

ОЛИМПИАДА ШКОЛЬНИКОВ «ШАГ В БУДУЩЕЕ»

**НАУЧНО-ОБРАЗОВАТЕЛЬНОЕ СОРЕВНОВАНИЕ
«ШАГ В БУДУЩЕЕ, МОСКВА»**

№60246

регистрационный номер

Робототехнические системы

название факультета

СМ7

название кафедры

Разработка и испытание оригинального

полетного контроллера на базе STM32

название работы

Автор:

Мельников Юрий Сергеевич

фамилия, имя, отчество

ГБОУ Школа 67, 11 класс

наименование учебного заведения, класс

Научный руководитель:

Тимофейчик Данил Сергеевич

фамилия, имя, отчество

Студент 3 курса СМ6

место работы

звание, должность

подпись научного руководителя

Москва - 2025

РАЗРАБОТКА И ИСПЫТАНИЕ ОРИГИНАЛЬНОГО ПОЛЕТНОГО КОНТРОЛЛЕРА НА БАЗЕ STM32

АННОТАЦИЯ

В данной научно-исследовательской работе рассматривается процесс проектирования, программной реализации и испытания оригинального полетного контроллера для беспилотных летательных аппаратов (БПЛА) мультироторного типа на базе микроконтроллера STM32F411.

Научная и инженерная новизна работы заключается в переходе от монолитных схем к модульной трехслойной архитектуре стека электроники. Такое решение позволило изолировать измерительный комплекс от силовых линий питания, а также распределить вычислительные задачи между микроконтроллерами STM32 и ESP32. Возможно создание 4-12 моторных мультироторных коптеров.

В ходе выполнения работы были достигнуты следующие результаты:

- Разработаны и изготовлены печатные платы вычислительного ядра, коммуникационного модуля и контроллера ШИМ в среде EasyEDA.
- Реализовано программное обеспечение на языке C++, включающее комплементарный фильтр данных инерциальной системы и каскадный ПИД-регулятор с защитой от интегрального насыщения
- Спроектирован и собран пульт дистанционного управления на базе технологии LoRa с возможностью использования оптимизированного бинарного протокола передачи данных.
- Проведены стендовые испытания на балансировочном стенде, подтвердившие работоспособность алгоритмов стабилизации.

Разработанный стек электроники обладает полным набором навигационных датчиков (акселерометр, гироскоп, барометр, компас и GPS), что создает базу для внедрения алгоритмов полностью автономного полета.

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ ИСПОЛЬЗУЕМЫХ СОКРАЩЕНИЙ	5
ПЕРЕЧЕНЬ ИСПОЛЬЗУЕМОГО ПО	5
ВВЕДЕНИЕ	5
Проблематика исследования	5
Актуальность работы	6
Объект и предмет исследования	7
Цель работы	7
Задачи исследования:	7
ОСНОВНАЯ ЧАСТЬ.....	8
1 РАЗРАБОТКА СТЕКА ЭЛЕКТРОНИКИ	8
1.1 Структура стека и разделение задач	8
1.2 Слои стека	8
2 СОЗДАНИЕ ПЛАТФОРМЫ ДЛЯ ИСПЫТАНИЯ КОНТРОЛЛЕРА.....	12
2.1 Расчеты и подбор компонентов.....	12
2.2 Проектирование рамы.....	13
3 РАЗРАБОТКА ПУЛЬТА РАДИОУПРАВЛЕНИЯ.....	14
3.1 Аппаратная реализация	14
3.2 Обработка сигналов и формирования пакета	14
3.3 Оптимизация настроек LoRa модуляции	15
4 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ (ПО КОНТРОЛЛЕРА).....	15
4.1 Фильтрация данных	15
4.2 ПИД Регуляторы.....	16
4.3 Защита от накопления ошибки	17
4.4 Матрица микширования	17
5 ИСПЫТАНИЯ И РЕЗУЛЬТАТЫ.....	17
5.1 Тестовый стенд.....	17
5.2 Результаты летных испытаний.....	18
ЗАКЛЮЧЕНИЕ	18
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	20
ПРИЛОЖЕНИЯ	21
Приложение А.....	21
Приложение Б	22
Приложение В.....	23

Приложение Г	24
Приложение Д.....	25
Приложение Е	26
Приложение Ж.....	27
Приложение З	28
Приложение И.....	29

ПЕРЕЧЕНЬ ИСПОЛЬЗУЕМЫХ СОКРАЩЕНИЙ

БПЛА – беспилотный летательный аппарат.

IMU – инерциальная навигационная система (Inertial Measurement Unit).

ESC – Electronic Speed Controller, регулятор оборотов мотора, в этой работе с управлением ШИМ сигналом.

ПИД – пропорционально-интегрально-дифференциальный (регулятор)

ШИМ (PWM) – широтно-импульсная модуляция.

LoRa – Long Range, тип модуляции радиосигнала для передачи на большие расстояния с высокой устойчивостью к помехам.

ЛУТ – Лазерно-Утюжная Технология, технология изготовления печатных плат путем защиты будущих медных дорожек тонером принтера, который припекают утюгом, при последующем травлении.

ПЕРЕЧЕНЬ ИСПОЛЬЗУЕМОГО ПО

PlatformIO – интегрированная среда разработки (IDE) в Visual Studio Code для написания программного кода для микроконтроллеров на языке C++ в среде Arduino.

EasyEDA – система автоматизированного проектирования (САПР) для разработки принципиальных электрических схем и трассировки печатных плат.

Компас-3D – САПР программа для трехмерного проектирования CAD-моделей рамы и конструктивных элементов дрона, создания их чертежей.

ВВЕДЕНИЕ

Проблематика исследования

На сегодняшний день рынок беспилотных летательных аппаратов перенасыщен готовыми контроллерами. Почти все они строятся по принципу

монолитной архитектуры (All-In-One), где процессор, датчики и силовые цепи находятся на одной многослойной плате. Программы для таких контроллеров (например, BetaFlight или ArduPilot) – это огромные массивы высокоуровневого кода, которые часто работают под управлением операционных систем.

Для рядового пользователя или FPV-пилота это удобно: система работает «из коробки» и не требует сложной настройки. Но для разработчика, который хочет изучать алгоритмы полета, такие контроллеры превращаются в «черный ящик». Скрытность процессов за тяжелыми библиотеками делает почти невозможным точный контроль времени выполнения команд или создание собственных функций. Разработчик не может гарантировать, сколько именно микросекунд займет опрос датчика, а сколько – расчет ПИД-регулятора, что критически важно при создании своих собственных алгоритмов стабилизации или навигации.

