Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

К ЗАЩИТЕ ДОПУСТИТЬ

Зав. каф. ЭВМ

\_\_\_\_\_\_\_\_ Б.В. Никульшин

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломному проекту

на тему

АППАРАТНО-ПРОГРАММНЫЙ КОМПЛЕКС «БЕСПИЛОТНЫЙ ЛЕТАТЕЛЬНЫЙ АППАРАТ С ДИСТАНЦИОННЫМ УПРАВЛЕНИЕМ»

БГУИР ДП 1–40 02 01 01 022 ПЗ

Студент А.В. Деркач

Руководитель Ю.А. Луцик

Консультанты:

от кафедры ЭВМ Ю.А. Луцик

по экономической части В.Г. Горовой

Нормоконтролер Е.Е. Клинцевич

Рецензент

МИНСК 2023

**РЕФЕРАТ**

Дипломный проект предоставлен следующим образом. Электронные носители: 1 компакт-диск. Чертёжный материал: 6 листов формата А1. Пояснительная записка: 136 страниц, 62 рисунка, 32 таблицы, 75 литературных источников, 4 приложения.

Ключевые слова: комплекс, аппаратная система, беспилотный летательный аппарат (БПЛА), квадрокоптер, дрон, устройство управления, ПИД-регулятор, стабилизация, инерционные датчики, микроконтроллер, ATmega, AVR C.

Предметная область находится на пересечении математической модели ПИД-регулятора и проектировании аппаратуры. Объектом разработки является аппаратно-программный комплекс «Беспилотный летательный аппарат с дистанционным управлением».

Целью данного дипломного проекта является разработка универсальной модели БПЛА с возможностью модернизации в зависимости от предполагаемой области использования.

В разработке использовался переносимый язык программирования AVR C и такие инструменты как интегрированная среда разработки для микроконтроллеров Arduino IDE, среда автоматизации проектирования электроники Sprint Layout, система автоматизированного проектирования для промышленного дизайна Autodesk Fusion 360. Использовались такие компоненты, как AVR микроконтроллеры серии ATmega, модуль GY-521 (MPU6050), GY-68 (BMP180), NRF24L01 и другие.

В процессе работы был выполнен анализ существующих аналогов и технической литературы, реализован контроллер полёта летательного аппарата и устройство управления, выполнена попытка разработки математической модели ПИД-регулятора, произведено тестирование разработанного комплекса и уделено внимание вопросам технико-экономического обоснования.

Результаты, полученные в ходе выполненной работы, предполагается использовать в различных областях, включая оборонную промышленность, сельское хозяйство, воздушные доставки, спасательные операции, контроль за транспортом и другие области, где требуется выполнение задач в труднодоступных и опасных условиях. Его пользователями могут быть как частные лица, так и предприятия.

Разработанный комплекс является экономически эффективным, инвестиции в его разработку целесообразны. Не рекомендуется эксплуатировать устройство без наличия опыта в управлении летательными аппаратами.

В дипломном проекте реализованы задачи дипломного проектирования, но необходима доработка программной части для удобства пилотирования. При этом присутствует возможность дальнейшего расширения и улучшения функциональности проекта.

Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет: КСиС. Кафедра: ЭВМ.

Специальность: 40 02 01 «Вычислительные машины, системы и сети».

Специализация: 40 02 01-01 «Проектирование и применение локальных компьютерных сетей».

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

\_\_\_\_\_\_\_\_\_\_ Б.В. Никульшин

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2023 г.

ЗАДАНИЕ

по дипломному проекту студента

Деркач Анжелики Валерьевны

**1** Тема проекта: Аппаратно-программный комплекс «Беспилотный летательный аппарат с дистанционным управлением» – утверждена приказом по университету от 24 марта 2023 г. № 743-с.

**2** Срок сдачи студентом законченного проекта: 1 июня 2023 г.

**3** Исходные данные к проекту:

**3.1** Тип управления устройством: дистанционно, по радиосвязи.

**3.2** Количество двигателей устройства: 4.

**3.3** Обязательные модули устройства: гироскоп, акселерометр, модуль  
радиопередачи.

**3.4** Среда разработки: Arduino IDE, Sprint Layout.

**3.5** Языки программирования: AVR C.

**3.6** Модель динамики полёта летательных аппаратов.

**4** Содержание пояснительной записки (перечень подлежащих разработкевопросов):

Введение. 1. Обзор литературы. 2. Разработка структурной схемы.  
3. Разработка функциональной схемы. 4. Разработка принципиальной схемы. 5. Моделирование. 6. Руководство пользователя. 7. Технико-экономическое обоснование разработки и реализации на рынке аппаратно-программного комплекса «Беспилотный летательный аппарат с дистанционным управлением». Заключение. Список использованных источников. Приложения.

**5** Перечень графического материала (с точным указанием обязательных чертежей):

**5.1** Вводный плакат. Плакат.

**5.2** Аппаратно-программный комплекс «Беспилотный летательный аппарат

с дистанционным управлением». Схема электрическая структурная.

**5.3** Аппаратно-программный комплекс «Беспилотный летательный аппарат

с дистанционным управлением». Схема электрическая функциональная.

**5.4** Аппаратно-программный комплекс «Беспилотный летательный аппарат

с дистанционным управлением». Схема электрическая принципиальная.

**5.5** Аппаратно-программный комплекс «Беспилотный летательный аппарат

с дистанционным управлением». Схема программы.

**5.6** Заключительный плакат. Плакат.

**6** Содержание задания по экономической части: «Технико-экономическое обоснование разработки и реализации на рынке аппаратно-программного комплекса «Беспилотный летательный аппарат с дистанционным управлением».

