

Рисунок 1. Робот

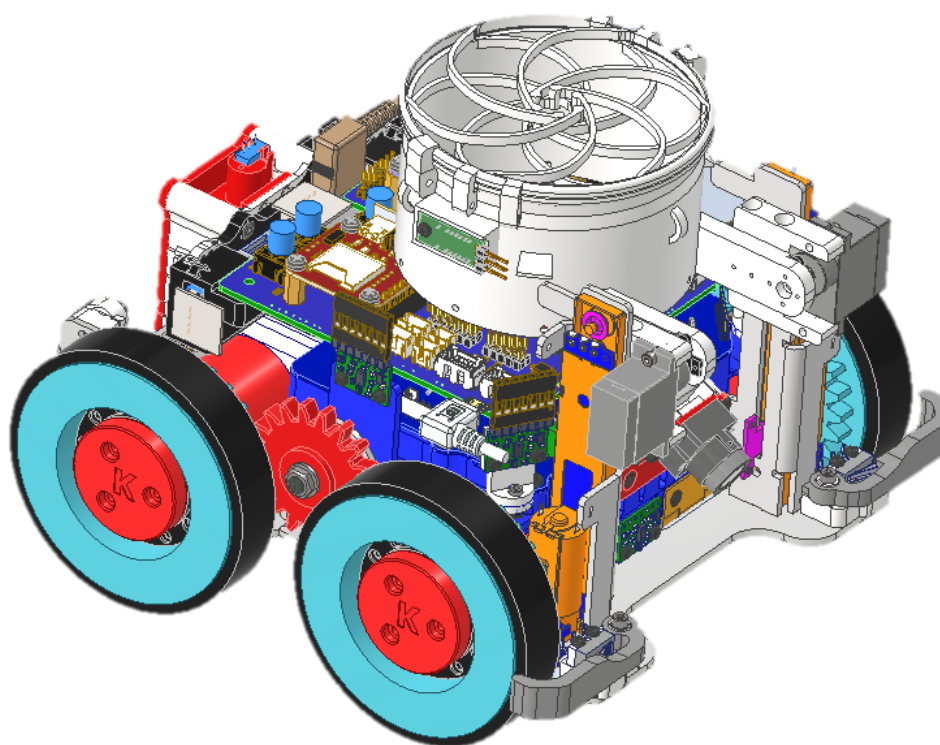


Рисунок 2. 3D-модель робота

СОДЕРЖАНИЕ

1. ПОИСКОВО-ИССЛЕДОВАТЕЛЬСКИЙ ЭТАП.....	4
1.1. Актуальность проекта.....	4
1.2. Цель и задачи.....	5
1.3. Анализ аналогов и прототипа.....	5
1.4. Структурная схема и техническое задание.....	8
2. КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКИЙ ЭТАП	11
2.1. Конструкция и механизмы робота.....	11
2.1.1. Выбор САПР.....	11
2.1.2. Разработка основы робота. 3D-модель робота в САПР.....	11
2.1.3. Проектирование планетарного редуктора для мотора.....	12
2.1.4. Проектирование механизма захвата.....	14
2.1.5. Проектирование механизма подъема захвата.....	15
2.1.6. Расположение датчиков.....	16
2.1.7. Изготовление конструкции и сборка робота	17
2.2. Электроника робота	18
2.2.1. Подбор электронных модулей и других комплектующих. Выбор САПР	18
2.2.2. Разработка принципиальных схем.....	20
2.2.3. Компоновка и трассировка печатных плат	20
2.2.4. Изготовление печатных плат и монтаж компонентов	28
2.3. Программное обеспечение робота.....	29
2.3.1. Выбор среды разработки и языка программирования	29
2.3.2. Общее описание	29
2.3.3. Компьютерное зрение.....	29
2.3.4. Конечный автомат.....	31
2.3.5. Управление захватным механизмом	32
2.3.6. Управление скоростями моторов.....	33
2.3.7. Определение углов.....	34
2.3.8. Движение по линии и распознавание меток.....	35
2.4. Отладка. Тестирование. Модификация.....	35
3. ЗАКЛЮЧИТЕЛЬНЫЙ ЭТАП.....	36
3.1. Эстетический вид и качество робота.....	36
3.2. Новизна и креативность проекта	37
3.3. Практическая значимость.....	37
3.4. Результат.....	37
4. ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ	38
ПРИЛОЖЕНИЕ А	39

1. ПОИСКОВО-ИССЛЕДОВАТЕЛЬСКИЙ ЭТАП

1.1. Актуальность проекта

В настоящее время реализуется автоматизация производств и складов, для этого используются роботы разных типов [1]. Логистические роботы применяются для решения вспомогательных задач, таких как обслуживание оборудования и доставка объектов как большими, так и малыми партиями. При этом на складах часто размещают контрастную разметку на полу, различные метки, например, визуальные или радиометки (Рисунок 3). Нанесенная разметка служит ориентиром для роботов, используемых на подобных складах.

В качестве примера логистического робота можно привести робота компании INTEC [2]: это тележка с манипулятором с грузоподъемностью до 10 кг (Рисунок 4).



Рисунок 3. Роботы на складе



Рисунок 4. Робот компании
INTEC

Навигация по заранее нанесенной разметке, поиск и перемещение объектов являются актуальными задачами современной робототехники. Для их решения подходит робот, обладающий аналогичными датчиками и другими компонентами, что и промышленный, в уменьшенном размере.

Разработкой логистических роботов занимаются как зарубежные, так и российские компании, такие как INTEC, Ronavi, Nissa Engineering и другие. Разработки имеют значимость как для российского, так и для мирового рынка складских и промышленных транспортных роботов.

Вместе с тем с 1996 года проводятся международные соревнования RoboCup[3]. RoboCup – это некоммерческий, научный и культурный проект по продвижению искусственного интеллекта, робототехники и других связанных областей науки и техники посредством организации и проведения робототехнических соревнований. Соревнования являются одним из способов для проверки и доработки новых технологий робототехники, а также привлечения новых участников к решению актуальных задач.

Я участвую в соревнованиях RoboCup в лиге Rescue Line. Полигон представляет собой упрощенную модель местности с разметкой, препятствиями и объектами для поиска и перемещения, поэтому хорошо подходит для тестирования алгоритмов навигации

мобильных роботов, которые затем могут быть реализованы в промышленной робототехнике для решения логистических задач. Мой проект посвящен данным проблемам.

1.2. Цель и задачи

Цель проекта заключается в создании мобильного робота с захватом для отладки алгоритмов навигации по заранее нанесенной цветовой разметке (черной линии с зелеными маркерами на перекрестках) с небольшими препятствиями, поиска объектов различной формы (шариков и кубика) на учебном соревновательном полигоне RoboCup Rescue Line.


Для достижения поставленной цели необходимо выполнить следующие задачи:

1. рассмотреть аналоги, разработать прототип и проанализировать его;
2. составить структурную схему робота и сформулировать техническое задание;
3. подобрать необходимые компоненты и материалы;
4. разработать 3D-модель конструкции в САПР;
5. спроектировать печатные платы;
6. изготовить детали и собрать конструкцию, произвести монтаж компонентов, подключить все модули;
7. запрограммировать и отладить робота;
8. протестировать и модифицировать конструкцию и электронику;
9. оценить и представить работу.

1.3. Анализ аналогов и прототипа

В качестве аналогов и прототипов были рассмотрены роботы других команд, участвовавших ранее в соревнованиях RoboCup Junior Rescue Line. Информация об аналогах представлена в таблице 1.

Таблица 1 – Анализ аналогов

Название	Изображение	Описание
Робот команды Školska Knjiga CRO Team [4]		Робот на 4 колесах. Имеет 10 датчиков освещенности, 8 инфракрасных дальномеров и 4 кнопки для определения стен, 2 RGB-датчика, IMU-модуль (используется в качестве компаса, гироскопа и акселерометра). Также используется камера OpenMV. Захват осуществляется с помощью сервомотора, в захвате установлен дальномер, позволяющий определить наличие мяча. Захват представляет собой ковш с маленькими клешнями внутри для захвата

Название	Изображение	Описание
		кубика. Основная плата Teensy 3.6.
Робот команды Double D [5]		Робот на гусеницах с 2 моторами. В передней части робота расположен захват-манипулятор с 3 степенями свободы. Отличительной особенностью робота является использование камеры OpenMV для прохождения полигона вместо стандартного решения – датчиков освещенности.

Проанализировав представленные и другие аналоги (роботов участников соревнований RoboCup [6]), были сделаны выводы:

Проходить препятствия робот может как на гусеницах, так и на колесах при правильно подобранном центре тяжести. Поэтому выбор сделан в пользу платформы на колесах, как более маневренного решения, а также лишенного проблем со “слетанием” гусеничного трака. Также изготовление гусениц сложнее, чем колес.

Использование датчиков освещенности и цвета – надежный метод, но использование камеры позволяет сделать прохождение полигона более быстрым. Также камера нужна для поиска объектов. Поэтому решено использовать и камеру, и датчики.

Управление захватом манипулятора – сложная задача, а также его сложно настроить на захват предметов разной формы, поэтому захват имеет нестандартную форму – Iris-механизм (механическая диафрагма) [7]. Для его движения требуется всего 1 мотор. При этом чем больше диаметр захвата, тем меньше требуемая точность перемещения. Также такая конструкция позволит схватить шарик даже в углу около стен.

Первоначально был создан прототип робота, имевший множество недостатков (Рисунок 5, Рисунок 6).

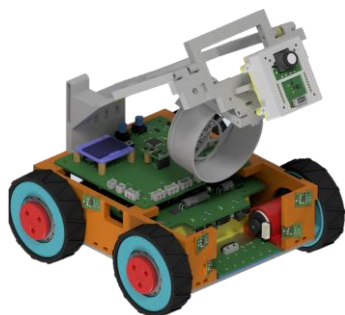


Рисунок 5. 3D-модель прототипа в САПР

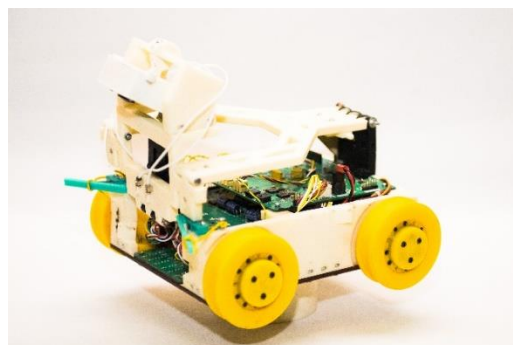


Рисунок 6. Прототип

Были выявлены недочеты и продуманы варианты их исправления. Изменения представлены в таблице 2.