Актуальность работы

Чтобы по-настоящему понимать и контролировать физику полета, инженерам и исследователям необходима полностью открытая и прозрачная аппаратная платформа. Большинство заводских полетных контроллеров имеют жесткую, неизменяемую конструкцию, которая не позволяет адаптировать плату под нестандартные задачи или гибко менять измерительные приборы. Актуальность данного проекта заключается в разработке оригинального полетного контроллера с нуля: от самостоятельного подбора компонентной базы до проектирования и разводки собственных печатных плат на базе микроконтроллера STM32. Это дает полный контроль над аппаратной частью устройства. Кроме того, в проекте используется модульная архитектура (сборка из нескольких плат). Это решение позволяет сделать сборку меньше и частично устраняет влияние сильных электромагнитных помех от силовых линий моторов на работу

чувствительных навигационных датчиков, так как они физически разнесены на разные уровни стека.

Объект и предмет исследования

Объектом исследования являются системы автоматического управления малыми квадрокоптерами.

Предметом исследования выступает аппаратно-программный комплекс полетного контроллера: его архитектура, интерфейсы передачи данных (I2C, UART) и математические алгоритмы пространственной стабилизации.

Цель работы

Разработать, собрать и испытать оригинальный полетный контроллер на базе микроконтроллера STM32. Система должна иметь модульную архитектуру, модули, соединяемые шлейфами, работать в режиме реального времени и обладать встроенной системой дальней радиосвязи при помощи LoRa технологии. Также в перспективе предполагается удобная настройка дрона по Wi-Fi через веб-страницу.

Задачи исследования:

1. Спроектировать модульную архитектуру контроллера, чтобы физически разделить измерительные и силовые цепи.
2. Подобрать электронные компоненты, разработать схемы и развести печатные платы в среде EasyEDA.
3. Изготовить и отладить полученные платы, провести тестирование на работоспособность
4. Провести математические расчеты и практические замеры интерфейсов (I2C, UART), чтобы цикл управления успевал выполняться с частотой 400 Гц.
5. Написать программу на языке C++, реализовав программный и аппаратный фильтры данных с датчиков и каскадный ПИД-регулятор.
6. Спроектировать и изготовить пульт управления и раму квадрокоптера.

7. Провести тесты на стенде и в воздухе для настройки алгоритмов стабилизации.

ОСНОВНАЯ ЧАСТЬ

1 РАЗРАБОТКА СТЕКА ЭЛЕКТРОНИКИ

1.1 Структура стека и разделение задач

Главное отличие разработанного контроллера от большинства заводских решений – это четкое разделение вычислительных задач. Вместо того чтобы перегружать один процессор и радиоэфиром, и математикой, и генерацией ШИМ-сигналов, функции разделены между специализированными чипами.

Архитектура выполнена в виде трехслойного стека («башни») диаметром 55 мм. Такое расположение решает две задачи: позволяет легко заменять модули в случае поломки и обеспечивает электромагнитную изоляцию. Силовые провода находятся на самом нижнем уровне, максимально далеко от датчиков.

Для питания логики используется встроенный линейный стабилизатор напряжения с 5В до 3.3 В, а для понижения до 5 для радиомодуля используется импульсный понижающий преобразователь. В Приложении А показана структурная схема стека, в Приложениях Б-Г принципиальные схемы каждого из слоев.

1.2 Слои стека

1.2.1 Слой STM32. Вычислительное ядро и датчик

Верхняя плата стека – это «мозг» системы. Здесь установлен микроконтроллер STM32F411RET6[2][3] (до 100 МГц). В нем есть аппаратный блок для вычислений с плавающей точкой. Это важно, потому что расчет углов и интегральных сумм ПИД-регулятора требует быстрой работы с дробными числами. Этот микроконтроллер занимается только математикой полета и опросом датчиков, отправкой команд для второго микроконтроллера.

На плате размешены следующие датчики и модули, необходимые для полетов:

1. Инерциальная система (IMU) Bosch BMI088[7] в отличие от популярного, но старого MPU6050, показывает отличные результаты при работе в условиях сильных вибраций. Является 6-осевым гироскопом-акселерометром, имеет встроенные цифровые фильтры низких частот. Датчик позволяет получать 3 угловых скорости и 3 ускорения по каждой оси соответственно. Подключен по SPI для большей скорости чтения.
2. Барометр Bosch BMP280 используется для измерения высоты. По нему планируется реализовать удержание высоты. В нем также есть встроенный фильтр (IIR-фильтр), который сглаживает скачки давления от воздушных потоков пропеллеров. Проблема задувания воздуха пропеллерами решается при помощи добавления кусочка поролона перед датчиком.
3. Магнитометр HMC5883L измеряет магнитное поле Земли по 3 осям для коррекции данных оси рыскания, т.к. иного способа получить абсолютный угол, в отличие от связки гироскопа и акселерометра для получения углов крена и тангажа, нет. Нужен преимущественно для автономных полетов.
4. GPS модуль u-blox NEO-M8N[6] принимает сигналы от систем GPS и ГЛОНАСС с частотой до 10 Гц . На плате предусмотрена связь по UART на частоте 115200 бод, доступны протоколы передачи NMEA и бинарный u-blox. В текущей конфигурации дрона для полетов в помещении не используется.

Наличие барометра, компаса и GPS закладывает аппаратную базу для того, чтобы в будущем реализовать автономные полеты со сбором данных на MicroSD карту или же отправку по радиоканалу.

1.2.2 Слой ESP32. Радиоканал и контроль питания

Средний уровень стека отвечает за связь. Здесь работает связка микроконтроллера ESP32-C3[4] и мощного радиомодуля EByte E22[5] (LoRa, 1 Вт) с частотой вещания 433 МГц. Возможна установка версии на более высокие частоты.

ESP32 берет на себя всю долгую работу, он общается с LoRa-модулем, принимает сложный пакет, проверяет его на ошибки и формирует короткий, чистый сигнал управления (6 байт). Только после этого он отправляется его на STM32 по UART. Такое решение принято из-за нехватки места для размещения крупного радиомодуля на плате с STM32. Если же располагать на второй плате без второго микроконтроллера, нужно вести 9 сигнальных линий от радиомодуля, 3 из которых работают на высоких частотах, через шлейф. Это было бы громоздко и имело высокий шанс перестать работать от единственного потерянного контакта.

В дополнение к обработке данных с радиомодуля, ESP имеет 4 МБ встроенной flash памяти, часть которой доступна для записи различных настроек, и поддержку WiFi.

Анализ задержек интерфейса UART:

Чтобы команды с пульта не запаздывали, необходимо грамотно выбрать скорость шины UART между ESP32 и STM32. Теоретическое время передачи рассчитывается по формуле: $T = \frac{N \cdot 10}{Baudrate}$ (мс), где N – количество байт в пакете, 10 (бит) – 8 бит данных + 1 старт бит + 1 стоп бит, Baudrate – скорость шины в бодах.

Для проверки расчетов проведены замеры с помощью аппаратных таймеров микроконтроллера (пакет 50 байт).