ЗАДАНИЕ ВЫДАЛ В.Г. Горовой

КАЛЕНДАРНЫЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапов  дипломного проекта | Объем  этапа,  % | Срок выполнения этапа | Примечания |
| Подбор и изучение литературы. Сравнение аналогов. Уточнение задания на ДП | 10 | 23.03 – 30.03 |  |
| Разработка структурной схемы | 15 | 30.03 – 08.04 |  |
| Разработка функциональной схемы | 25 | 08.04 – 19.04 |  |
| Разработка принципиальной схемы | 20 | 19.04 – 08.05 |  |
| Моделирование | 10 | 08.05 – 15.05 |  |
| Расчёт экономической эффективности | 5 | 15.05 – 20.05 |  |
| Оформление пояснительной записки | 15 | 20.05 – 30.05 |  |

Дата выдачи задания: 23.03.2023

Руководитель Ю.А. Луцик

ЗАДАНИЕ ПРИНЯЛ К ИСПОЛНЕНИЮ \_\_\_\_\_\_\_\_\_ А.В. Деркач

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 9](#_Toc134959291)

[1 ОБЗОР ЛИТЕРАТУРЫ 11](#_Toc134959292)

[1.1 Обзор и анализ существующих аналогов 11](#_Toc134959293)

[1.1.1 Мини-дрон MJX BUGS 7 4K V2 11](#_Toc134959294)

[1.1.2 Мини-дрон DJI Mavic Mini 2 13](#_Toc134959295)

[1.1.3 Мини-дрон Syma X56W-P 14](#_Toc134959296)

[1.2 Состав устройства 16](#_Toc134959297)

[1.3 Обзор микроконтроллеров 16](#_Toc134959298)

[1.4 Обзор элементной базы 20](#_Toc134959299)

[1.4.1 Обзор модулей радиопередачи 20](#_Toc134959300)

[1.4.2 Обзор модулей гироскопа и акселерометра 23](#_Toc134959301)

[1.4.3 Обзор барометров 25](#_Toc134959302)

[1.4.4 Обзор модулей GPS 27](#_Toc134959303)

[1.4.5 Обзор двигателей 30](#_Toc134959304)

[1.4.6 Обзор регуляторов оборотов двигателей 32](#_Toc134959305)

[2 РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ 34](#_Toc134959306)

[2.1 Описание основных блоков устройства 34](#_Toc134959307)

[2.2 Блок обработки информации контроллера полёта 34](#_Toc134959308)

[2.3 Блок определения ориентации в пространстве 35](#_Toc134959309)

[2.4 Блок определения высоты 35](#_Toc134959310)

[2.5 Блок приёма и передачи радиосигнала контроллера полёта 36](#_Toc134959311)

[2.6 Блок индикации 36](#_Toc134959312)

[2.7 Блок управления двигателями 37](#_Toc134959313)

[2.8 Блок питания контроллера полёта 37](#_Toc134959314)

[2.9 Блок обработки информации устройства управления 37](#_Toc134959315)

[2.10 Блок управления летательным аппаратом 38](#_Toc134959316)

[2.11 Блок взаимодействия с настройками управления 38](#_Toc134959317)

[2.12 Блок приёма и передачи радиосигнала устройства управления 39](#_Toc134959318)

[2.13 Блок отображения информации 39](#_Toc134959319)

[2.14 Блок питания устройства управления 39](#_Toc134959320)

[3 РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ СХЕМЫ 40](#_Toc134959321)

[3.1 Блоки летательного устройства 40](#_Toc134959322)

[3.1.1 Блок обработки информации контроллера полёта 40](#_Toc134959323)

[3.1.2 Блок определения ориентации в пространстве 42](#_Toc134959324)

[3.1.3 Блок определения высоты 43](#_Toc134959325)

[3.1.4 Блок приёма и передачи радиосигнала контроллера полёта 44](#_Toc134959326)

[3.1.5 Блок индикации 45](#_Toc134959327)

[3.1.6 Блок управления двигателями 46](#_Toc134959328)

[3.1.7 Блок питания контроллера полёта 48](#_Toc134959329)

[3.2 Блоки устройства управления 48](#_Toc134959330)

[3.2.1 Блок обработки информации устройства управления 48](#_Toc134959331)

[3.2.2 Блок управления летательным аппаратом 50](#_Toc134959332)

[3.2.3 Блок взаимодействия с настройками управления 50](#_Toc134959333)

[3.2.4 Блок приёма и передачи радиосигнала устройства управления 50](#_Toc134959334)

[3.2.5 Блок отображения информации 51](#_Toc134959335)

[3.2.6 Блок питания устройства управления 51](#_Toc134959336)

[3.3 Интерфейсы и протоколы 52](#_Toc134959337)

[3.3.1 Интерфейс I2C 52](#_Toc134959338)

[3.3.2 Интерфейс SPI 54](#_Toc134959339)

[3.3.3 Интерфейс UART 55](#_Toc134959340)

[3.3.4 Протокол радиопередачи ESB 56](#_Toc134959341)

[3.3.5 GFSK модуляция 58](#_Toc134959342)

[3.3.6 Широтно-импульсная модуляция 58](#_Toc134959343)

[4 РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ СХЕМЫ 60](#_Toc134959344)

[4.1 Блок обработки информации контроллера полёта 60](#_Toc134959345)

[4.1.1 Подключение питания и внешнего резонатора микроконтроллера 60](#_Toc134959346)

[4.1.2 Подключение порта для программирования микроконтроллера 61](#_Toc134959347)

[4.1.3 Подключение pull-up резисторов 62](#_Toc134959348)

[4.2 Блок определения ориентации в пространстве 62](#_Toc134959349)

[4.2.1 Подключение модуля GY-521 62](#_Toc134959350)

[4.2.2 Программное управление модулем GY-521 63](#_Toc134959351)

[4.3 Блок определения высоты 66](#_Toc134959352)

[4.3.1 Подключение модуля GY-68 66](#_Toc134959353)

[4.3.2 Программное управление модулем GY-68 66](#_Toc134959354)

[4.4 Блок приёма и передачи радиосигнала контроллера полёта 67](#_Toc134959355)

[4.4.1 Подключение модуля NRF24L01 67](#_Toc134959356)

[4.4.2 Программное управление модулем NRF24L01 68](#_Toc134959357)

[4.5 Блок индикации 69](#_Toc134959358)

[4.5.1 Расчёт номинала ограничивающих резисторов 69](#_Toc134959359)

[4.5.2 Программное управление светодиодами 70](#_Toc134959360)

[4.6 Блок управления двигателями 70](#_Toc134959361)