Таблица 2 – Недочеты прототипа и их исправление

Реализация на прототипе	Вариант изменения
Конструкция	
Неудовлетворительные размеры: слишком широкая колея робота не позволяет ему быстро проходить резкие повороты и объезжать препятствия, т.к. робот не помещается на клетках и горке.	Уменьшение габаритов за счет упразднения 2 моторов из 4 с сохранением полного привода за счет паразитной шестерни: на колесах располагаются зубья, с помощью которых они входят в зацепление за шестерню между колесами.
Расположение датчиков освещенности неоптимально: они располагаются впереди геометрического центра основы, из-за чего робот рано реагирует на поворот.	Выбор другого расположения платы датчиков: под аккумулятором, примерно посередине между парами колес.
Расположение камеры не позволяет достаточно точно следовать по линии и определять метки. На изображение не попадают требуемые объекты, но могут попадать лишние элементы. В добавление, камера расположена на механизме захвата, из-за чего может происходить тряска во время движения.	Более удобная реализация расположения камеры: запланировано расположение модуля камеры отдельно от платы, к которой он будет подключен с помощью специального шлейфа.
Дальномеры не позволяют объезжать препятствие. Концевые выключатели не выполняют свою функцию из-за нерационального расположения.	Изменено расположение дальномеров в соответствии с задачей. Продумано положение концевых выключателей и их нажатие.
Конструкция подъема захвата сложно контролируемая.	Решено полностью изменить концепцию подъемника: использовать механизм для вертикального движения захватного механизма и 2 сервомотора для его переворачивания.
Корпус робота представляет собой фанерную основу и стенки из пластика, которые крепятся на нее.	Основу робота решено сделать из алюминия, а для крепления различных модулей – использовать изгибы основы вместо дополнительных деталей.
Электроника	
Движение захватного механизма и подъемника не имеет обратной связи, из-	Решено реализовать обратную связь для контроля открытия захватного механизма с помощью

Реализация на прототипе	Вариант изменения
за чего не может работать корректно.	создания модуля серволебедки. Для контроля высоты подъема решено использовать линейные потенциометры.
С помощью камеры не всегда возможно корректно определить зеленые метки.	Решено использовать датчики цвета совместно с камерой для большей надежности.
Использование 2 микроконтроллеров для разделения задач управления мощностью моторов с корректировкой по энкодерам и задач опроса датчиков и реализации основного алгоритма.	Решено оставить возможность использования 2 микроконтроллеров, но в качестве резервного варианта. Робот имеет такую принципиальную схему, в которой вся периферия управляется лишь одним микроконтроллером.
Отсутствует возможность контроля тока на моторах для механизма захвата.	Добавлена возможность считывания тока с двигателей для их ограничения и предотвращения выхода из строя.
Прошивка с помощью внешнего программатора.	Реализация программатора на материнской плате.
Неудобство прошивки нескольких устройств (модуля камеры, IMU, основного микроконтроллера), необходимость использовать несколько кабелей.	Внедрение USB-хаба в конструкцию робота, подключение к нему всех программируемых устройств.
Программирование	
Реализован линейный алгоритм с задержками, из-за чего невозможно адекватно реализовать выполнение параллельных процессов.	Решено реализовать конечный автомат для устранения задержек.
Неконтролируемые движения захвата.	Реализация регуляторов для контроля высоты подъема.
Отсутствие контроля углов поворота.	Использование показаний углов гироскопа для точных поворотов.

1.4. Структурная схема и техническое задание

Перед началом проектирования была создана электрическая структурная схема робота Э1, которая представлена на с.9.

Было сформулировано следующее техническое задание:

1. Робот должен удовлетворять требованиям, указанным в регламенте RoboCup Rescue Line [8]. Робот должен отслеживать черную линию и зеленые метки на перекрестках, определять угол поворота по 2 осям; преодолевать препятствия на линии высотой до 1 см, клетки-качели и горки, а также проезжать под клетками 2 этажа, не задевая их; объезжать препятствия на линии, возвращаясь в нужную клетку; находить на линии синий кубик со стороной 3 см и перемещать его; определять вход в зону эвакуации (прямоугольная зона со стенами) и находить выход из нее.

2. Геометрические размеры робота не должны превышать 200 мм по ширине и длине, 250 мм по высоте с учетом захвата.

3. Основа робота должна представлять деталь из листового металла.

4. На основе должны быть закреплены 2 мотора с энкодерами, 2 крепления ведомых колес. Робот должен иметь полный привод, для этого между колесами должна быть установлена паразитная шестерня. Необходимо реализовать планетарный редуктор с отношением 3:1 внутри колеса, в котором ведущая шестерня закреплена на валу мотора, а ведомая закреплена на колесе. Колесо не должно быть закреплено только на валу мотора.

5. Робот должен иметь захватный механизм, чтобы захватывать шарики необходимого диаметра и массы, перемещать их и сбрасывать их в зону эвакуации, которая ограничена стенкой высотой 10 см, т.е. захват должен подниматься над поверхностью. Должна быть реализована обратная связь, чтобы робот мог определить высоту расположения захвата и ток на моторах для их защиты. Захват должен быть способным захватывать и шарики, и кубик. При перемещении по линии робот должен иметь возможность расположить захват так, чтобы не превысить допустимую высоту робота. Робот должен определять наличие шарика в захвате с помощью цифрового дальномера.

6. Робот должен обладать 4 дальномерами, 6 датчиками освещенности, 2 датчиками цвета. Робот должен иметь модуль с датчиком-гироскопом и камеру. Робот должен иметь по 2 концевых выключателя спереди и сзади для обнаружения препятствий и стен. Робот должен иметь дисплей и 2 кнопки для возможности отладки.

7. Должны быть созданы 3 печатных платы для робота. На плате датчиков должны быть расположены датчики освещенности и цвета. Должна быть возможность подключения 4 моторов с энкодером и 3 сервомоторов. Микроконтроллеры должны обладать достаточным количеством пинов и внутренней периферии. Робот должен управляться контроллером STM32F407ZGT6 на материнской плате, на плате моторов должна быть предусмотрена возможность установки второго контроллера

STM32F103CBT6. На материнской плате должны быть разъемы для подключения всех датчиков и модулей, а также программатор для прошивки контроллера.

8. В конструкцию должен быть внедрен USB-хаб для удобного подключения модулей и их прошивки.

9. Детали и платы для робота должны быть выполнены в гармоничном сочетании красного, синего и черного цветов.

2. КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКИЙ ЭТАП

2.1. Конструкция и механизмы робота

2.1.1. Выбор САПР

Для разработки конструкции робота была выбрана САПР Autodesk Inventor [9]. Она была выбрана среди аналогов, т.к. у меня уже имеется достаточно много навыков работы в ней, а также имеется большое количество учебных материалов и руководств по ней.

2.1.2. Разработка основы робота. 3D-модель робота в САПР

Разработка конструкции (Рисунок 7, Рисунок 8) была начата с основы. Это деталь из листового материала, на которую крепятся все элементы робота (Рисунок 9). На ней установлена деталь крепления аккумулятора, на которой крепятся также датчики и камера, фланцы для крепления моторов и держатели ведомых колес, а также паразитные шестерни (Рисунок 10). Чертеж основы представлен на с.13.

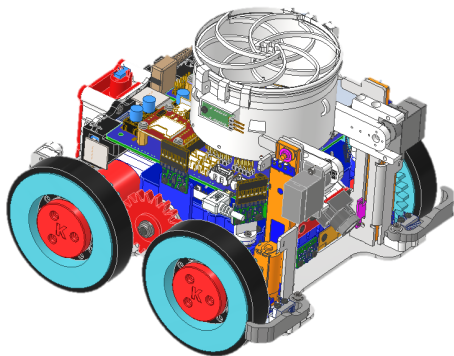


Рисунок 7. 3D-модель робота в САПР

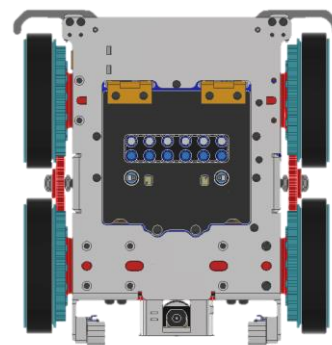


Рисунок 8. 3D-модель робота в САПР, вид снизу

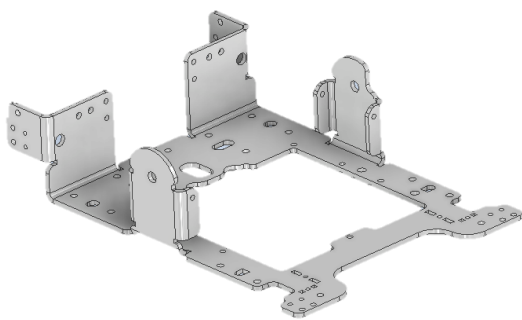


Рисунок 9. 3D-модель основания робота

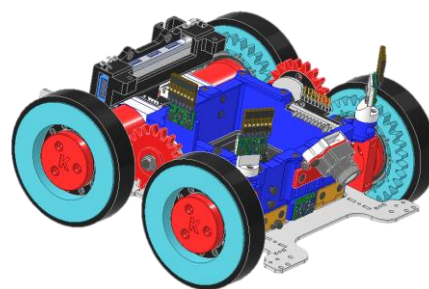


Рисунок 10. 3D-модель основы

2.1.3. Проектирование планетарного редуктора для мотора

Для снижения нагрузки на вал мотора и уменьшения скорости его вращения разработан планетарный редуктор (Рисунок 11). Выбран данный тип редуктора, так как он позволяет сделать конструкцию компактной (редуктор расположен внутри колеса). Также редуктор закрыт крышкой.

Механизм состоит из солнечной шестерни на валу двигателя, 3 закрепленных сателлитов и подвижной коронной шестерни, вместе с которой на подшипнике вращается колесо. Передаточное отношение редуктора 3:1.

Основная нагрузка благодаря данной конструкции приходится не на вал мотора, а на подшипник, на котором закреплено колесо.

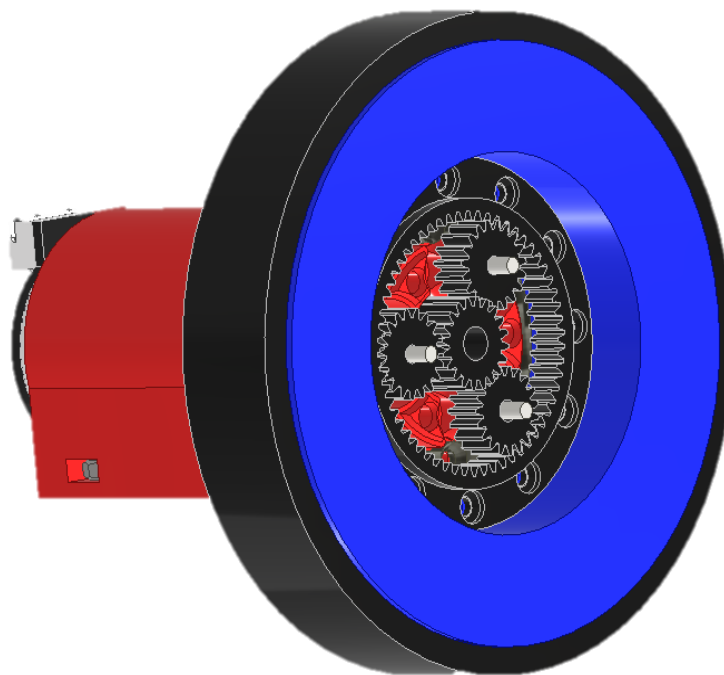


Рисунок 11. 3D-модель редуктора

2.1.4. Проектирование механизма захвата

За основу конструкции захвата был взят Iris механизм, представляющий собой механическую диафрагму (Рисунок 12, Рисунок 13). В захвате расположен цифровой дальномер для обнаружения объекта (Рисунок 14). Захват позволяет хватать и удерживать объекты разной формы (Рисунок 15).