Пакет, байт	Скорость, бод	9600	19200	38400	57600	115200	250000	500000	1000000
50	Эксп. задержка, мс	52,87	26,33	13,16	8,78	4,39	2,12	1,06	0,77

	Теор. задержка, мс	52,08	26,04	13,02	8,68	4,34	2,00	1,00	0,50
	Отклонение, %	1,49	1,10	1,06	1,13	1,13	5,66	5,66	35,06
6	Эксп. задержка, мс	6,36	3,18	1,59	1,07	0,53	0,26	0,13	0,09
	Теор. задержка, мс	6,25	3,13	1,56	1,04	0,52	0,24	0,12	0,06
	Отклонение, %	1,73	1,73	1,73	2,65	1,73	7,69	7,69	33,33

Таблица 1. Сравнительный анализ задержек передачи по UART

В реальности рабочий пакет команд занимает всего 6 байт. Опираясь на полученные данные, была выбрана скорость 115200 бод. Передача 6 байт на такой скорости займет всего 0,53 мс. Учитывая, что весь цикл управления дроном длится 2,5 мс, время в 0,53 мс оставляет запас для вычислений. Для большего уменьшения задержек STM сперва ждет накопления 6 байт в буфере и только потом производит чтение.

1.2.3 Слой ШИМ- контроллера. Управление моторами

На нижней плате установлен чип PCA9685[8], который общается с STM32 по шине I2C и генерирует ШИМ-сигналы частотой 400 Гц для ESC моторов. Использование отдельного чипа дает возможность простого подключения до 12 моторов к одному контроллеру без изменения главной платы. Однако в текущей конфигурации платы выведено лишь 6 каналов из 12 доступных (4 заняты иными функциями), что исправляется весьма несложно.

Выбор частоты шины I2C:

Моторы должны получать новые команды 400 раз в секунду – частота ПИД-цикла. Был проведен анализ времени передачи пакета для обновления 4 каналов:

- В стандартном режиме на частоте 100 кГц передача занимает около 1,35 мс. Это занимает больше половины цикла в 2,5 мс. Важно, что передача блокируют все процессы в главном цикле, оставляя меньше половины цикла на опрос датчиков, проведение расчетов и подготовки команд.
- Переход шины в быстрый режим работы на частоте 400 кГц сократил время передачи до 0,38 мс. Таким образом остается около 2,1 мс на опрос

датчиков, который тоже ускорился за счет ускорения шины, чего уже более чем достаточно.

2 СОЗДАНИЕ ПЛАТФОРМЫ ДЛЯ ИСПЫТАНИЯ КОНТРОЛЛЕРА

2.1 Расчеты и подбор компонентов

Для испытания электроники был спроектирован компактный квадрокоптер. Малые дроны обладают низкой инерцией и высокой резкостью, что делает их настройку сложной и интересной инженерной задачей. Масса электронного стека составляет около 45 г. На раму закладывается около 40-50 граммов, на 4 ESC около 40 граммов, около 120 на аккумулятор. Ожидаемая взлетная масса аппарата ограничена 280 г.

Для стабильного полета тяговооруженность (отношение максимальной тяги всех моторов к весу дрона) должна быть не менее 2:1. Оценка тяги моторов производится по следующей формуле:

$$T = \frac{D^3 * P * RPM^2 * \sqrt{\frac{N}{2}} * 28,35}{10^{10}} \quad \text{где } T - \text{тяга в граммах, } D - \text{диаметр винта в дюймах, } P - \text{шаг винта в дюймах, } RPM - \text{частота вращения в об/мин, } N - \text{число лопастей винта, } 28,35 - \text{коэффициент перевода из унций в граммы.}$$

Для выбранных бесколлекторных моторов формата 1404 3500 KV при использовании 2S Li-Po аккумулятора (7.4 В) максимальные обороты составляют 25900 об/мин, реально же под нагрузкой будет около 70-80% от этого значения, то есть 19500 об/мин. Винты имеют диаметр и шаг по 3 дюйма и 3 лопасти. Подставим данные в формулу:

$$T = \frac{3^3 * 3 * 19500^2 * \sqrt{\frac{3}{2}} * 28,35}{10^{10}} = 109 \text{ (г)}$$

Один мотор дает около 109 граммов тяги в среднем или 436 граммов на все 4. Это дает тяговооруженность примерно 1.5, что позволит летать, но будет затруднительно. Для 3S аккумулятора тяга одного винта составит 238 граммов, то есть 955 граммов на все 4 мотора и тяговооруженность 3.4. Так как энерговооруженности 2S батареи недостаточно, будет использоваться 3S.

Каждый мотор потребляет около 8 А на полном газу и около 2-4 А при обычном полете. Для батареи емкостью 1850 мА*ч получится около 3.5 минут полета на полной мощности или 10-15 минут на средней. Исходя из максимального тока выбирается токоотдача батареи. Максимальный ток с запасом – 40 А. Следовательно для батареи емкостью 1850 мА*ч необходима токоотдача от 22С.

2.2 Проектирование рамы

Рама смоделирована в САПР «Компас-3D» и напечатана из ABS-пластика. Она имеет посадочные места под 4 мотора и полетный контроллер. Также сделаны ножки, чтобы аккумулятор, крепящийся снизу, не бился о землю при посадке. Их видно на фото дрона. Для упрощения коммутации проводов в центр рамы сделано отверстие. Чертеж рамы показан в Приложении Д.

Текущая форма лучей имеет расширение к центру рамы. Такое решение было принято для упрощения изготовления и проектирования, прочности. С точки зрения аэродинамики это решение не является оптимальным, так как плоская поверхность создает аэродинамическое затенение потоку воздуха от пропеллеров, снижая эффективную тягу.

2.3 Виброразвязка

Плата с PCA9685 закреплена жестко на раме, так как она не чувствительна к вибрациям от нее. Электронный стек, состоящий из главного слоя и слоя связи, установлен через демпфер из поролона, так как высокочастотные вибрации от моторов вызывают неконтролируемый шум акселерометра и срыв стабилизации, что было подтверждено тестированиями. Механический демпфер решает эту проблему на физическом уровне. Также, как было сказано ранее, используются аппаратные фильтры датчиков и программные непосредственно при получении данных.

3 РАЗРАБОТКА ПУЛЬТА РАДИОУПРАВЛЕНИЯ

3.1 Аппаратная реализация

Для тестирования системы разработан собственный пульт дистанционного управления. Устройство собрано на печатной плате, изготовленной ЛУТ, и имеет корпус из ABS-пластика с выводом SMA типа «папа» для антенны. В Приложении Е показаны внешний вид пульта и его принципиальная схема.

Основа пульта – Amperka Iskra Nano Pro (аналог Arduio Nano), считывающая показания двух джойстиков и двух тумблеров. Данные выводятся на I2C 128x64 OLED-экран. Передатчиком выступает модуль Ai-Thinker LoRa Ra-01 на частоте 433 МГц с широконаправленной антенной. Использование Ra-01 (мощность до 100 мВт) вместо более мощных модулей полностью оправдано условиями: для испытаний на полигоне этой дальности достаточно, а управление на расстояниях более 200 м без FPV камеры затруднительно.