[4.6.1 Подключение регуляторов оборотов и двигателей 71](#_Toc134959362)

[4.6.2 Программное управление двигателями 71](#_Toc134959363)

[4.7 Блок питания контроллера полёта 72](#_Toc134959364)

[4.8 Блок обработки информации устройства управления 73](#_Toc134959365)

[4.8.1 Подключение внешнего EEPROM 73](#_Toc134959366)

[4.8.2 Подключение расширителя портов 74](#_Toc134959367)

[4.8.3 Подключение преобразователя интерфейса USB в UART 75](#_Toc134959368)

[4.8.4 Программное управление внешним EEPROM 77](#_Toc134959369)

[4.8.5 Программное управление расширителем портов 77](#_Toc134959370)

[4.9 Блок управления летательным аппаратом 78](#_Toc134959371)

[4.9.1 Подключение джойстиков 79](#_Toc134959372)

[4.9.2 Программное управление джойстиками 79](#_Toc134959373)

[4.10 Блок взаимодействия с настройками управления 79](#_Toc134959374)

[4.10.1 Подключение кнопок 79](#_Toc134959375)

[4.10.2 Программное управление кнопками 80](#_Toc134959376)

[4.11 Блок приёма и передачи радиосигнала устройства управления 81](#_Toc134959377)

[4.11.1 Подключение модуля NRF24L01 81](#_Toc134959378)

[4.11.2 Программное управления модулем NRF24L01 81](#_Toc134959379)

[4.12 Блок отображения информации 81](#_Toc134959380)

[4.12.1 Подключение дисплея 82](#_Toc134959381)

[4.12.2 Программное управление дисплеем 82](#_Toc134959382)

[4.13 Блок питания устройства управления 84](#_Toc134959383)

[5 МОДЕЛИРОВАНИЕ 85](#_Toc134959384)

[5.1 Тестирование аппаратных средств 85](#_Toc134959385)

[5.1.1 Тестирование блока питания котроллера полёта 85](#_Toc134959386)

[5.1.2 Тестирование блока управления двигателями 85](#_Toc134959387)

[5.1.3 Тестирование блока радиопередачи контроллера полёта и блока радиопередачи устройства управления 86](#_Toc134959388)

[5.1.4 Тестирование блока отображения информации 86](#_Toc134959389)

[5.1.5 Тестирование блока управления летательным аппаратом 87](#_Toc134959390)

[5.1.6 Тестирование блока взаимодействия с настройками управления 87](#_Toc134959391)

[5.1.7 Тестирование блока питания устройства управления 88](#_Toc134959392)

[5.2 Моделирование программного обеспечения 89](#_Toc134959393)

[5.2.1 Алгоритм работы устройства 89](#_Toc134959394)

[5.2.2 Подключение к шине I2C 91](#_Toc134959395)

[5.2.3 Моделирование работы модуля GY-521 92](#_Toc134959396)

[5.2.4 Моделирование изменение скоростей двигателей 93](#_Toc134959397)

[6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 94](#_Toc134959398)

[6.1 Руководство по сборке проекта 94](#_Toc134959399)

[6.1.1 Требования к аппаратному и программному обеспечению 94](#_Toc134959400)

[6.1.2 Настройка окружения 95](#_Toc134959401)

[6.1.3 Загрузка программного обеспечения 96](#_Toc134959402)

[6.1.4 Настройка контроллера полёта 97](#_Toc134959403)

[6.2 Руководство по эксплуатации устройства 101](#_Toc134959404)

[6.2.1 Включение и выключение устройства 102](#_Toc134959405)

[6.2.2 Зарядка устройства 102](#_Toc134959406)

[6.2.3 Индикаторы состояний летательного аппарата 102](#_Toc134959407)

[6.2.4 Управление летательным аппаратом 103](#_Toc134959408)

[6.2.5 Блокировка и разблокировка двигателей 104](#_Toc134959409)

[6.2.6 Отображаемая информация на устройстве управления 105](#_Toc134959410)

[6.2.7 Изменение настроек на устройстве управления 106](#_Toc134959411)

[7 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ И РЕАЛИЗАЦИИ НА РЫНКЕ АППАРАТНО-ПРОГРАММНОГО КОМПЛЕКСА «БЕСПИЛОТНЫЙ ЛЕТАТЕЛЬНЫЙ АППАРАТ С ДИСТАНЦИОННЫМ УПРАВЛЕНИЕМ» 107](#_Toc134959412)

[7.1 Характеристика аппаратно-программного комплекса 107](#_Toc134959413)

[7.2 Расчёт экономического эффекта от производства аппаратно-программного комплекса 107](#_Toc134959414)

[7.3 Расчёт инвестиций в проектирование и производство аппаратно-программного комплекса 112](#_Toc134959415)

[7.3.1 Расчёт инвестиций на разработку аппаратно-программного комплекса 113](#_Toc134959416)

[7.3.2 Расчёт инвестиций в прирост основного капитала 113](#_Toc134959417)

[7.3.3 Расчёт инвестиций в прирост собственного оборотного капитала 113](#_Toc134959418)

[7.4 Расчёт показателей экономической эффективности инвестиций в проектирование и производство аппаратно-программного комплекса 114](#_Toc134959419)

[ЗАКЛЮЧЕНИЕ 116](#_Toc134959420)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 117](#_Toc134959421)

[ПРИЛОЖЕНИЕ А 123](#_Toc134959422)

[ПРИЛОЖЕНИЕ Б 134](#_Toc134959423)

[ПРИЛОЖЕНИЕ В 135](#_Toc134959424)

[ПРИЛОЖЕНИЕ Г 136](#_Toc134959425)

# **ВВЕДЕНИЕ**

В настоящее время беспилотные летательные аппараты (БПЛА) заполнили большую область сфер нашей жизнедеятельности: военные, космические, медицинские, логистические и другие. В научной сфере продолжают вести работу над модернизацией существующих летательных аппаратов (дронов) и разработкой новых моделей для более эффективного выполнения поставленных задач. Темой дипломного проекта является разработка аппаратно-программного комплекса «Беспилотный летательный аппарат с дистанционным управлением».