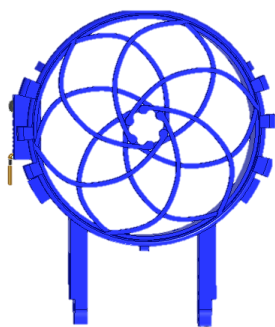


Рисунок 12. Вид снизу

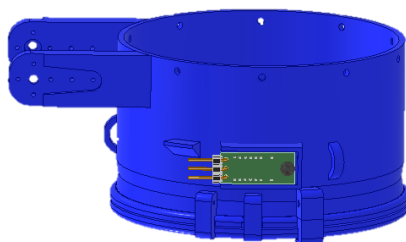


Рисунок 13. Вид сбоку

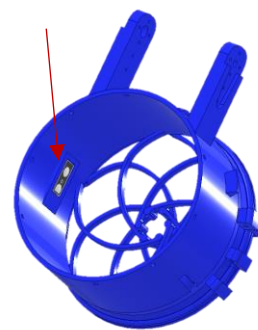


Рисунок 14. Дальномер в захвате

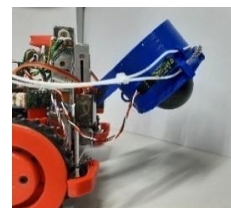
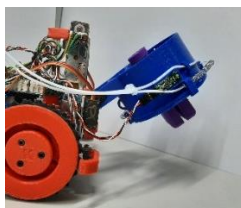
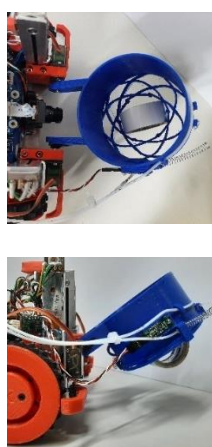


Рисунок 15. Захват и удерживание объектов разной формы

Захват открывается за счет вращения подвижного контура относительно неподвижного при сокращении лески в трубке между серволебедкой и Iris механизмом. Был разработан специальный модуль – серволебедка. Привычный сервопривод (Рисунок 16) способен принимать заданное положение за счет обратной связи в виде показаний потенциометра, который может вращаться на 180 или 360 градусов. Но этого недостаточно, т.к. одного оборота катушки мало, чтобы открыть механизм с диафрагмой.

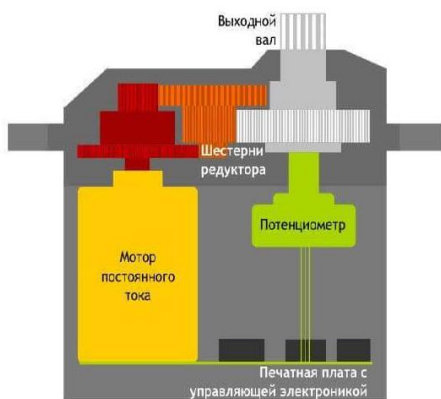


Рисунок 16. Сервопривод в разрезе



Рисунок 17. 3D-модель модуля серволебедки

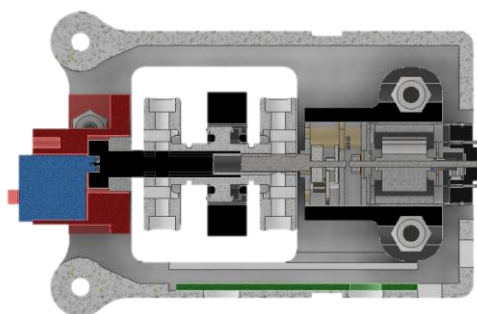


Рисунок 18. Серволебедка в разрезе

Для устранения проблемы ограниченного угла поворота было разработано оригинальное решение. Суть заключается в замене переменного резистора обратной связи на многооборотный резистор такого же сопротивления. Для него разработана новая конструкция корпуса привода, заключающая в себе мотор-редуктор, катушку для двух мотков лески, резистор и плату управления от поврежденного сервопривода. Внешний вид представлен в собранном состоянии (Рисунок 17) и в разрезе (Рисунок 18). Катушка лебедки сделана так, чтобы мотки лески не слетали с нее при вращении. Модуль лебедки крепится в задней части робота.

2.1.5. Проектирование механизма подъема захвата

Для подъема захватного механизма была разработана специальная конструкция с использованием модулей линейных потенциометров с мотором (Рисунок 19). Ход потенциометра по вертикали 60 мм.

Для поворота захватного механизма на 180 градусов используется 2 сервомотора. В верхней точке траектории сервомоторы одновременно поворачиваются на 180 градусов и переворачивают захват (Рисунок 20).

Сервомоторы закреплены на подвижных частях потенциометра с помощью пластиковых деталей.



Рисунок 19. Линейный
потенциометр

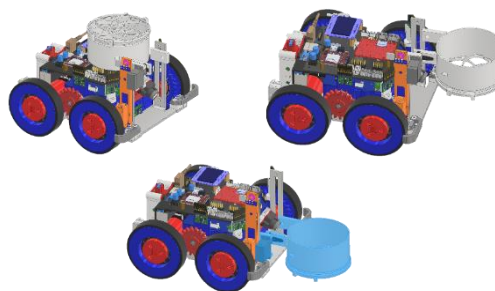


Рисунок 20. Поворот и подъем захвата

2.1.6. Расположение датчиков

Дальномеры нужно было расположить оптимально, чтобы робот мог выполнить требуемую задачу. Один дальномер расположен под камерой, спереди робота и нужен для обнаружения стены спереди или препятствия. Еще два дальномера расположены спереди робота по углам и развернуты под некоторым углом к траектории робота в 2 стороны. Угол подбирается в процессе отладки: крепление можно вращать. Эти дальномеры нужны для ориентирования по стенам и объезда препятствия. Также один дальномер расположен справа и направлен вбок для определения выхода из зоны эвакуации и для объезда препятствия (Рисунок 21).

В отличие от прототипа, расположение концевых выключателей продумано: два расположено спереди, так, что, если робот врежется колесом в препятствие, срабатывает кнопка. И два концевых выключателя расположено сзади для выравнивания по стене в зоне эвакуации (Рисунок 22).

Камера специально вынесена отдельно от платы для ее удобного расположения. Камера расположена на неподвижной части робота и близко к поверхности, в отличие от прототипа. Положение камеры определено исходя из опыта отладки прототипа до начала моделирования робота. Камера закреплена на детали, с помощью которой можно регулировать как высоту над полом, так и наклон (Рисунок 23).

Плата датчиков закреплена внутри крышки аккумулятора (Рисунок 24). На плате расположены 6 датчиков освещенности и 2 датчика цвета. Размещение датчиков в центре основания признано наиболее уместным для навигации на полигоне Reasue Line. По сравнению с прототипом, расстояния между датчиками уменьшены. На плате также расположены датчики цвета со светодиодами. Микросхема датчика цвета и светодиод расположены так, чтобы расстояние между 2 микросхемами равнялось расстоянию между метками на перекрестке, а при нахождении датчиков линии над перекрестком датчики цвета находились над цветными метками. Светодиоды и фототранзисторы разделены деталью из

черного пластика, а также с помощью этой детали сделана ограниченная «зона видимости» для датчика цвета (Рисунок 25).

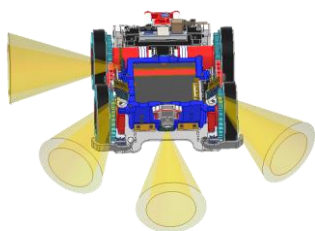


Рисунок 21. Расположение дальномеров

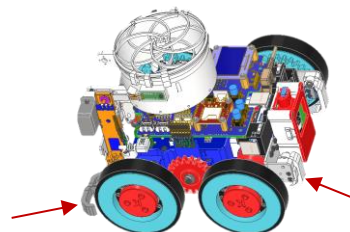


Рисунок 22. Расположение концевых выключателей



Рисунок 23. Крепление камеры

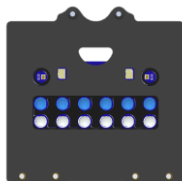


Рисунок 24. Крышка аккумулятора

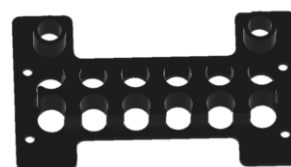


Рисунок 25. Крепление для платы датчиков

2.1.7. Изготовление конструкции и сборка робота

Сначала основание для робота было изготовлено из ABS-пластика для проверки размеров (Рисунок 26). Деталь вырезана из листа пластика на лазерном станке и согнута под нагревом. Так как в условиях школы сложно вырезать подобную, деталь основания заказана на производстве LimaLux [10], где изготовлена из листа алюминия толщиной 2мм (Рисунок 27).

Детали в основном напечатаны на 3D-принтере из ABS и PLA пластика. Шестерня для редуктора изготовлена из PLA пластика для большей прочности, а также из PLA напечатан захват, т.к. он должен обладать упругостью и не быть хрупким. Остальные детали напечатаны из ABS.



Рисунок 26. Основание из пластика

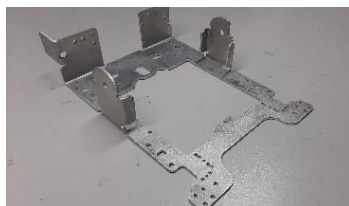


Рисунок 27. Основание из алюминия

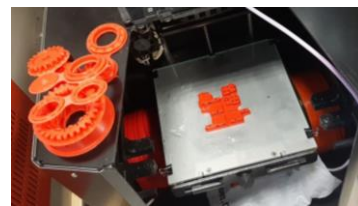


Рисунок 28. 3D-печать деталей

Для удобного соединения деталей используются не стандартные шестигранные гайки, а впаляемые гайки. Они удобны при размещении в модели и при сборке (Рисунок 29, Рисунок 30, Рисунок 31).



Рисунок 29. Впаляемая гайка



Рисунок 30. Вид гайки в разрезе

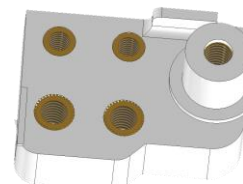


Рисунок 31. Гайки внутри детали

Механизм лебедки улучшен по сравнению с прототипом. Катушка состоит из нескольких деталей, леска закрепляется внутри катушки. Благодаря улучшенной конструкции катушки лески почти не слетают с катушки в состоянии отсутствия натяжения. Т.к. детали катушки имеют маленькие размеры, и требуется точность, которой при FDM печати не достигнуть, использовался фотополимерный принтер.

После изготовления всех деталей произведена сборка робота (Рисунок 32, Рисунок 33, Рисунок 34).

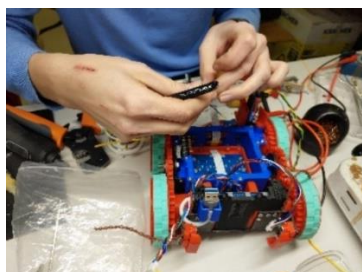


Рисунок 32. Сборка робота

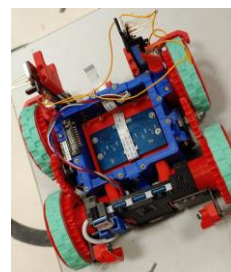


Рисунок 33. Робот в процессе сборки



Рисунок 34. Собранный робот

2.2. Электроника робота

2.2.1. Подбор электронных модулей и других комплектующих. Выбор САПР

Для передвижения платформы выбраны коллекторные моторы с редуктором 34:1 от

DFRobot с напряжением 12В и моментом 12 кг*см [11].

Механизм захвата должен иметь 3 степени свободы: перемещаться вертикально, поворачиваться на 180 градусов и открывать механическую диафрагму. Для подъема захватного механизма были выбраны приводы линейного перемещения с обратной связью (моторизированные линейные потенциометры) ALPS RS60N11M9A0E [12]. С помощью напряжения на потенциометре можно определить положение (высоту) захвата. При напряжении 10 В, рабочее усилие 0.8 Н. В качестве потенциометра обратной связи для серволебедки выбран переменный резистор 3296W-103 [13]

Для реализации датчиков цвета используется 6 пар ИК-светодиод L-7113F3C [14] – фототранзистор L-7113P3C [15]. Питание светодиодов реализовано через драйвер управления светодиодами CAT4237 [16], благодаря чему можно программно управлять их яркостью с помощью ШИМ.