3.2 Обработка сигналов и формирования пакета

Чтобы команды передавались максимально быстро, данные проходят обработку перед отправкой в эфир:

1. Программная мертвая зона: для исключения самопроизвольного дрейфа дрона из-за неидеального центра джойстиков, в коде задана зона нечувствительности (около 10 единиц АЦП).
2. Масштабирование: АЦП выдает значения от 0 до 1023 (тип int16_t). В микроконтроллере они масштабируются в диапазон от -128 до 127 (тип int8_t) для удобства передачи. В итоге пакет состоит всего из 6 байт (4 оси по 1 байту + 2 тумблера).
3. Перспективы: в будущем планируется переход на побитовую упаковку. 3 оси по 10 бит (газ, крен, тангаж), 1 ось 8 бит (рыскание). Два тумблера – 2 бита, то есть 5 байт, но разрешение стиков управления возрастет в 4 раза (с 256 до 1024 шагов).

3.3 Оптимизация настроек LoRa модуляции

LoRa, в отличие от стандартных модуляций радиосигнала по амплитуде или фазовому сдвигу на фиксированной частоте, передает данные постоянно меняя сигнал по частоте. Диапазон расширения частот определяется настройкой bandwidth. Чем она больше, тем больше данных пройдет за единицу времени, но и тем больше различных шумов попадет в эфир, что снизит дальность. Spreading Factor отвечает за скорость LoRa модуляции, длительность одного символа в эфире. Для коррекции ошибок используется параметр CR, который устанавливает соотношение количества полезных бит данных к передаваемому.

Для реализации передачи команд от пульта частота их передачи должна быть не менее 50 Гц. При объеме одного пакета в 6 байт были подобраны настройки Spreading Factor 7, Bandwidth 500 КГц, CR 4/5 [4]. На этих настройках частота передачи составляет 70 Гц, а дальность около 500 м в условиях прямой видимости.

Для увеличения частоты передачи необходимо либо снижать Spreading Factor, либо же переходить на другую модель приемо-передатчика.

4 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ (ПО КОНТРОЛЛЕРА)

Прошивка главного контроллера написана на языке C++ в среде разработки PlatformIO. Основной цикл работает с жесткой частотой 400 Гц. Эта частота позволяет дрону стабилизироваться, но не упирается в пределы чтения с датчиков.

4.1 Фильтрация данных

Перед запуском алгоритм проводит калибровку. В течение 2 секунд накапливается 2000 значений с гироскопа и акселерометра для вычисления математического смещения (offset). Калибровка должна происходить, когда дрон меньше всего подвержен вибрациям. В рабочем цикле эти смещения вычитаются из сырых данных, задавая идеальный программный ноль.

Для получения углов наклона аппарата используется комплементарный фильтр[1]: $Angle = \alpha * (Angle + \omega * \Delta t) + (1 - \alpha) * Angle_{acc}$

$Angle_{acc}$ вычисляется простой тригонометрией в зависимости от желаемого угла (крен или тангаж). Углы с акселерометра не имеют дрейфа, но не так отзывчивы к резким движениям, как гироскоп. В результате получается отзывчивое значение с крайне низким дрейфом значений. Чувствительность к изменениям настраивается коэффициентом α , в моем случае он равен 0,98. Фрагмент кода показан в Приложении Ж.

Также используются низкочастотные фильтры самого BMI088, которые показали достаточно эффективную работу. Они срезают все шумы выше 145 Гц для акселерометра и выше 116 Гц для гироскопа.

4.2 ПИД Регуляторы

Управление реализовано в виде каскадной системы:

1. Внешний контур (угол): сравнивает требуемый угол с пульты с реальным углом наклона дрона. Выходом этого контура является требуемая угловая скорость.
2. Внутренний контур (угловая скорость): принимает требуемую скорость и сравнивает её с реальной скоростью от гироскопа. Именно он генерирует корректирующее значение для моторов.

Данная реализация позволяет чутко реагировать на мелкие отклонения от желаемого положения, в отличие от одноконтурного регулятора по углу, которому нужно отклониться на большее значение чтобы начать действовать. Также это позволяет ограничить резкость дрона, искусственно ограничив максимальную угловую скорость.

Ось рыскания имеет ПИ-регулятор, Д часть не требуется из-за инертности вращения, что уже подавляет мелкие колебания. Он учитывает лишь на угловую скорость с гироскопа и ведет ее к требуемому значению. Для управления осью рыскания газ противоположных моторов изменяют так, чтобы моторы одной диагонали стали быстрее или медленнее другой

диагонали. В этом случае возникает реактивный момент, который и будет вращать дрон.

4.3 Защита от накопления ошибки

Важным решением стала программная защита от интегрального насыщения. Когда дрон включен и стоит на неровной поверхности он видит ошибку, которую пока не может исправить, так как моторы выключены. Эта ошибка накапливается в интегральной части и при появлении газ вызывает скачок, что вызывает переворот дрона. В коде введена проверка уровня газа. Если дрон стоит на земле (газ убран), интегральная часть жестко обнуляется. Это предотвращает накопление ошибки на земле.

4.4 Матрица микширования

Финальным этапом вычислений является распределение управляющих воздействий ПИД-регуляторов по осям на четыре мотора. Программная логика микшера выглядит так:

- Передний левый = Throttle + Pitch - Roll + Yaw
- Передний правый = Throttle + Pitch + Roll - Yaw
- Задний правый = Throttle - Pitch + Roll + Yaw
- Задний левый = Throttle - Pitch - Roll - Yaw

Полученные значения ограничиваются диапазоном 1000–2000 мкс.

5 ИСПЫТАНИЯ И РЕЗУЛЬТАТЫ

5.1 Тестовый стенд

Для безопасной настройки ПИД-коэффициентов был собран тестовый стенд. Квадрокоптер закрепляется за диагональные лучи на туго натянутой горизонтальной нити. Это позволяет изолировать вращение только по одной оси (например, по тангажу). В таком положении можно искусственно возмущать систему (толкать дрон) и безопасно подбирать коэффициенты P, I и D, анализируя скорость затухания колебаний без риска поломки дрона и получения травм. Настройка показана в Приложении 3

Сперва был настроен внутренний контур до стабильного сопротивления угловой скорости. После чего настраивался внешний контур, стабилизирующий дрон по углу. В результате дрон стабильно стоял на нити не раскачиваясь.

5.2 Результаты летных испытаний

В данный момент проводятся первичные тестовые полеты, требуется дополнительная настройка регуляторов, выставление электроники параллельно раме, переход на другой протокол передачи данных с пульта, описанный выше. Текущий внешний вид дрона показан в Приложении И, взлетная масса составляет 257 г, используется 2S 450 мА*ч 80С батарея.

Идеально стабильного висения на текущем этапе не достигнуто, однако аппарат корректно и без ощутимых задержек реагирует на команды с пульта, пытается отрабатывать изменения угла. Механические демпферы и программные фильтры показали свою эффективность: высокочастотный шум на датчиках удалось минимизировать.