Стимулом к развитию беспилотной авиации во всем мире послужила потребность в лёгких, относительно дешёвых летательных аппаратах, обладающих высокими характеристиками манёвренности и способных выполнять широкий круг задач. Беспилотные аппараты успешно применяются в ходе военных операций по всему миру, и при этом они так же успешно выполняют задачи гражданского назначения. Основным преимуществом БПЛА является то, что оператор не рискует своей жизнью, недостатком является уязвимость систем дистанционного управления, что особенно важно для устройств военного назначения. Основными сферами применения летательных аппаратов являются: оборонная промышленность, сельское хозяйство, видеосъёмка, воздушные доставки, спасательные операции, научно-исследовательские цели, строительство [2].

В данном дипломном проекте будет рассмотрен БПЛА, представляющий платформу с четырьмя роторами (квадрокоптер). Вращающиеся пропеллеры квадрокоптера создают вертикальную силу тяги. Диагонально расположенные пропеллеры вращаются в разные стороны и обеспечивают компенсацию создаваемых пропеллерами противомоментов. По сравнению с устройствами вертолётного типа, квадрокоптеры обладают рядом преимуществ, таких как: надёжность и простота конструкции, большая стабильность, компактность и манёвренность, малая взлётная масса при существенной массе полезной нагрузки.

Квадрокоптер является сложным устройством, которое трудно классифицировать, так как они имеют очень разные характеристики. Это разнообразие происходит от обилия конфигураций и компонентов БПЛА. Производители пока не ограничены никакими стандартами. В результате сегодня отсутствуют требования со стороны авиационных регуляторов о том, как летательный аппарат должен быть оснащён. При всем многообразии БПЛА условно можно разделить на 4 группы: мини, маленькие, средние и тяжёлые. Компоненты делятся на основные и дополнительные. Основные компоненты включают в себя полётные контроллеры, систему питания, регуляторы и двигатели. Дополнительные компоненты определяются в зависимости от специализации устройства.

Одна из базовых задач управления БПЛА – это обеспечение стабильности полёта. При пилотировании присутствуют внешние воздействия, основные из них это: порывы ветра, резкие перепады давления, турбулентные потоки, радиопомехи, магнитные аномалии. Полётный контроллер – один из наиболее важных компонентов, используемых в БПЛА, который влияет на его полётное поведение. Контроллер обеспечивает стабильность и точность полёта, он может корректировать мощность двигателей, а также изменять скорость или направление движения устройства.

Полётный контроллер представляет собой микроконтроллер, который управляет всей системой БПЛА, а также включает дополнительные датчики для определения положения в пространстве, такие, как: гироскоп, акселерометр, магнитометр, барометр, система GPS и другие. Контроллер также подключается к передатчику радиоуправления, если летательный аппарат управляется оператором. Полётный контроллер работает на основе программного обеспечения, в котором используют различные виды регуляторов [3].

В настоящее время наиболее часто встречающимся регулятором при построении квадрокоптера является пропорционально-интегрально-дифференциальный регулятор (ПИД-регулятор). Причинами столь высокой популярности ПИД-регулятора являются простота построения и промышленного использования, изученность свойств и принципа действия, пригодность для решения большинства практических задач и низкая стоимость. Существует достаточно много методов настройки ПИД-регулятора для управления параметрами устройства, однако, традиционный метод регулирования не может обеспечить приемлемое качество управления при меняющихся характеристиках материалов и среднем качестве используемого оборудования [4]. Чем лучше подобраны параметры регуляторов и выше тактовая частота обработки данных, тем выше точность и стабильность управления.

Целью данного дипломного проекта является разработка и реализация аппаратно-программного комплекса «Беспилотный летательный аппарат с дистанционным управлением» с открытой аппаратной и программной частями. Такая концепция превращает БПЛА в конструктор. В зависимости от своих нужд пользователь может менять параметры управления, полётные функции, добавлять датчики или же изменять программное обеспечение.

В соответствии с поставленной целью были определены следующие задачи:

- разработка структуры будущего аппаратного и программного обеспечения;

- выбор и обоснование элементной базы устройства, учитывая их надёжность, стоимость, функциональность и размеры;

- проектирование и разработка контроллера полёта, представленного в виде платы микроконтроллера и набора датчиков для сбора информации;

- разработка программной части для корректной обработки информации и осуществления связи между элементами;

- тестирование и проведение испытаний разработанной системы с целью проверки правильности сборки и исключения сбоев при эксплуатации устройства.

# **1 ОБЗОР ЛИТЕРАТУРЫ**

## **1.1 Обзор и анализ существующих аналогов**

Одним из основных и начальных этапов разработки устройства является определение перечня необходимых элементов и выбор конкретных моделей. Чтобы сделать правильный выбор, необходимо проанализировать недостатки и преимущества устройств, представленных на рынке.

Далеко не все имеют представление, что такое мини дроны, каким набором функций они обладают и на что способны в техническом плане. Обычно размер этих летательных аппаратов находится в диапазоне от нескольких сантиметров до нескольких десятков сантиметров, что позволяет активно эксплуатировать их в небольших помещениях. Далее будут рассмотрены модели, имеющие схожий функционал с разрабатываем устройством.

### **1.1.1 Мини-дрон MJX BUGS 7 4K V2**

**Радиоуправляемый дрон MJX Bugs 7 с камерой 4K MJX-B7** – это новая модель дронов от компании MJX с мощными бесколлекторными моторами и Ultra HD видеокамерой 4K [5]. Дрон представляет собой небольшой БПЛА для любительских и профессиональных фото и видеосъёмок и может летать на расстоянии до 300 метров от пульта управления. На рисунке 1.1 представлено изображение данного дрона.

*а)**б)*

Рисунок 1.1 – Мини-дрон MJX BUGS 7 4K V2 в двух состояниях:

*а* – рабочее; *б* – сложенное

MJX BUGS 7 4K V2 оснащён профессиональными и **мощными бесколлекторными моторами** с индивидуальными электронными регуляторами скорости. **Автоматическая система блокировки моторов**, установленная на ESC, не только эффективно защищает их от преждевременного выхода из строя, но также обеспечивает безопасность полёта.