В качестве датчиков цвета выбраны микросхемы TCS34725 [17], общение с которыми реализовано по I2C.

Для робота выбраны лазерные VL53l0X [18], связь с которыми осуществляется по I2C. Выбор обусловлен их малыми габаритными размерами и отсутствием слепой зоны на малых расстояниях.

В качестве гироскопа выбран модуль SparkFun 9DoF IMU Razor M0 [19]. В качестве камеры выбран модуль OpenMV [20].

Внутри захватного механизма установлен цифровой дальномер GP2Y0D810 [21] для определения, есть ли объект внутри после захватывания.

Для вывода отладочной информации используется OLED дисплей 128*64 [22], связь с которым осуществляется по SPI.

Для более удобного подключения и расположения компонентов в работе используются гибкие шлейфы. Коммутационная плата датчиков расположена внутри крепления аккумулятора, шлейф подключается к материнской плате. Отдельно от платы OpenMV вынесен модуль камеры и тоже подключен с помощью шлейфа.

Для управления роботом выбран контроллер STM32F407ZGT6 [23]. Выбор обусловлен достаточным количеством пинов и интерфейсов для подключения всей периферии и управления ей, а также наличием широкой поддержки сообщества использующих данный микроконтроллер.

Было решено спроектировать собственные печатные платы, т.к. намного удобнее использовать плату под конкретную задачу, а не готовую отладочную плату. Можно заранее предусмотреть ее размеры и все необходимое для надежного подключения периферии. Решено разработать 3 платы: материнскую плату, плату управления моторами

и коммутационную плату датчиков.

Список компонентов для материнской платы представлен в Приложении А.

Для проектирования печатных плат была выбрана САПР Altium Designer [24]. Система автоматизированного проектирования Altium Designer — программно-аппаратный комплекс для построения электронных средств на базе печатных плат и программируемых логических интегральных схем, давно ставший популярным среди профессиональных разработчиков печатных плат. Этот САПР удобен тем, что позволяет отслеживать изменения на предыдущих этапах проекта (файл схемы) и переносить их в следующие (файл платы), а также реализовать весь цикл разработки от принципиальной схемы до заказа на производстве.

САПР был освоен в ходе создания робота и печатных плат для него.

2.2.2. Разработка принципиальных схем

Перед созданием электрических принципиальных схем были созданы библиотеки компонентов. Библиотека содержит УГО компонента и посадочное место. Также к каждому посадочному месту добавляется 3D-модель компонента.

При оформлении схем были использованы следующие документы:

- ГОСТ 2.701-2008 ЕСКД. Схемы. Виды и типы. Общие требования к выполнению
- ГОСТ 2.702-2011 Правила выполнения электрических схем
- ГОСТ 2.710-81 Обозначения буквенно-цифровые в электрических схемах
- ГОСТ 2.709-89 Обозначения условные проводов и контактных соединений

электрических

элементов, оборудования и участков цепей в электрических схемах

- ГОСТ 2.725-68 Устройства коммутирующие
- ГОСТ 2.728-74 Резисторы, конденсаторы
- ГОСТ 2.730-73 Приборы полупроводниковые
- ГОСТ 2.759-82 Элементы аналоговой техники

Были разработаны схемы материнской платы, платы управления моторами и коммутационной платы датчиков.

Представлена принципиальная схема материнской платы на с.22-27.

На 1 листе расположен микроконтроллер, на 2 листе — микросхемы управления двигателями, на 3 и 5 листах — разъемы для подключения периферии, на 4 листе — схема программатора, на 6 листе — схемы питания.

2.2.3. Компоновка и трассировка печатных плат

После разработки принципиальных схем было начато создание файла платы.

Сначала в САПР для 3D-моделирования в общей сборке робота были созданы 3D-модели контуров платы со всеми разъемами, чтобы заранее предусмотреть их расположение, затем контур был экспортирован в DXF и перенесен в Altium.

Компоненты из схем были перенесены в файл платы и скомпонованы.

Материнская плата содержит 4 слоя, плата управления моторами – тоже 4 слоя, плата датчиков – 2. Ниже представлена трассировка всех слоев материнской платы (Рисунок 35, Рисунок 36, Рисунок 37, Рисунок 38).

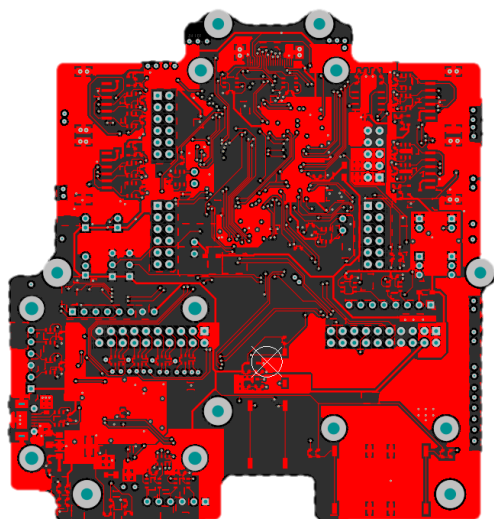


Рисунок 35. Верхний слой

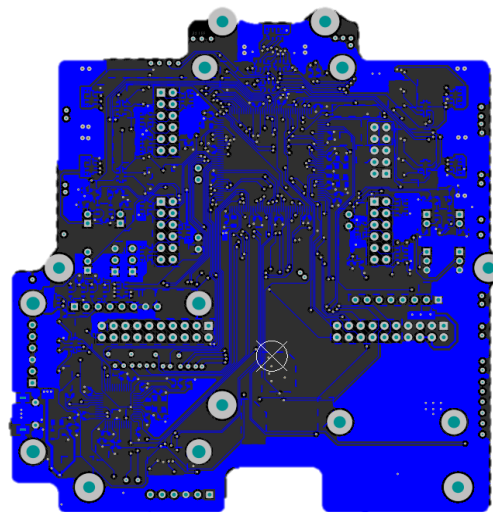


Рисунок 36. Нижний слой

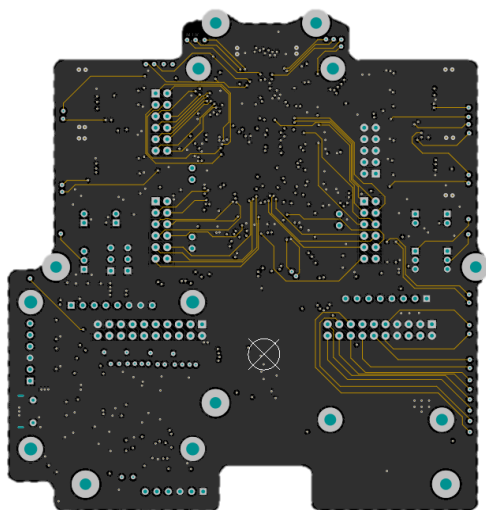


Рисунок 37. Первый сигнальный слой

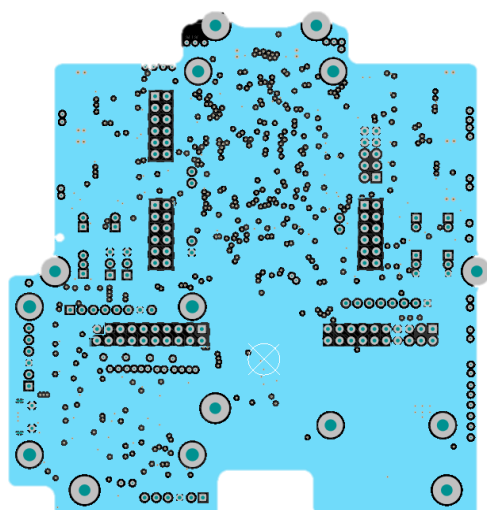
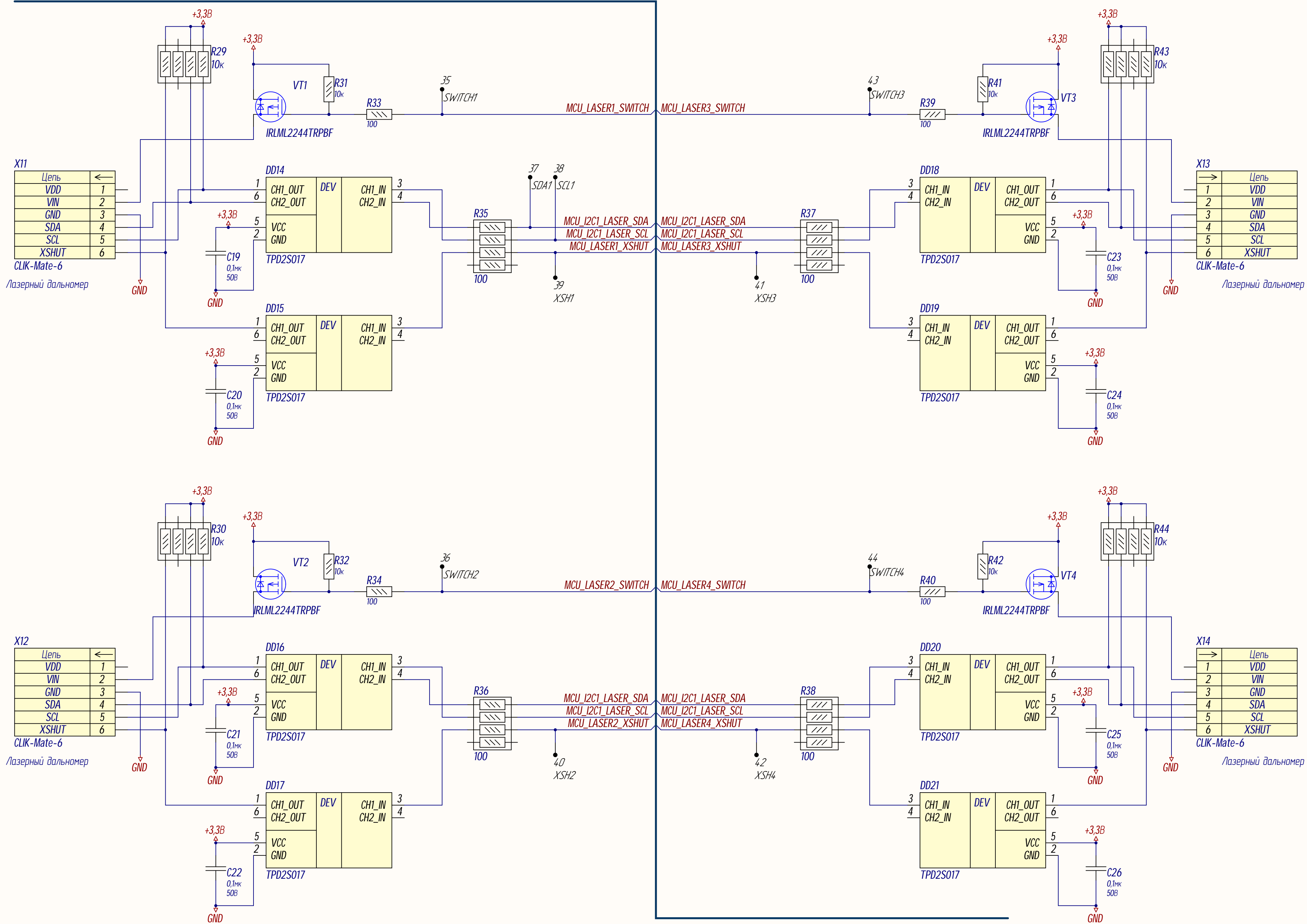