ЗАКЛЮЧЕНИЕ

В ходе выполнения научно-исследовательской работы была успешно достигнута поставленная цель – с нуля разработан, собран и испытан оригинальный полетный контроллер для БПЛА мультироторного типа на базе микроконтроллера STM32. Отказ от использования готовых полетных систем позволил создать полностью прозрачную платформу, научиться работать с данным контроллером.

Основные итоги проделанной работы:

- Реализация модульной архитектуры: спроектирована трехслойная архитектура стека электроники, физически разделившая силовые цепи (ESC), вычислительное ядро (STM32) и модуль связи (ESP32).

Вибрации от рамы передаются на стек лишь через поролон и гибкий шлейф, а не жесткие провода от ESC благодаря модульной структуре

- Оптимизация обмена данными: практические замеры и расчеты позволили определить необходимые частоты для шин данных I2C и UART. Это гарантировало стабильную и быструю передачу, выполнение цикла вычислений на частоте 400 Гц без задержек.
- Создание радиоканала: разработан пульт дистанционного управления и настроена радиосвязь на базе модуляции LoRa. Создан 6 байтовый протокол передачи, обеспечивающий частоту передачи команд до 70 Гц
- Программная реализация: на языке C++ написана для базовых полетов, включающая комплементарный фильтр данных инерциальных датчиков и каскадный ПИД-регулятор с защитой от интегрального насыщения.
- Успешные испытания: проведенные стендовые и первичные летные испытания подтвердили, что разработанный аппаратно-программный комплекс корректно обрабатывает команды и стабилизирует положение дрона в пространстве.

Дальнейшие направления исследований: Разработанный полетный контроллер обладает избыточной базой навигационных приборов (барометр, магнитометр и GPS), что формирует надежный фундамент для дальнейшей модернизации. В перспективе планируется:

- Завершить тонкую настройку регуляторов и реализовать функции удержания высоты с помощью барометра BMP280
- Внедрить алгоритмы автономного полета.
- Оптимизация протокола передачи данных
- Доработать аэродинамику рамы, изменив форму лучей

- Реализовать удобную дистанционную настройку параметров дрона по Wi-Fi через веб-интерфейс
- Исследовать алгоритмы смещения данных с датчиков, в частности фильтра Калмана для удержания дрона на месте без GPS.

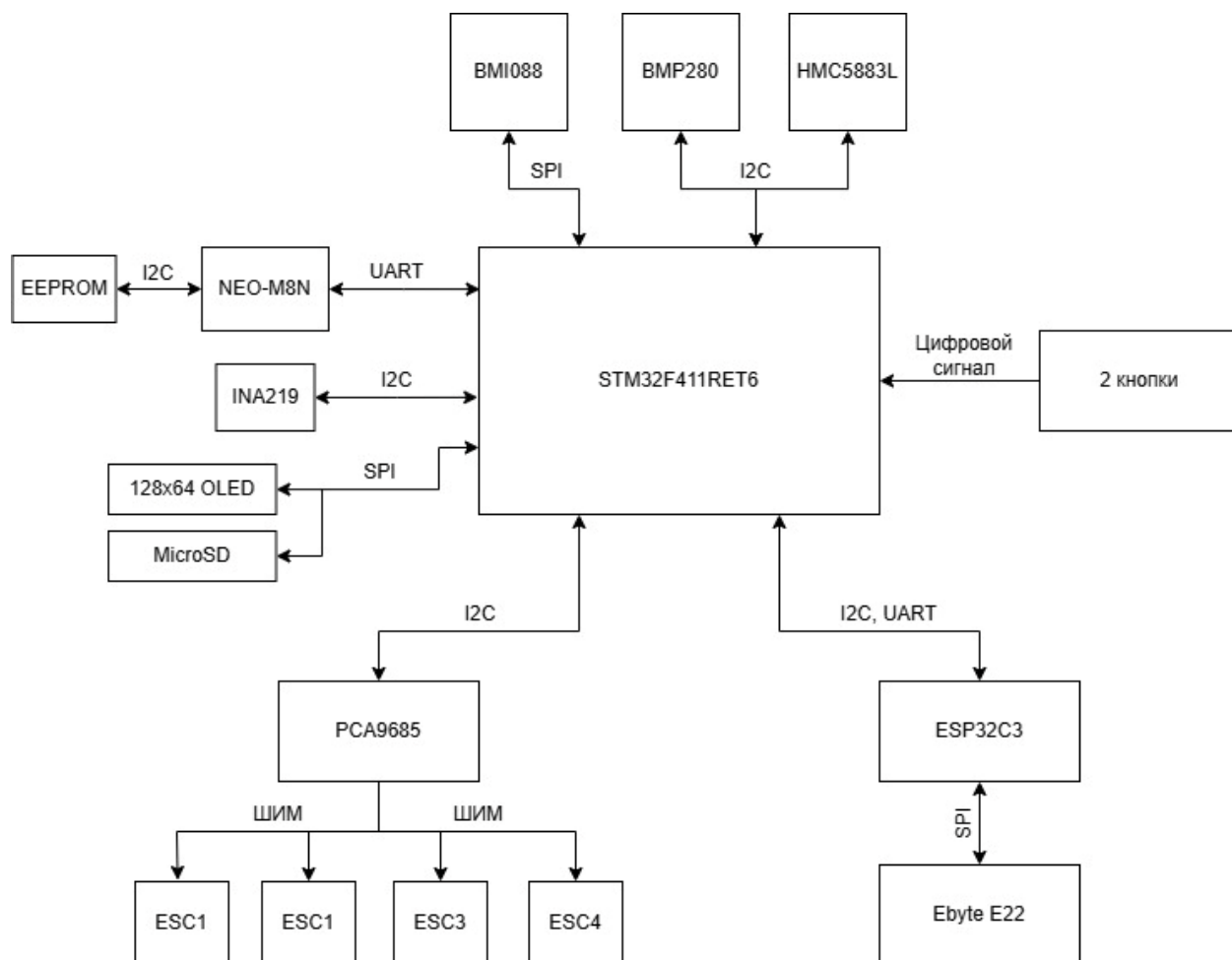
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. [Применение комплементарного фильтра и фильтра Калмана для улучшения стабильности роторных БПЛА – интернет статья](#)
2. [Пособие по первым шагам с STM32 – STM32 Wiki](#)
3. [Документация к STM32F411RET6](#)
4. [Документация к ESP32-C3](#)
5. [Документация к EByte E22](#)
6. [Документация к NEO-M8N](#)
7. [Документация к Bosch BMI088](#)
8. [Документация к NXP PCA9685](#)

