачи становится намного тяжелее.

Для решения последнего ещё в начале разработки была введена и поддерживается структура классов и объектов. Это сильно помогает при разработке и позволяет меньше задумываться об уже готовых элементах и на всём проекте и больше времени уделять конкретным задачам.