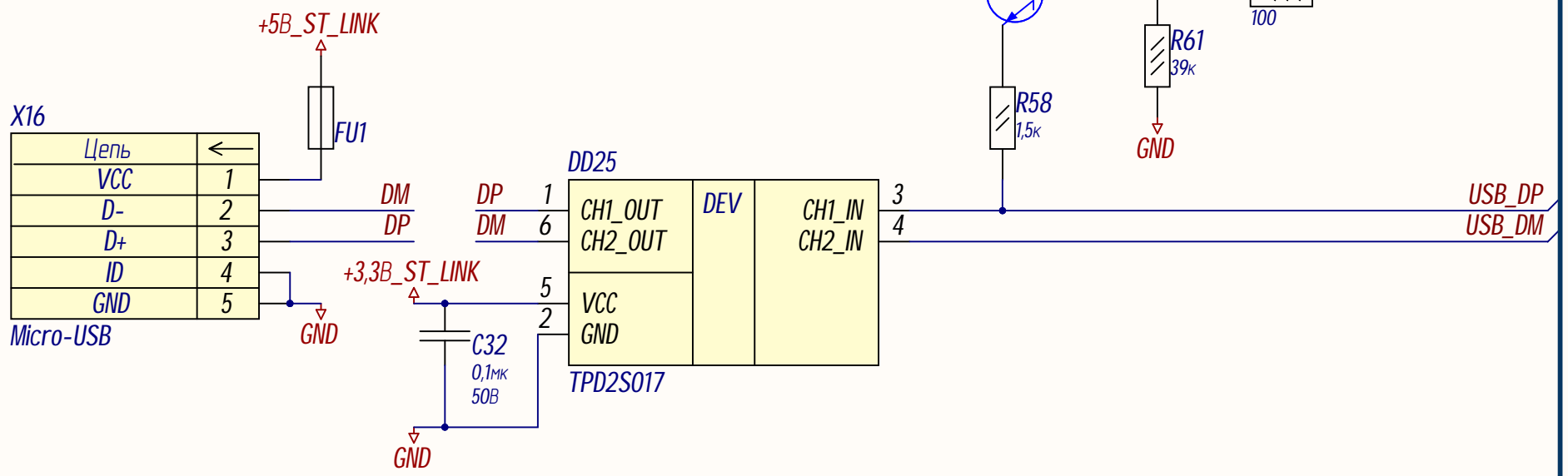
Рисунок 38. Второй сигнальный слой

Стор. №	Перв. намен
---------	-------------

Разработчик		Нормоконтроль:		
Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № з/дод.	Подп. и дата

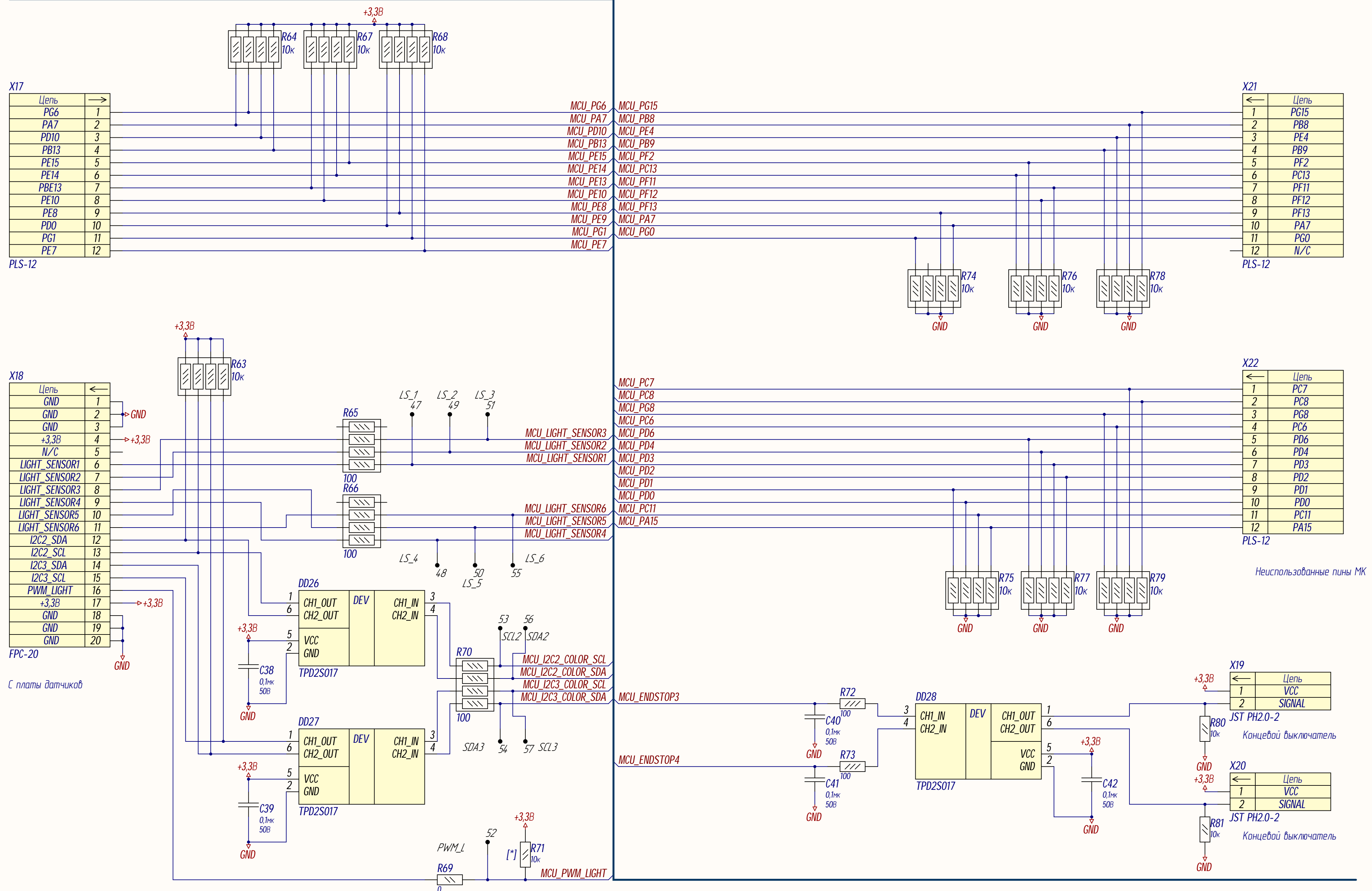
[illegible]

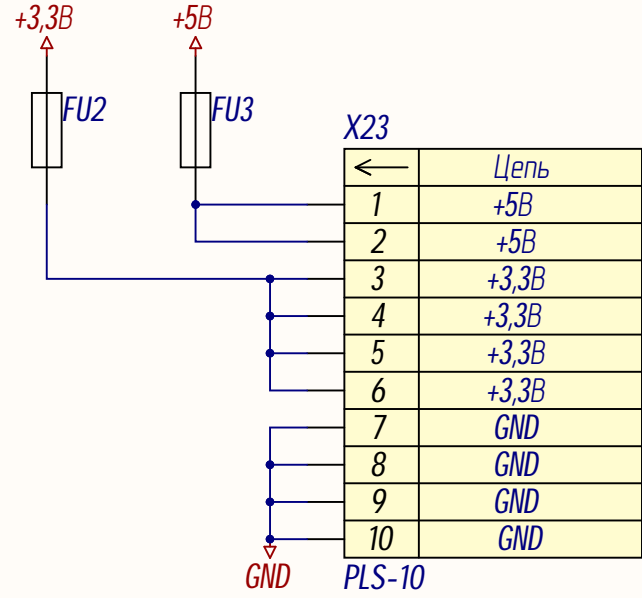
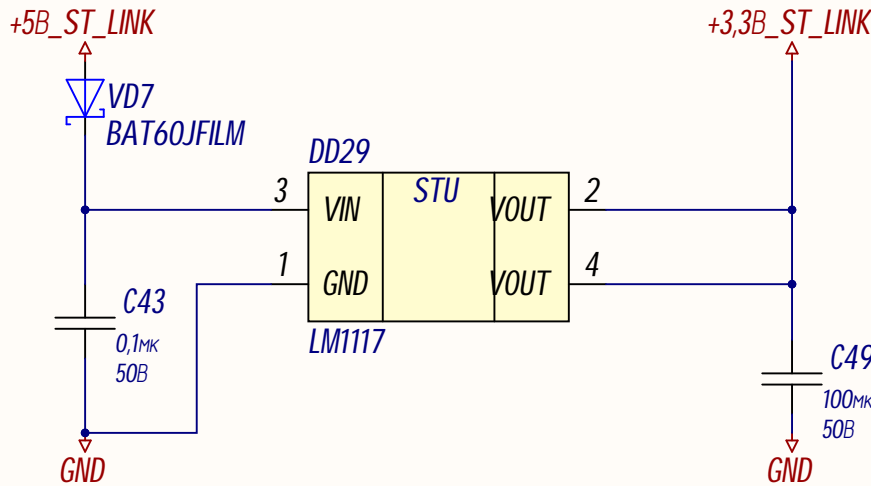