ПРИЛОЖЕНИЯ

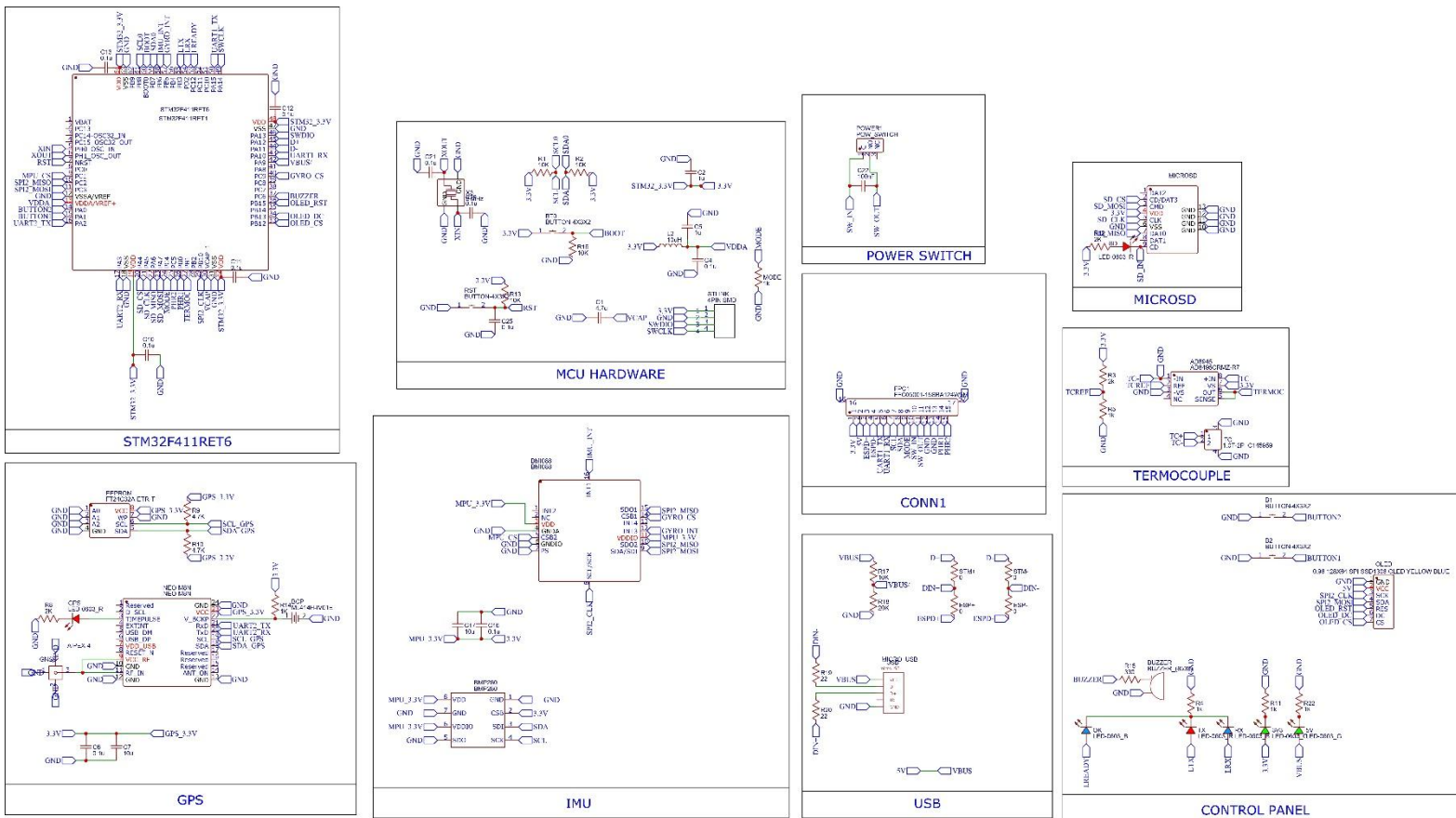
Приложение А

Структурная схема полетного контроллера



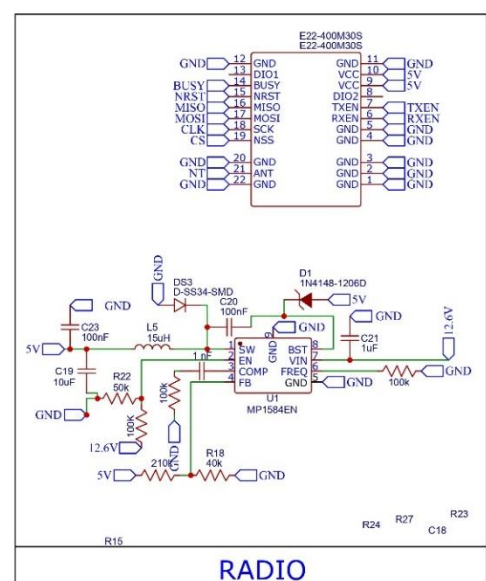
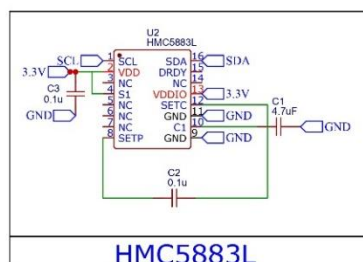
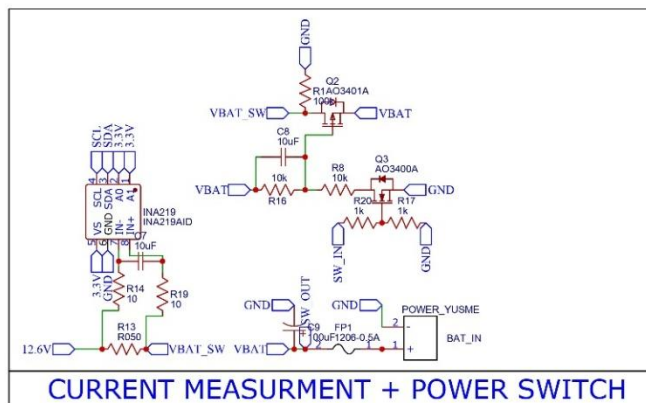
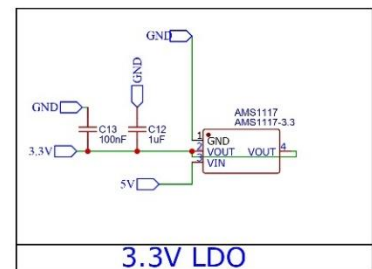
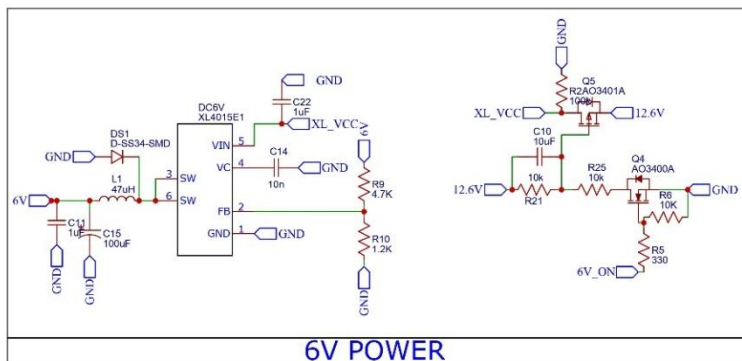
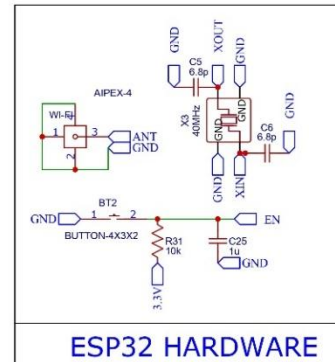
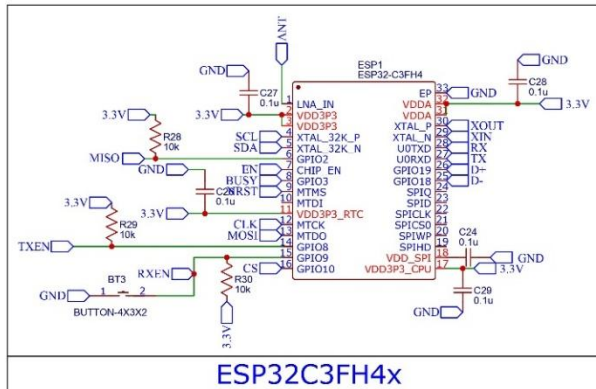
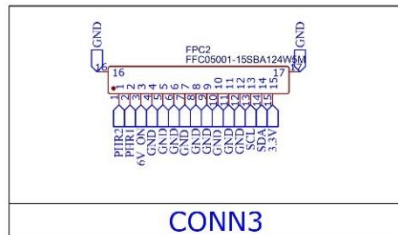
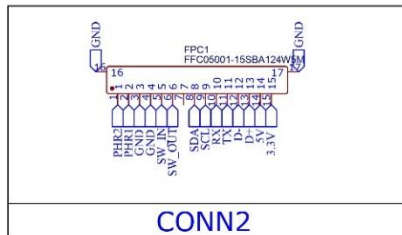
Приложение Б