Дрон имеет компактный размер и весит всего 249 грамм, что делает его лёгким и манёвренным. Кроме того, он имеет функцию стабилизации камеры, которая позволяет получать более чёткие и плавные видео и фото. Также MJX BUGS 7 4K V2 оснащён двухосевым гироскопом, который обеспечивает стабильный полет дрона и позволяет легко управлять им даже в сложных условиях.

На MJX BUGS 7 4K V2 присутствует система **оптического позиционирования** (Vision Sensor), позволяющая дрону удерживать позицию в дополнение или же в отсутствие сигнала GPS, а также делающей приземление более плавным и точным. GPS позволяет дрону реализовывать такие популярные современные функции, как **«Возврат домой», «Полет по маршруту», «Следуй за мной», «Облёт цели».**

Летает дрон на аккумуляторе Li-Po 2S 1500 mAh, что даёт ему примерно **15 минут полётного времени**.

Основные характеристики данного устройства представлены в таблице 1.1.

Таблица 1.1 – Характеристики дрона MJX BUGS 7 4K V2

|  |  |
| --- | --- |
| Характеристики | MJX BUGS 7 4K V2 |
| Максимальная высота взлёта | 120 м |
| Максимальное время полёта | 15 мин |
| Максимальная скорость полёта | 32 км/ч |
| Максимальная скорость подъёма | 3 м/с |
| Максимальная скорость снижения | 2 м/с |
| Размер рамы | 175 мм |
| Размеры дрона | в сложенном виде – 140 x 75 x 55 мм,  в разложенном виде – 240 х 235 х 55 мм |
| Вес | 245 г |
| Рабочая температура | от 0 °С до 40 °C |
| Двигатели | бесколлекторные, 1306 2750KV |
| Встроенные датчики | GPS, оптический датчик, барометр |
| Стоимость | 180$ |

К недостаткам MJX BUGS 7 4K V2 можно отнести:

1. Время работы батареи. Автономность данной модели составляет всего 15 минут.

2. При низком заряде дрон перестаёт корректно выполнять команды, полученные с пульта управления.

3. Плохо удерживает позицию при наличии сильного ветра.

4. Иногда некорректно отрабатывает команда «Вернуться домой», когда пропадает связь с пультом управления.

5. Качество сборки оставляет желать лучшего.

6. Отсутствует управление поворотом камеры.

7. Долгая зарядка батареи.

### **1.1.2 Мини-дрон DJI Mavic Mini 2**

DJI Mavic Mini 2 – это любительская модель ультракомпактного класса, созданная для фото и видеосъёмки от компании DJI [6]. Установленная на крохотном, но очень эффективном стабилизирующем подвесе камера может снимать качественное 4K-видео и 12-мегапиксельные фото. Данная модель является улучшенной версией DJI Mavic Mini и предлагает более высокое качество видео и фотографий.

На рисунке 1.2 представлено изображение данного дрона в двух состояниях.

*а)**б)*

Рисунок 1.2 – Мини-дрон DJI Mavic Mini 2 в двух состояниях:

*а* – рабочее; *б* – сложенное

При своём небольшом весе **DJI Mini 2** предлагает удивительную производительность. Новейшая технология OcuSync 2.0 обеспечивает стабильную и бесперебойную передачу данных между пультом дистанционного управления и дроном. Дальность передачи сигнала составляет до 10 км. Встроенные двигатели позволяют находиться в воздухе до 31 минуты, разгоняться до 57.6 км/ч, причём с моментальным ускорением, а еще дрон выдерживает порывы ветра до 10 метров в секунду.

Управляется **DJI Mini 2** с помощью пульта управления и приложения DJI Fly, доступное для скачивания на устройства на базе iOS и Android.

Встроенный датчик GPS обеспечивает связь с пультом через мобильное приложение, где в режиме реального времени можно отслеживать нахождение дрона, даже если он удалился на максимальное расстояние. В нижней части корпуса присутствует два датчика, которые отвечают за безопасное приземление устройства. При снижении они позволяют дрону определить расстояние до земли и анализируют рельеф поверхности для обеспечения более плавной и мягкой посадки. Ещё одна технология для безопасного полёта, отвечающая за своевременное возвращение на место взлета, так и называется – «Возврат домой».

Основные характеристики данного устройства представлены в таблице 1.2.

Таблица 1.2 – Характеристики дрона DJI Mavic Mini 2

|  |  |
| --- | --- |
| Характеристики | DJI Mavic Mini 2 |
| Максимальная высота взлёта | 2000 м |
| Максимальное время полёта | 31 мин |
| Максимальная скорость полёта | 16 м/с |
| Максимальная скорость подъёма | 5 м/с |
| Максимальная скорость снижения | 3.5 м/с |
| Рабочая частота | 2.4 – 2.4835 ГГц |
| Размер рамы | 213 мм |
| Размеры квадрокоптера | в сложенном виде – 138 x 81 x 58 мм,  в разложенном виде – 245 х 289 х 56 мм |
| Вес | 249 г |
| Рабочая температура | от 0 °С до 40 °C |
| Двигатели | бесколлекторные, 1408 3000KV |
| Стоимость | 940$ |

К недостаткам DJI Mavic Mini 2 можно отнести:

1. Время работы батареи. Автономность данной модели составляет всего 31 минуту, чего недостаточно для устройства высокой ценовой категории.

2. Высокая стоимость. Данная модель находится в одной из высших ценовых категорий данного класса дронов, из-за чего не все могут позволить его к покупке.

3. Нет функции определения препятствий, из-за чего необходимо самостоятельно отслеживать положение дрона.

4. Разъёмы открыты, что может привести к попаданию туда воды и грязи.

5. Отсутствие функции следования за объектом, которая присутствует в других устройствах более бюджетного класса.

### **1.1.3 Мини-дрон Syma X56W-P**

Одним из самый простых и миниатюрных БПЛА является мини-дрон Syma X56W-P (или X56W PRO) – доработанная модификация популярного Suma X56W [7].

Изображение данного дрона представлено на рисунке 1.3.