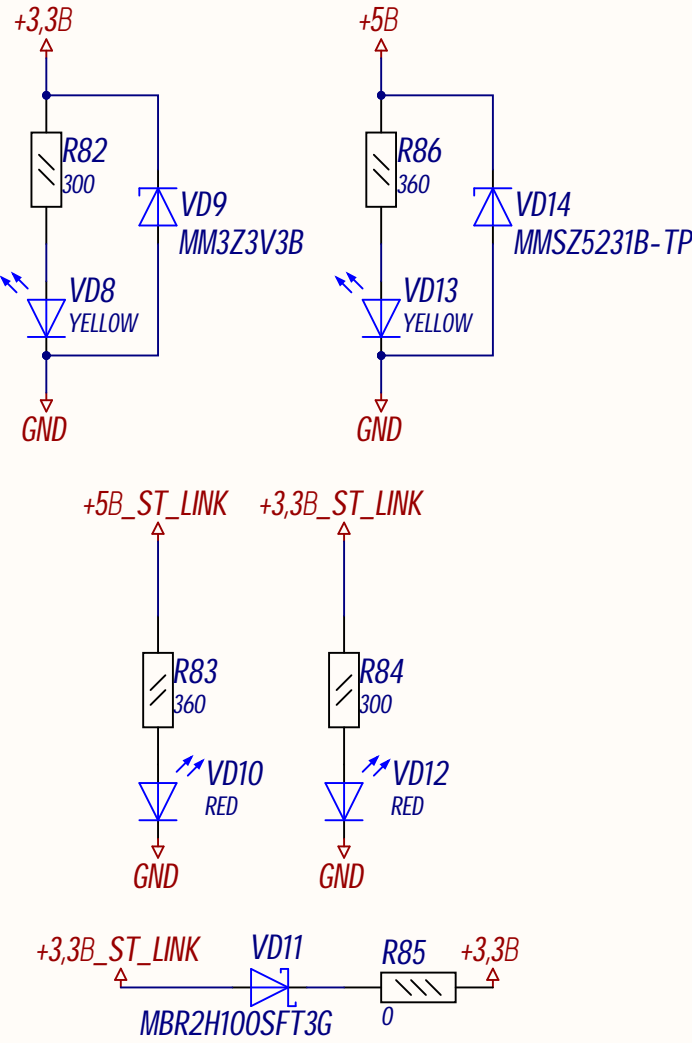
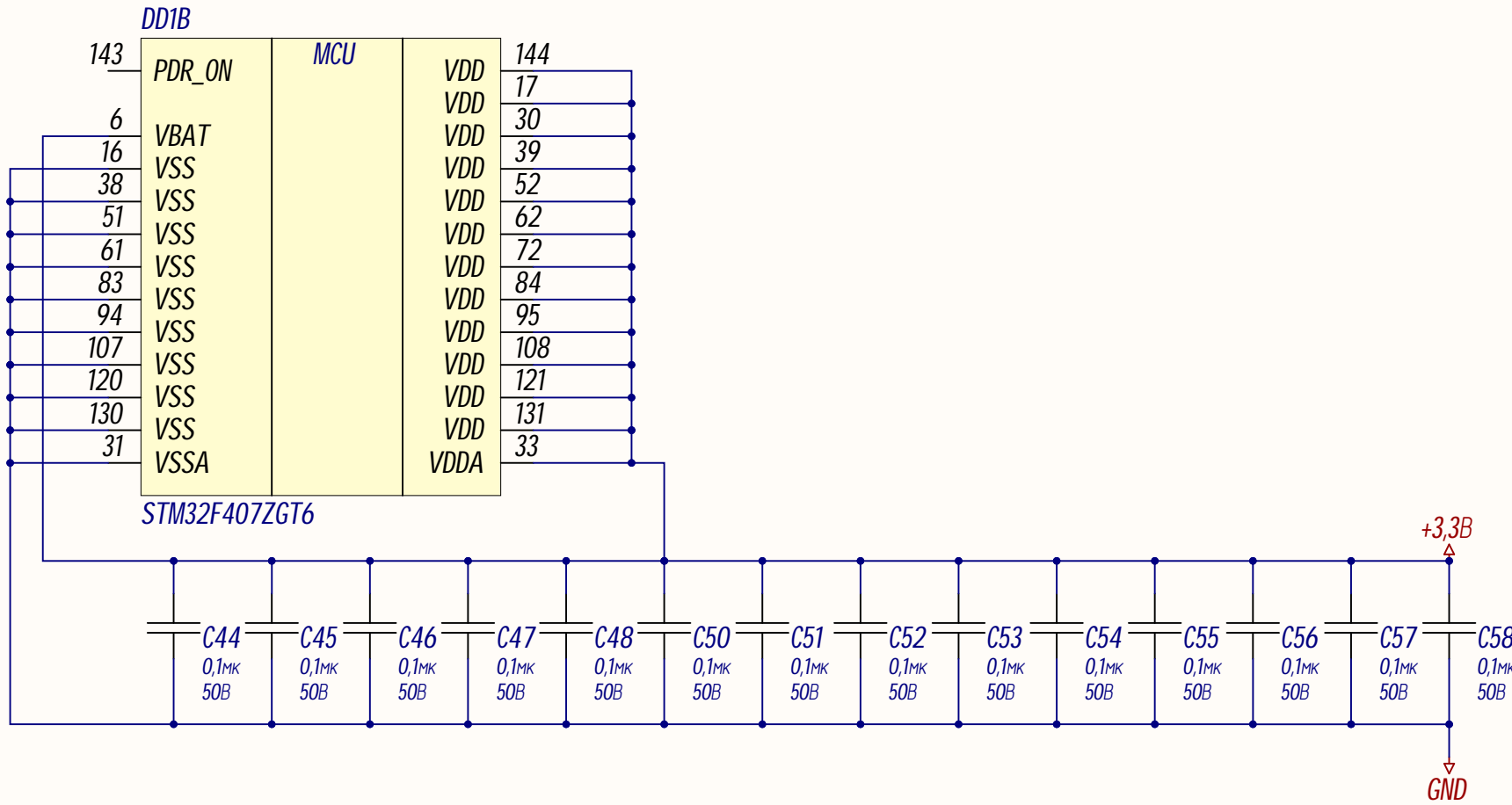
Micro-USB

SWD прозрaммaтopa





Выход земли и питания



Объединение питания st-link и основного МК

Из Altium Designer есть возможность выгрузить полную 3D-модель готовой платы в формате STEP.

3D-модели платы управления моторами (Рисунок 39), материнской платы (Рисунок 40) и платы датчиков (Рисунок 41) в Altium Designer представлены ниже.

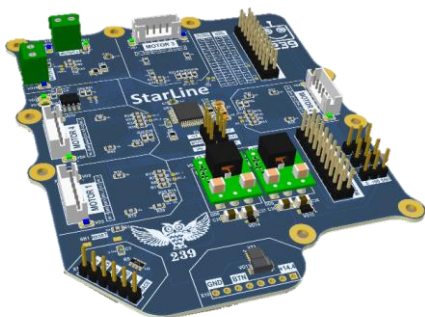


Рисунок 39. Плата управления моторами



Рисунок 41. Плата датчиков

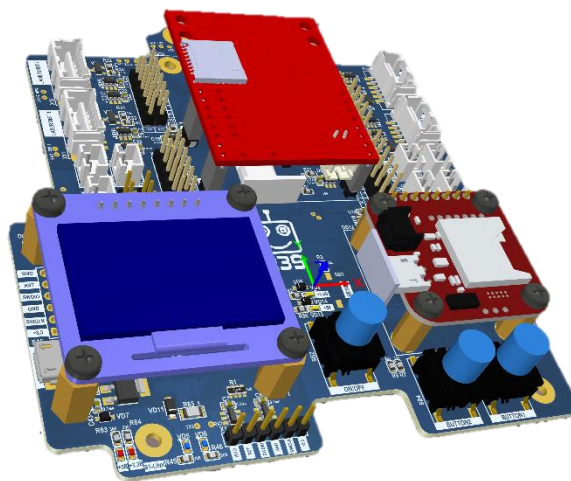


Рисунок 40. Материнская плата

2.2.4. Изготовление печатных плат и монтаж компонентов

Четырехслойные платы невозможно изготовить в школе вручную, поэтому их производство было заказано в ООО «Резонит» [25]. Для этого оформлены документы для производства в виде GERBER-файлов.

После изготовления плат (Рисунок 42) произведен монтаж компонентов с помощью паяльника и термофена (Рисунок 43). Также использовалась другая технология: из пленки был вырезан трафарет (на основе DXF-файла, где обозначены все контактные площадки), через него нанесена паяльная паста, расположены компоненты, и затем плата нагрета на плитке.



Рисунок 42. Платы, полученные с производства

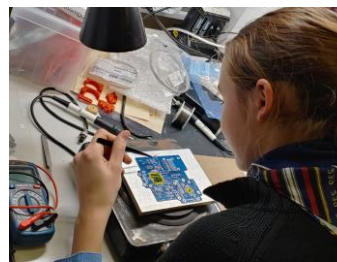


Рисунок 43. Монтаж компонентов

2.3. Программное обеспечение робота

2.3.1. Выбор среды разработки и языка программирования

Для программирования основного микроконтроллера STM32 решено использовать среду Arduino IDE с оболочкой под STM32 [26] и язык C++. Для программирования модуля SparkFun IMU также выбрана Arduino IDE. А для программирования модуля камеры целесообразно использовать OpenMV IDE и язык MicroPython, предназначенную специально для данного модуля камеры [27].

2.3.2. Общее описание

Есть несколько программируемых устройств.

Камера OpenMV программируется на MicroPython в среде разработки OpenMV IDE. Камера в данном случае – самостоятельный модуль, который работает независимо. Микроконтроллер на плате OpenMV считывает информацию с камеры и обрабатывает ее. STM32 на плате OpenMV по UART отправляет данные в некотором формате отправляет данные на основной микроконтроллер.

Аналогично работает модуль SparkFun 9DoF Razor IMU M0, опрашивая датчик акселерометр-гироскоп-магнитометр. Этот модуль также является программируемым. На основе примеров производителя реализовано определение углов тангажа и рыскания и отправки данных на основной микроконтроллер по UART.

Основной микроконтроллер STM32F407ZGT6 опрашивает периферию и управляет всеми двигателями.

2.3.3. Компьютерное зрение

Для распознавания линии изображение, получаемое с камеры, разбивается на несколько горизонтальных полос. Каждая полоса создается как отдельный регион интереса (roi). Далее в этом регионе интереса проводится поиск черных областей с помощью встроенной функции `find_blobs()`. Для каждой области определяется ее центр по осям x и y и ширина области (Рисунок 44).



Рисунок 44. Поиск фрагментов линии

Для каждой вычисляется отклонение по формуле:

$$deviation = \left(\frac{size_x}{2} - center_x \right) * width$$

$size_x$ – ширина изображения по горизонтали

$center_x$ – x-координата черной области в регионе интереса.

$width$ - ширина области

$deviation$ – величина, характеризующая расположение данной черной области на изображении

Домножение на ширину области нужно для того, чтобы более протяженные по горизонтали области (на резких поворотах) влияли на движение робота сильнее.

Теперь о распознавании зеленых меток.

Для начала ищем зеленые области с помощью функции *find_blobs*.

Далее с помощью функции *min_corners* определяем вершины минимального описывающего прямоугольника (он отрисовывается на изображении желтыми точками).

Далее вершины сортируются по возрастанию y-координаты.

Далее вычисляются координаты точки – середины отрезка, соединяющего 2 верхние вершины (Рисунок 45). И вычисляется угол поворота относительно вертикали по следующим формулам.

Далее определяются координаты точек справа, слева, сверху и снизу каждой метки. Вокруг каждой описывается окрестность в виде прямоугольника. В каждом из них проверяется наличие черных областей, если они есть, считается, что это черная линия, если их нет, то значит, что это белое поле (Рисунок 46).

$$x_{up} = \frac{(dist_{coeff} + 1) * (x_1 + x_2 - x_c)}{2 * dist_{coeff}}$$

$$y_{up} = \frac{(dist_{coeff} + 1) * (y_1 + y_2 - y_c)}{2 * dist_{coeff}}$$

x_{up} – x-координата точки сверху

y_{up} – y-координата точки сверху

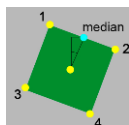


Рисунок 45. Середина верхней стороны

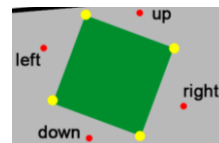


Рисунок 46. Определение координат точек с четырех сторон от квадрата

На изображении прямоугольник обозначается белым, если в нем найдены черные области, и черным – если их нет. Цвета по периметру метки кодируются 1 массивом из 4 элементов. 0 – белый, 1- черный. Далее в зависимости от конфигурации меток определяется направление поворота (Рисунок 47).

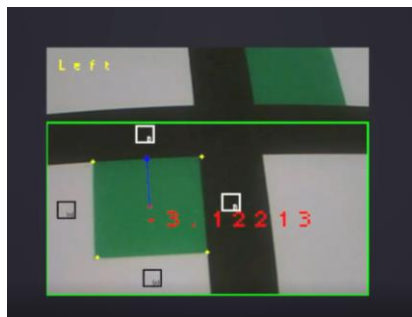


Рисунок 47. Определение направления поворота на перекрестке

Цифрами от 0 до 4 закодированы направления поворота.

Данные о линии и метках передаются от камеры на STM32 по протоколу UART.

Также с помощью компьютерного зрения определяются шарики и синий кубик. Используются встроенные в OpenMV функции компьютерного зрения для распознавания объектов по шаблонам (template_matching). Для этого делается несколько снимков объекта (я написала программу, которая делает их автоматически, нужно только перемещать камеру). А затем алгоритм «ищет» место на изображении, наиболее близкое к шаблону (в специальной метрике). Определяется сумма квадратов разностей цветов пикселей. То место, на изображении, где разница минимальна, оответсвует положению объекта. (Фотографии делаются с разных ракурсов и с разным увеличением на фоне полигона, чтобы можно было найти объект при разных расстояниях его до робота).