Принципиальная схема платы с STM32 из EasyEDA



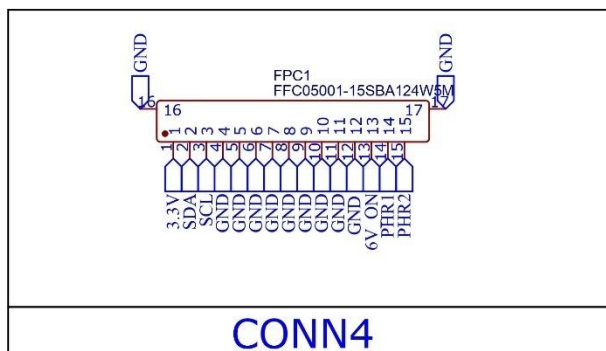
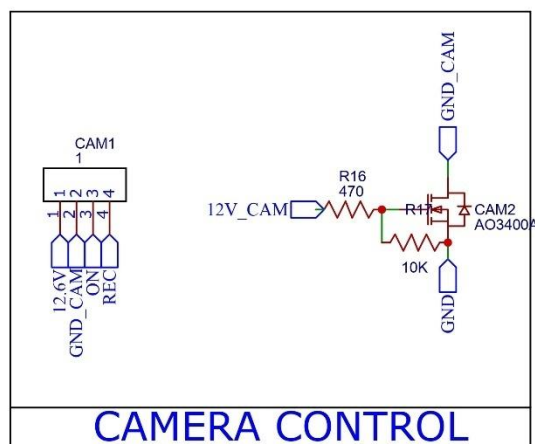
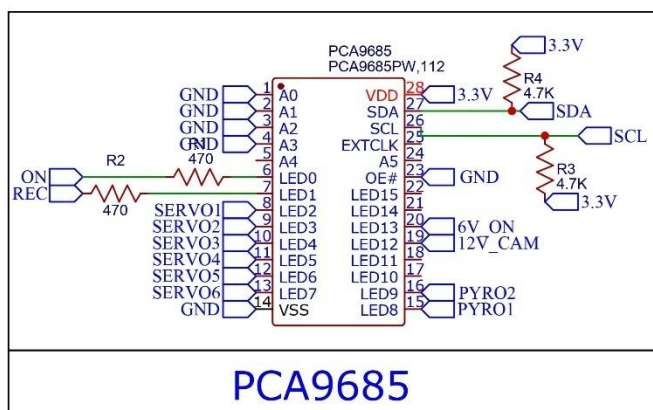
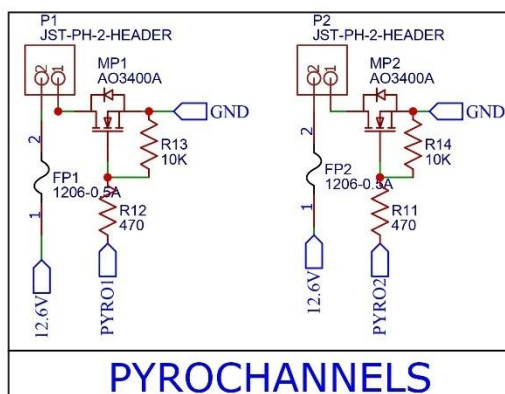
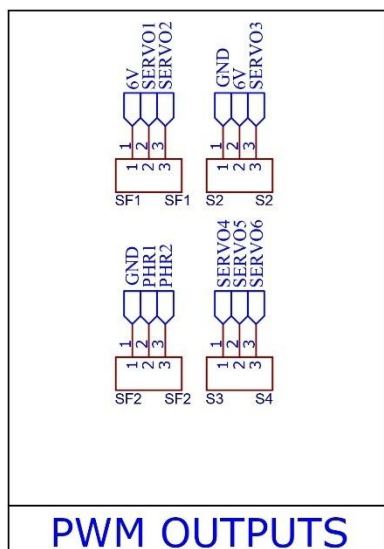
Приложение В