Рисунок 1.3 – Мини-дрон Syma X56W-P

На Syma X56W-P установлены моторы коллекторного типа с передачей крутящего момента через редуктор закрытого типа.

Дрон оснащён 1 Мп HD камерой, установленной на корпусе без подвеса. Полет от первого лица для этой модели реализован посредством Wi-Fi на частоте 2.4 ГГц. Вывод картинки в режиме реального времени осуществляется через мобильное приложение «Syma Fly» (для IOS/Android). Для комфортного полёта требуется свободная от помех зона, и расстояние удаления от спряжённого смартфона не далее 10 – 15 метров. В противном случае отрицательное влияние задержки сводит на нет дальнейшее пилотирование. Разрешение видеопотока 720 p.

Главная отличительная особенность представленной модели – это наличие оптического сенсора, благодаря которому обеспечивается позиционирование дрона. Работает датчик в паре с барометром, посредством которого реализована функция удержания высоты. Благодаря функции «План полёта» Syma X56W-P может летать по нарисованному маршруту.

Основные характеристики данного устройства представлены в таблице 1.3.

Таблица 1.3 – Характеристики квадрокоптера Syma X56W-P

|  |  |  |
| --- | --- | --- |
| Характеристики | | Syma LH-X56 |
| 1 | | 2 |
| Максимальная высота взлёта | | 30 м |
| Максимальное время полёта | | 15 мин |
| Максимальная скорость полёта | | 4 м/с |
| Максимальная скорость подъёма | | 1 м/с |
| Максимальная скорость снижения | | 1 м/с |
| Рабочая частота | | 2.4 ГГц |
| Радиус соединения | 80 м | |
| Размеры квадрокоптера | 355 x 355 x 50 мм | |

Продолжение таблицы 1.3

|  |  |
| --- | --- |
| 1 | 2 |
| Вес | 157 г |
| Рабочая температура | от 0 °С до 40 °C |
| Двигатели | коллекторные |
| Стоимость | 70$ |

Так как устройство бюджетное и не предназначено для выполнения особых задач, то ему присуще все недостатки бюджетных устройств. Основные из них: малое время работы, дешёвые материалы, низкая скорость и радиус связи, отсутствие популярных функций.

## **1.2 Состав устройства**

Как сказано ранее, разрабатываемое микропроцессорное устройство представляет собой мини БПЛА, который выполняет функции полёта и управляется посредством радиосигнала, а также информирует пользователя об измеряемых параметрах. Для решения этих задач в состав устройства должны входить:

- микроконтроллер;

- модуль радиопередачи;

- гироскоп;

- акселерометр;

- барометр;

- двигатели;

- регуляторы оборотов двигателей;

- аккумуляторы;

- другие устройства контроля и управления.

## **1.3 Обзор микроконтроллеров**

Управление полётом и всеми датчиками дрона осуществляется при помощи контроллера полёта, основным элементом которого является микроконтроллер. Микроконтроллер – это микроэлектронное программируемое устройство, предназначенное для получения информации и передачи этой информации тем устройствам, которые её обработают, а также управления этими устройствами и выполнения простейших арифметических операций. Микроконтроллеры выпускают десятки компаний, причём производятся не только современные 32-битные микроконтроллеры, но и 16, и даже 8-битные (как i8051 и аналоги). Внутри каждого семейства часто можно встретить почти одинаковые модели, различающиеся скоростью работы центрального процессора (ЦПУ) и объёмом памяти.

Для сравнения был выбран микроконтроллер ATmega328P-AU и аналоги других производителей. Результаты сравнения приведены в таблице 1.4.

Таблица 1.4 – Сравнение микроконтроллеров

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметры сравнения** | **MEGA328P-AU** | **STM32F103** | **ESP32-DOWDQ6** |
| Архитектура | AVR | ARM Cortex-M3 | Xtensa Dual-Core LX6 |
| Количество входов/выходов | 23 | до 80 | до 34 |
| Тактовая частота | 20 МГц | до 72 МГц | до 240 МГц |
| Flash-память | 32 Кб | до 512 Кб | до 4 Мб |
| SRAM-память | 2 Кб | до 64 Кб | до 520 Кб |
| Количество ядер | 1 | 1 | 2 |
| Разрядность | 8 бит | 32 бит | 32 бит |
| ШИМ-каналы | 6 | до 15 | до 16 |
| Интерфейсы | SPI, I2C, UART | SPI, I2C, UART, CAN, USB, Ethernet | SPI, I2C, UART, CAN, Ethernet,  Wi-Fi, Bluetooth |
| Рабочая температура | от -40 до +85 ºС | от -40 до +85 ºС | от -40 до +85 ºС |
| Размеры | 7 мм × 7 мм | 10 мм × 10 мм | 6 мм × 6 мм |

**Микроконтроллер** ATmega328P-AU– это компактная модель 8-битного микроконтроллера, которая создана на базе picoPower AVR RISC от компании ATMEL. Данный микроконтроллер имеет объем FLASH-памяти в 32 Кб, объем EEPROM 1 Кб и RAM 2 Кб. Он работает на тактовой частоте до 20 МГц, что позволяет выполнять сложные вычисления и обработку данных.

Несмотря на малые размеры, в устройстве достаточно обширный интерфейс и 23 универсальных порта ввода-вывода. ATmega328P-AU поддерживает различные протоколы связи, включая UART, SPI и I2C, что обеспечивает универсальность в подключении к другим устройствам и системам. Питание микроконтроллера происходит от источника в 1.8 – 5.5 В, а широкий диапазон рабочих температур от -40 до +85 °C в сочетании с малыми размерами и низкой стоимостью, даёт широкие возможности для использования в самых различных роботизированных устройствах.

Микроконтроллер ATmega328P-AU может быть запрограммирован и отлажен через стандартный интерфейс программирования ISP (In-System Programming), а также имеет Bootloader, что делает его удобным для разработки и тестирования проектов.