2.3.4. Конечный автомат

Алгоритм имеет структуру конечного автомата. Состояния меняются при срабатывании определенных условий (перекрестка, наличия препятствия и других).

С помощью кнопок можно заходить в меню, где можно переключаться в режимах просмотра значений со всех датчиков. Для кнопки импортирована специальная библиотека, с помощью которой можно гарантированно контролировать нажатия, избегаядребезга, а также задавать длинные и короткие нажатия.

Для визуализации работы алгоритма составлена диаграмма состояний при движении по линии (Рисунок 48).

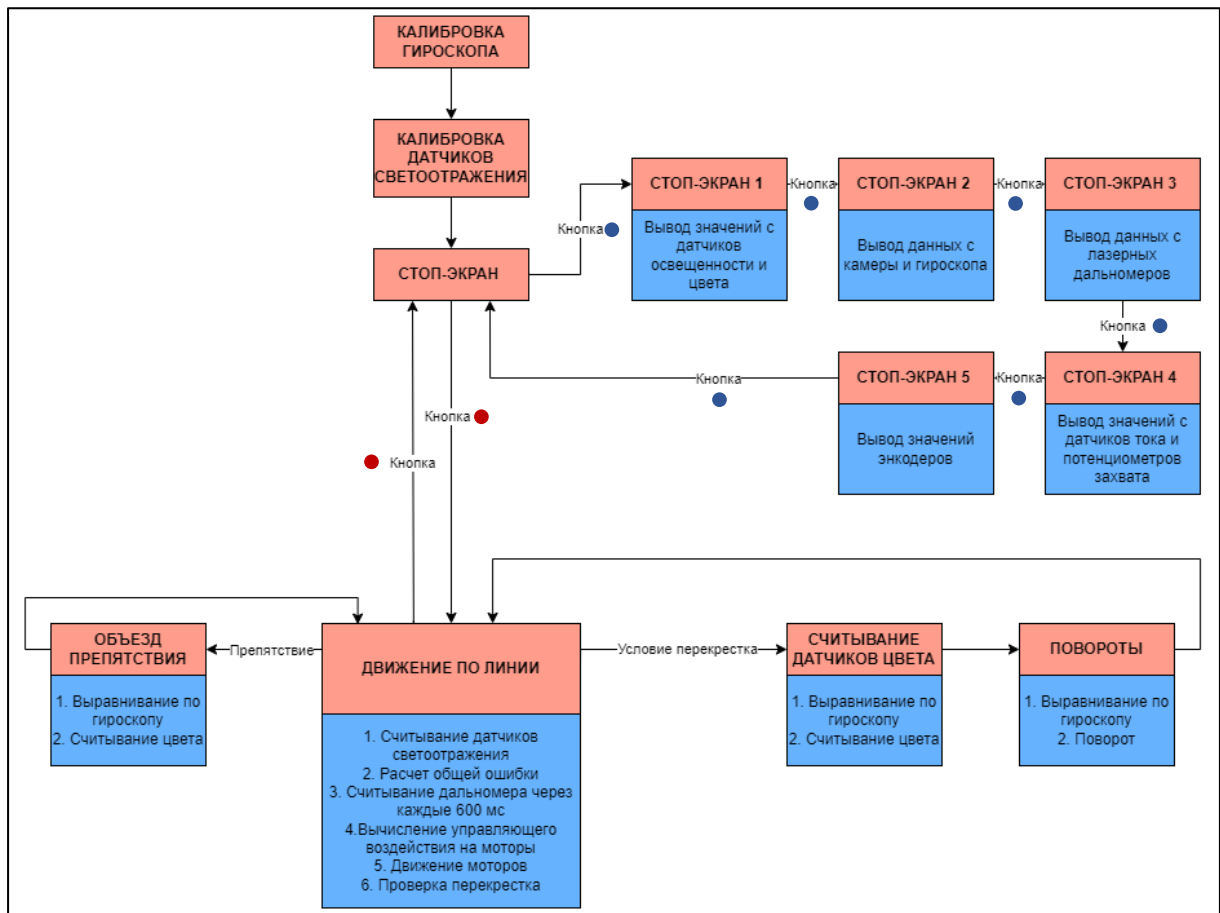


Рисунок 48. Диаграмма состояний

2.3.5. Управление захватным механизмом

Для того, чтобы захватное устройство открывалось в определенное положение, достаточно просто подавать ШИМ-сигнал определенного заполнения. Отсутствует необходимость контролировать степень открытия, т.к. регулятор для этого реализован на микроконтроллере на плате сервомотора, к которой подключен мотор лебедки.

Для того, чтобы захват принимал определенную высоту, реализован ПИД-регулятор. Значения с потенциометра переводятся масштабированием в значение высоты подъема.

Для защиты моторов от застреваний и перегорания предусмотрено считывание значения напряжения обратной связи по току с драйвера двигателя. При застревании мотора ток превышает максимально допустимый и ШИМ перестает подаваться на мотор. Приведен фрагмент функции движения в определенное положение.

Ошибка вычисляется как $h1 - target_position$, где $h1$ – текущая высота, а $target_position$ – требуемая высота.

```
while ((abs(err_h1) > max_err_h1) or (abs(err_h2) > max_err_h1)) {
    h1 = map(analogRead(POTENTIOMETER1), 0, 1023, 0,
```

```

max_target_position); //преобразование значений с АЦП в высоту
    h2 = map(analogRead(POTENTIOMETER2), 0, 1023, 0,
max_target_position);
    err_h1 = (h1 - target_position); //ошибка для регулятора
    integral1 = integral1 + (float)err_h1 * dt * ki_sliders; //И-составляющая
    err_h2 = (h2 - target_position);
    integral2 = integral2 + (float)err_h2 * dt * ki_sliders;
    float D1 = (err_h1 - prevErr1) / dt; //Д-составляющая
    float D2 = (err_h2 - prevErr2) / dt;
    prevErr1 = err_h1; //предыдущая ошибка
    prevErr2 = err_h2;
    float result1 = err_h1 * kp_sliders + integral1 + D1 * kd_sliders;
//управляющее воздействие
    float result2 = err_h2 * kp_sliders + integral2 + D2 * kd_sliders;
    sliders(-result2, result1); //движение серворезисторов
}

```

2.3.6. Управление скоростями моторов

Для контроля скорости моторов реализован ПИД регулятор.

Ошибка вычисляется как разность текущей скорости *current_speed1* и требуемой скорости *sp_rpm*.

Скорость в оборотах в минуту вычисляется по формуле:

$$speed = \left(\frac{\Delta encoders * 60000}{CPR * motors_delay} \right)$$

Где *CPR* – число импульсов на оборот колеса, *Δencoders* – разность показаний энкодеров на данном и предыдущем шаге, *motors_delay* – задержка, с которой пересчитываются скорости в миллисекундах. Функция вычисления требуемой мощности для достижения заданной скорости представлена ниже.

```

int vel1(float speed) {
    int sp_rpm = SpeedRPM1(speed); //вычисление текущей скорости
    current_speed1 = ((Enc1() - prev_enc1) * 60000 / ((float)CPR * MOTORS_DELAY));
    err1 = current_speed1 - (float)sp_rpm;
    up1 = SPEED_COEFF_P * err1; //П-составляющая
    ui1 = ui1 + SPEED_COEFF_I * err1; //И-составляющая
    ud1 = (err1 - err1_old) * SPEED_COEFF_D; //Д-составляющая
}

```

```

if (abs(err1) < ERR_LIM) {
    ui1 = 0;
}
u1 = up1 + ui1 + ud1; //общее управляющее воздействие
if (u1 > U_MAX) u1 = U_MAX; //ограничение
else if (u1 < -U_MAX) u1 = -U_MAX;
int v1 = map(sp_rpm + round(u1), 0, RPM1, 0, 255); //преобразование из об/мин в
значения ШИМ

prev_enc1 = Enc1(); //предыдущее значение энкодера
err1_old = err1;
return v1;
}

```

2.3.7. Определение углов

Для реализации поворотов на заданный угол используются показания гироскопа (угол рыскания). Реализован ПИД-регулятор.

Гироскоп выдает абсолютные углы, они могут быть от 0 до 360 градусов. Для примера представлена блок-схема алгоритма поворота на данный угол (Рисунок 49). В качестве ошибки для регулятора берется минимальное по модулю из значений разности углов, разности углов+360 и разности углов-360 (корректные переходы через 0).

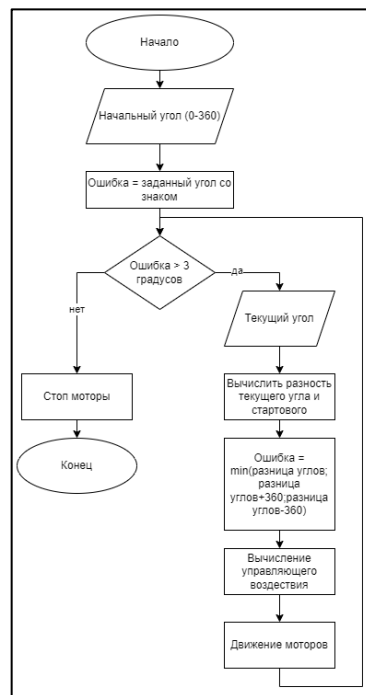


Рисунок 49. Блок-схема алгоритма поворота на данный угол

Также с помощью показаний гироскопа (угла тангажа) также определяется горка.

Скорость движения корректируется на горке, чтобы не допустить схода с линии.

2.3.8. Движение по линии и распознавание меток

Для каждого из 6 датчиков определяется степень освещенности. Для каждого датчика задается свой весовой коэффициент, для крайних датчиков он по модулю больше, тем для тех, которые близко к линии. Для симметричных датчиков коэффициенты противоположны по знаку.

Значения отмасштабированы в диапазон 0-1 (в зависимости от калибровки), чтобы при изменении освещенности не менялось управляющее воздействие.

Ошибка для ПИД-регулятора вычисляется следующим образом:

$$err = color1 * w[0] + color2 * w[1] + (1 - color3) * w[2] + (1 - color4) * w[3] + color5 * w[4] + color6 * w[5]$$

Набор весовых коэффициентов может выглядеть, например, так:

$$float w [6] = \{ -0.5, -0.3, -0.2, 0.2, 0.3, 0.5 \};$$

Далее – обычный ПИД-регулятор.

Метки распознаются двумя датчиками цвета. При нахождении датчиков освещенности над перекрестком датчики цвета находятся точно над метками, благодаря выравниванию по гироскопу. Определяется, зеленый или белый цвет под датчиком цвета. Цвет распознается с помощью вычисления Евклидова расстояния между цветами в цветовом кубе.

$$dist^2 = (green - green_0)^2 + (red - red_0)^2 + (blue - blue_0)^2$$

Red, green, blue - текущие значения, *red₀, green₀, blue₀* – значения на зеленом

Если расстояние меньше заданного значения, цвет можно считать зеленым.

2.4. Отладка. Тестирование. Модификация

Перед началом разработки данной версии робота был создан прототип (Рисунок 50). Его анализ представлен на с. 7. Все его недостатки были учтены и исправлены в итоговой версии робота.



Рисунок 50. Новый робот и прототип

Прототип был протестирован на соревнованиях RoboCup Russia Open 2022. Помимо недостатков, указанных в таблице анализа, были выявлены некоторые другие, например, не плоское дно и неправильно подобранное расстояние от дна робота до полигона: робот «садился» на горку.