Принципиальная схема платы с ESP32 из EasyEDA



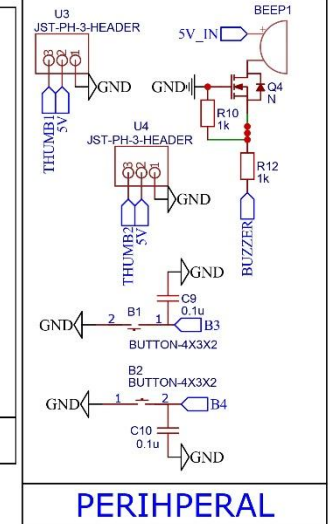
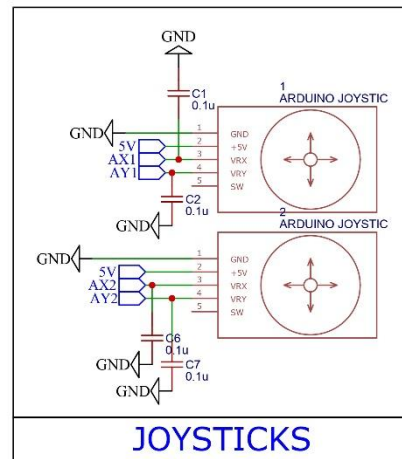
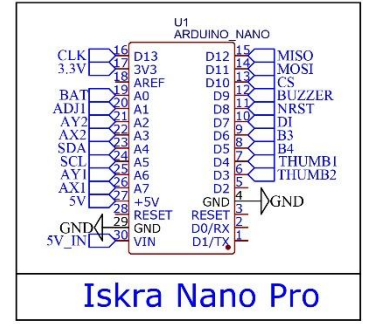
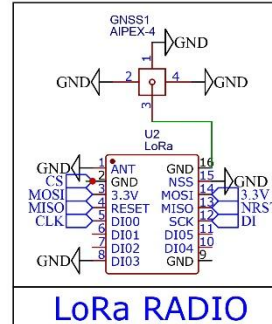
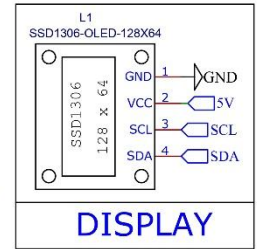
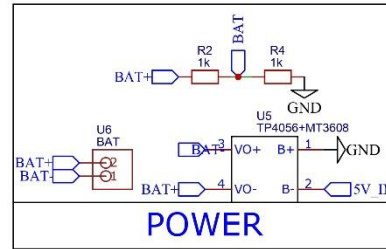
Приложение Г

Принципиальная схема платы с PCA9685



Приложение Е

Внешний вид и принципиальная схема пульта



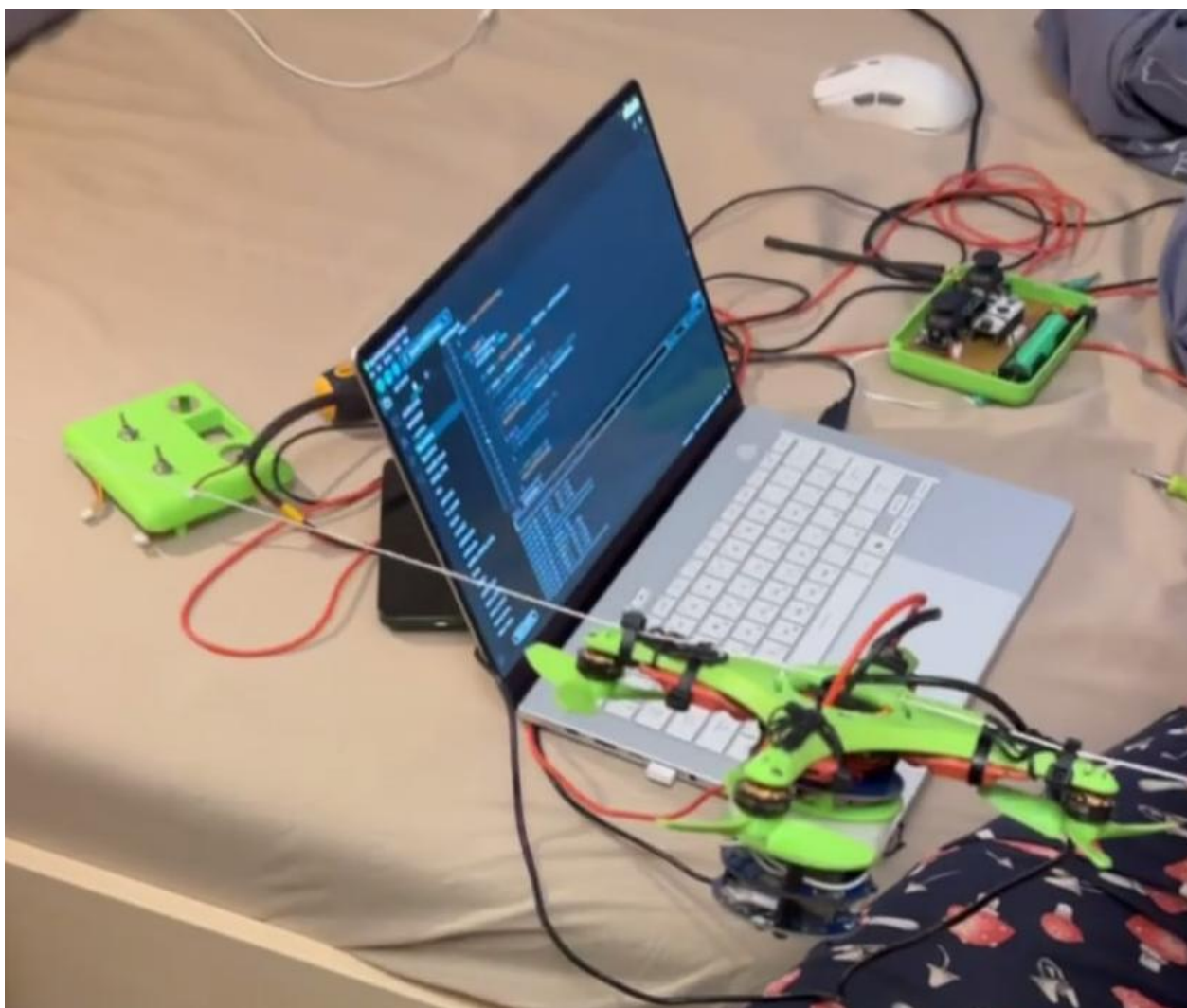
Приложение Ж

Часть кода с получением и фильтрацией углов

```
void loop() {  
  
    accel.readSensor(); gyro.readSensor();  
  
    float gx_raw = (gyro.getGyroX_rads() - gyroX_offset) * 57.2958;  
    float gy_raw = (gyro.getGyroY_rads() - gyroY_offset) * 57.2958;  
    float gz_raw = (gyro.getGyroZ_rads() - gyroZ_offset) * 57.2958;  
  
    static float gx = 0, gy = 0, gz = 0;  
    float filter_coef = 0.3;  
    gx = (gx * (1.0 - filter_coef)) + (gx_raw * filter_coef);  
    gy = (gy * (1.0 - filter_coef)) + (gy_raw * filter_coef);  
    gz = (gz * (1.0 - filter_coef)) + (gz_raw * filter_coef);  
  
    float ax = accel.getAccelX_mss() - accX_offset;  
    float ay = accel.getAccelY_mss() - accY_offset;  
    float az = accel.getAccelZ_mss();  
  
    float acc_roll = atan2(ay, az) * 57.2958;  
    float acc_pitch = atan2(-ax, sqrt(ay * ay + az * az)) * 57.2958;  
    angle_roll = 0.98 * (angle_roll + gx * dt) + 0.02 * acc_roll;  
    angle_pitch = 0.98 * (angle_pitch + gy * dt) + 0.02 * acc_pitch;  
}
```

Приложение 3

Установка для настройки ПИД-регулятора



Приложение И

Фото дрона