Микроконтроллеры ATMEL AVR применяются в классической линейке устройств Arduino, поэтому к достоинствам можно отнести низкий порог вхождения – удобная кроссплатформенная интегрированная среда разработки, используемые в библиотеках высокоуровневые языки программирования. Все это существенно облегчает работу с микроконтроллером и делает его удобными как для новичков, так и для опытных разработчиков, что является наиболее существенным достоинством, так как концепция открытой аппаратной и программной частей предполагает, что пользователь без специализированных навыков должен справиться с настройкой и усовершенствованием устройства.

Изображение микроконтроллера представлено на рисунке 1.4.

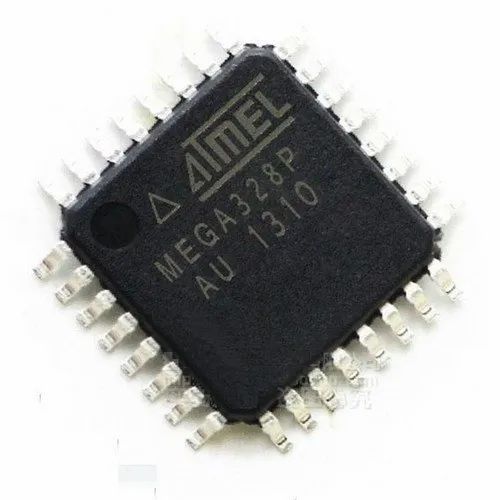


Рисунок 1.4 – Микроконтроллер ATmega328P-AU

Микроконтроллер семейства STM32F10x: серии STM32F103 является 32-битными микроконтроллером, разработанным компанией STMicroelectronics на основе ядра ARM Cortex-M3 и работает на тактовой частоте до 72 МГц, что даёт ему одну из лучших в классе 32-битных микроконтроллеров производительность.

STM32F103 имеет встроенную FLASH-память до 512 Кб и SRAM-память объёмом до 64 Кб, а также предоставляет широкие возможности для периферийных устройств так как поддерживает много интерфейсов: USB, CAN, I2C, SPI, UART, ADC и другие.

Помимо высокой производительности, микроконтроллер отличается низким энергопотреблением. Потребление энергии зависит от режима работы микроконтроллера. В активном режиме STM32F103 потребляет от 6 мА до 50 мА при частоте 72 МГц и выполнении программ из FLASH-памяти. Для управления потреблением энергии в STM32F103 предусмотрены различные механизмы, такие как механизмы снижения частоты ядра, динамическое управление напряжением ядра и автоматическое переключение режимов питания. В целом, микроконтроллер идеально подходит для проектов с батарейным питанием и может быть использован с ограниченным бюджетом энергопотребления.

К главным минусам микроконтроллера можно отнести:

1. Высокая сложность. Использование STM32F103 может быть сложным для начинающих разработчиков, так как он имеет множество интерфейсов и многоуровневую архитектуру.

2. Высокая стоимость. Хотя STM32F103 имеет множество преимуществ, его стоимость выше, чем у других производителей.

Изображение микроконтроллера STM32F103 представлено на рисунке 1.5.

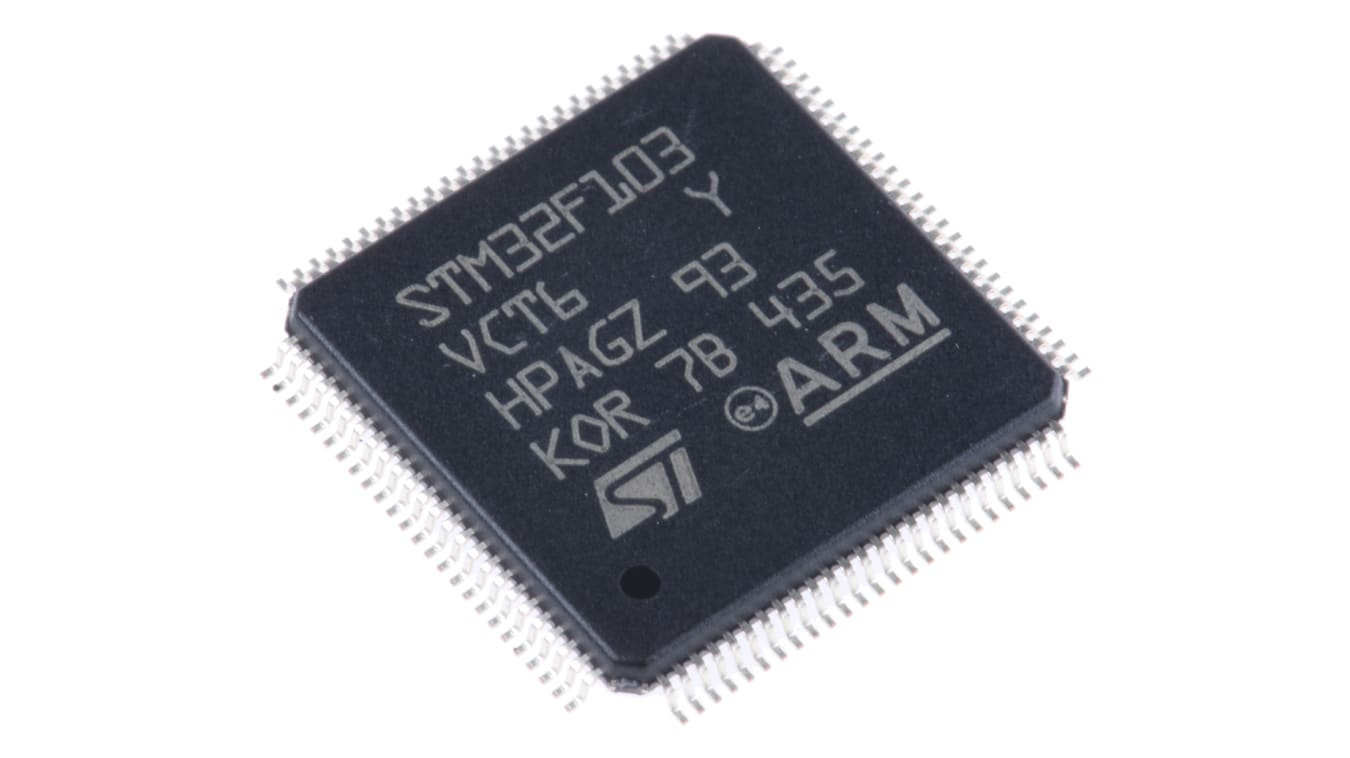


Рисунок 1.5 – Микроконтроллер STM32F103

ESP32-DOWDQ6 - это микроконтроллер, разработанный компанией Espressif Systems на основе архитектуры Xtensa LX6. Он представляет собой двухъядерный процессор с тактовой частотой до 240 МГц, который обеспечивает высокую производительность и низкое энергопотребление.