Также я в составе команды выступила с докладом на научно-техническом семинаре на этих соревнованиях. Был представлен доклад «Опыт реализации продвинутых технических решений для мобильного робота». В таблице 3 представлен календарный график модификаций робота.

Тестирование проводится на Открытых соревнованиях Санкт-Петербурга по робототехнике в апреле 2023 года в соответствующей соревновательной категории.

Таблица 3 – Календарный график модификаций робота

Даты	Модификации
Май – июнь 2022	Разработка принципиальной схемы платы моторов, плата моторов; тестирование периферии на STM32F407.
Июль – август 2022	Разработка 3D-модели основы робота, сборки платформы с аккумулятором и моторами.
Сентябрь 2022	Разработка принципиальных схем материнской платы и платы датчиков; разработка 3D-модели захвата.
Октябрь 2022	Разработка материнской платы, платы датчиков; доработка 3D-модели (крепления датчиков); изготовление и сборка редукторов.
Ноябрь 2022	Сборка робота на пластиковой основе; доработка конструкции.
Декабрь 2022	Монтаж компонентов, подключение датчиков и моторов; программный запуск всей периферии, программирование движения по линии.
Январь 2023	Программирование движений захвата, движений по гироскопу (определение горки и повороты), программирование движений платформы; сборка робота на металлической основе.

3. ЗАКЛЮЧИТЕЛЬНЫЙ ЭТАП

3.1. Эстетический вид и качество робота

Робот получился эстетичным и эргономичным.

Для удобства отладки в конструкцию внедрен USB-хаб для подключения всех программируемых устройств к нему. Достаточно подключить удлинитель USB-кабеля к хабу, чтобы запрограммировать все 3 устройства.

На материнской плате на дополнительные разъемы выведены все неиспользованные пины микроконтроллера, что позволяет подключать дополнительные устройства. Также

разъемы для разных типов устройств разные, чтобы не было возможности перепутать при переподключении.

Для вывода отладочной информации есть OLED-дисплей, при этом можно переключаться в меню с помощью кнопок. В разных режимах меню отображаются показания всех датчиков.

Уменьшено количество проводов по сравнению с прототипом за счет использования межплатных соединений.

Для удобной замены аккумулятора крышка держится на петлях, а открывается при отпускании защелки, что является удобным при эксплуатации.

3.2. Новизна и креативность проекта

Проект является оригинальной разработкой. Конструкция робота и печатные платы разработаны самостоятельно, создан алгоритм именно для данного робота с данным набором датчиков и моторов. При создании робота реализованы нестандартные технические решения: планетарный редуктор внутри колеса, конструкция захватного механизма на основе механизма Iris и подъемника. Особенно в данном проекте хотелось бы отметить реализацию модуля серволебедки. При использовании концепции уже существующего решения (сервопривода) конструкции создан новый модуль только за счет переработки конструкции без дополнительных программных реализаций.

3.3. Практическая значимость

Робот создан для отладки алгоритмов навигации по напольной цветовой разметке с небольшими препятствиями, а также поиска и транспортировки небольших объектов. Поэтому он может использоваться как прототип промышленного робота, работающего по тем же принципам.

Соревнования RoboCup являются одним из способов для проверки и доработки новых технологий робототехники. Эти соревнования позволяют оценить прогресс, который был сделан в этой области, и поддерживают развитие образовательной робототехники. Опыт разработки может быть использован участниками сообщества RoboCup.

3.4. Результат

В результате проделанной работы была достигнута поставленная цель. Создан робот для отладки алгоритмов навигации по разметке, распознаванию и транспортировки объектов. Алгоритмы протестированы на учебном соревновательном полигоне RoboCup Rescue Line.

В процессе работы над проектом я познакомилась с САПР для проектирования печатных плат Altium Designer, изучила его и использовала его для разработки собственных

печатных плат. Также я изучила ГОСТы, касающиеся составления принципиальных и структурных схем. Также я улучшила навыки 3D-моделирования: сборка робота в САПР Autodesk Inventor создана полностью со всем крепежом, чего я ранее никогда не делала.

Я довольна тем, что у меня получилось учесть все недостатки прототипа и создать эстетичного робота с требуемыми характеристиками.

4. ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ

1. ГОСТ 60.0.0.2-2016 Роботы и робототехнические устройства. Классификация
2. Роботы компании INTEC [сайт]. URL: <https://agvsystems.ru/>
3. Соревнования RoboCup [сайт]. URL: <https://junior.robocup.org>
4. Видео-описание робота команды Školska Knjiga CRO Team [Электронный ресурс]//Virtual Presentation Event 2020 [сайт]. URL: <https://rescue.rcj.cloud/events/2020/virtual/video/junior17>
5. Видео-описание робота команды DoubleD. Robocup Asia-Pacific Virtual 2020 [Электронный ресурс]. URL: https://www.youtube.com/playlist?list=PLFg-N_d0BOGcpRlcJYnPkeo_SfmWtCnpm
6. Virtual Presentation Event 2020 (видео роботов различных команд Rescue Line) [сайт]. URL: <https://rescue.rcj.cloud/events/2020/virtual/overview>
7. Iris-диафрагма [Электронный ресурс]. URL: <https://yandex.ru/video/touch/preview/589695745796158219>
8. Регламент соревнований RoboCup Rescue Line [Электронный ресурс]// RoboCup Junior [сайт]. URL: <https://junior.robocup.org/wp-content/uploads/2023/01>
9. Autodesk Inventor [сайт]. URL: <https://www.autodesk.com/products/inventor/overview?term=1-YEAR&tab=subscription>
10. LimaLux [сайт]. URL: limalux.ru
11. 12V DC Motor 350RPM w/Encoder (12kg*cm) [сайт]. URL: <https://www.dfrobot.com/product-1462.html>
12. Линейные потенциометры ALPS RS60N11M9A0E [сайт]. URL: <https://tech.alpsalpine.com/e/products/detail/RS60N11M9A0E/>
13. Переменный резистор 3296W-103 [Электронный ресурс]. URL: <https://www.promelec.ru/fs/sources/31/5c/bc/4d/d8e3c0712e01acd69e206e51.pdf>
14. ИК-светодиод L-7113F3C [Электронный ресурс]. URL: <https://static.chipdip.ru/lib/283/DOC000283279.pdf>
15. Фототранзистор L-7113P3C [Электронный ресурс]. URL: <https://static.chipdip.ru/lib/195/DOC000195179.pdf>

16. Драйвер светодиодов CAT4237 [Электронный ресурс]. URL: <https://www.onsemi.com/pdf/datasheet/cat4237-d.pdf>
17. Микросхема TCS34725 [Электронный ресурс]. URL: <https://cdn-shop.adafruit.com/datasheets/TCS34725.pdf>
18. Дальномер VL53L0X [сайт]. URL: <https://www.st.com/en/imaging-and-photonics-solutions/vl53l0x.html>
19. Модуль SparkFun 9DoF IMU Razor M0 [сайт]. URL: <https://learn.sparkfun.com/tutorials/9dof-razor-imu-m0-hookup-guide/all#resources--going-further>
20. Модуль камеры OpenMV [сайт]. URL: <https://openmv.io/collections/products/products/openmv-cam-h7-r2> OpenMV документация
21. Цифровой дальномер GP2Y0D810 [Электронный ресурс]. URL: <https://www.alldatasheet.com/datasheet-pdf/pdf/412687/SHARP/GP2Y0D810Z0F.html>
22. OLED дисплей [Электронный ресурс]. URL: <https://static.chipdip.ru/lib/663/DOC001663940.pdf>
23. STM32F407 [сайт]. URL: <https://www.st.com/en/microcontrollers-microprocessors/stm32f407-417.html#overview>
24. Altium Designer [сайт]. URL: <https://www.altium.com/ru/altium-designer/>
25. ООО «Резонит» [сайт]. URL: rezonit.ru
26. STM32duino. Библиотеки Arduino IDE для STM32 [сайт]. URL: <https://www.stm32duino.com>
27. Документация по MicroPython и функциям OpenMV IDE [сайт]. URL: <https://docs.openmv.io/OpenMV-MicroPython-Documentation>

ПРИЛОЖЕНИЕ А

Таблица А.1 – Список компонентов материнской платы

Наименование	Номинал/комментарий	Посадочное место	Кол-во
Микроконтроллер	STM32F103CB	LQFP48_H	1
Микроконтроллер	STM32F407ZGT6	LQFP-144	1
Операционный усилитель	LMV358	SOIC8	1
ESD защита	TPD2S017	SOT-23	23
Дисплей	OLED дисплей 128*64	PLS-8	1
Камера	OpenMV		1
Модуль гироскопа	SparkFun 9DoF IMU	PLS-8	1
Стабилизатор напряжения	LM1117	SOT-223	1
Предохранитель самовосстанавливающийся	MFPSMF050X	SMD0805	2
N-канальный полевой транзистор	IRLML0030TRPBF	SOT-23	4
Кварцевый резонатор	12 МГц	KX-7T	1
Кварцевый резонатор	8 МГц	KX-7T	1

Наименование	Номинал/комментарий	Посадочное место	Кол-во
Диод Шоттки	MBR2H100SFT3G	SOD-123FL	3
Стабилитрон 5В	MMSZ5231B-TP	SOD-323	1
Стабилитрон 3,3В	MM3Z3V3B	SOD-323	1
Диод Шоттки	BAT60JFILM	SOD-323	1
Керамический конденсатор	0,1 мкФ	SMD0603	50
Керамический конденсатор	2,2 мкФ	SMD0805	2
Керамический конденсатор	0.1 мкФ	SMD0603	1
Керамический конденсатор	12 пФ	SMD0805	2
Керамический конденсатор	20 пФ	SMD0603	2
Керамический конденсатор	100 мкФ	SMD1206	1
Резисторная сборка	10 кОм	SMD0603x4	15
Резисторная сборка	0 Ом	SMD0603x4	4
Резисторная сборка	0 Ом	SMD0603x4	10
Резистор	10 кОм	SMD0603	17
Резистор	300 Ом	SMD0603	6
Резистор	1М Ом	SMD0603	2
Резистор	330 Ом	SMD0603	2
Резистор	100 Ом	SMD0603	16
Резистор	0 Ом	SMD0603	6
Резистор	100 кОм	SMD0603	2
Резистор	4,7 кОм	SMD0603	1
Резистор	1,5 кОм	SMD0603	1
Резистор	39 кОм	SMD0603	1
Резистор	360 Ом	SMD0603	2
Резистор	0 Ом	SMD1206	1
Кнопка SMD	2,5*3*6mm H DIP	2.5*3*6mm	1
Кнопка SMD	5GSH935+1SS09-10.4	5GSH935+1SS09-10.4	3
Светодиод SMD	зеленый	SMD0603	2
Светодиод SMD	синий	SMD0603	2
Светодиод SMD	желтый	SMD0603	2
Светодиод SMD	красный	SMD0603	2
PLS-6	6 вилка	PLS-6	1
PLS-3	3 вилка	PLS-3	2
PLS-20	2*10 розетка	PBS-2-10	2
JST PH2.0-2	2 вилка	PH2.0-2pin	4
JST PH2.0-3	3 вилка	PH2.0-3pin	3
CLIK-Mate-6	6 розетка	CLIK-Mate-6 1,5мм	4
PLS-6	6 розетка	PLS-6 угловой	1
Micro-USB	5 розетка	Micro-USB	1
PLS-12	2*6 розетка	PLS-2*6	3
FPC-20	20 контактов	FPC-20	1
PLS-10	2*5 розетка	PLS-2*5	1