Микроконтроллер имеет встроенный Wi-Fi и Bluetooth, что делает его идеальным для различных приложений, связанных с Интернетом вещей (IoT), таких как умный дом, автоматизация промышленных процессов и т.д. Кроме того, устройство имеет встроенную память Flash объёмом до 4 Мб, а также 520 Кб RAM.

ESP32-DOWDQ6 имеет множество периферийных интерфейсов, таких как SPI, I2C, UART, ADC, что позволяет подключать к нему различные датчики и устройства. Устройство также имеет встроенный USB-контроллер, который обеспечивает поддержку USB-устройств и USB-хоста.

В целом, ESP32-DOWDQ6 является мощным и гибким микроконтроллером, который обеспечивает высокую производительность и множество функций для различных приложений, связанных с IoT.

К главным недостаткам микроконтроллера можно отнести:

1. Низкий уровень энергоэффективности. Микроконтроллер ESP32-DOWDQ6 потребляет довольно много энергии, что может ограничивать его использование в приложениях, где важна низкая энергопотребляемость.

2. Высокая стоимость. ESP32-DOWDQ6 является достаточно дорогим микроконтроллером, что увеличивает стоимость проектов, в которых он используется.

3. Сложность настройки. Настройка микроконтроллера ESP32-DOWDQ6 может быть сложной и требует определённых навыков, что может оказаться проблемой для начинающих пользователей.

На рисунке 1.6 представлено изображение микроконтроллера ESP32-DOWDQ6.



Рисунок 1.6 – Микроконтроллер ESP32-DOWDQ6

Для получения более подробной информации о рассмотренных микроконтроллерах использовалась техническая спецификация [9, 10, 11, 13] и источники [8, 12].

После рассмотрения характеристик микроконтроллеров видно, что ESP32-DOWDQ6 имеет наилучшие параметры при маленьких размерах. Микроконтроллер является двухъядерным, с разрядностью 32 бит, и работает на тактовой частоте до 240 МГц, чем превосходит своих аналогов. Также микроконтроллер имеет хороший запас ОЗУ и флэш-памяти из-за чего подходит для реализации более сложных проектов.

Модель **MEGA328P-AU является самой простой, из рассмотренных, и подходит для реализации несложных проектов, а также является наиболее подходящей для начинающих пользователей. Микроконтроллер STM32F103 представляет собой нечто среднее между двумя другими.**

## **1.4 Обзор элементной базы**

### **1.4.1 Обзор модулей радиопередачи**

Для осуществления связи между летательным устройством и пультом управления, используются модули радиопередачи. Передача радиосигнала очень важна для контроля и управления дроном, а также для получения полезной информации в режиме реального времени. Существуют модули с однонаправленной и двунаправленной передачей радиосигнала.

Для сравнения были выбраны радиомодули NRF24L01 и HC-12 с двунаправленной передачей данных, что позволит обмениваться данными, как от устройства управления к дрону, так и передавать информацию, полученную с дрона, на устройство управления.

Ниже, в таблице 1.5, приведены сравнительные характеристики рассматриваемых радиомодулей.

Таблица 1.5 – Сравнение модулей радиопередачи

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметры сравнения** | **NRF24L01** | **HC-12** | |
| Производитель | Nordic Semiconductor | HopeRF | |
| Частота радиосигнала | 2.4 ГГц | 433 МГц, 868 МГц | |
| Напряжение питания | 1.9 – 3.6 В | 3.2 – 5.5 В | |
| Потребляемый ток | до 14 мА | до 100 мА | |
| Рабочая температура | от -40 до +85 ºС | от -40 до +85 ºС | |
| Скорость  приёма/передачи  данных | до 2 Мбит/с | до 9.6 Кбит/с | |
| Расстояние  приёма/передачи | до 100 м без внешней антенны / до 1 км с внешней антенной | до 1 км |
| Интерфейс | SPI | UART |
| Поддержка режима энергосбережения | Да | Да |
| Поддержка множества устройств | Да | Да |
| Количество радиоканалов | 126 | 100 |
| Тип модуляции | GFSK | FSK |
| Размеры | 29 мм × 15 мм | 27.8 мм × 14.4 мм |

Одни из самых популярных микросхем приёмопередатчиков – это NRF24L01. Высокую популярность они получили за счёт сочетания низкой цены, относительно небольшого энергопотребления, простоты работы с ними и высокой гибкостью при построении сетей различных топологий и сложности. NRF24L01 – это высокоинтегрированная микросхема с пониженным потреблением энергии и скоростью передачи до 2 Мбит/с для диапазона 2.4 ГГц, которая подключается к микроконтроллеру с помощью SPI-интерфейса и поддерживает 126 радиоканалов. При помощи модуля можно объединить до семи приборов в одну общую радиосеть. В этом случае один из модулей будет выступать в роли ведущего, остальные – ведомые.

Модуль NRF24L01 (Gaussian Frequency Shift Keying) использует GFSK модуляцию – это тип цифровой модуляции, который использует гауссово сглаживание (или фильтрацию) на смещение частоты для передачи цифровых данных. Фильтрация сигнала передатчика GFSK приводит к сглаживанию спектра, что означает, что его спектральная плотность находится в пределах более узкой полосы частот, чем при использовании других форм модуляции. Это позволяет более эффективно использовать радиочастотный канал и обеспечивает более надёжную передачу данных с меньшей вероятностью ошибок при низкой скорости передачи.

Модули бывают разных габаритов, с печатной антенной, керамической или внешней, с усилителем или без. На плате может быть стабилизатор питания на 3.3 В, может и не быть. На рисунке 1.7 приведены наиболее популярные модели радиомодулей NRF24L01.

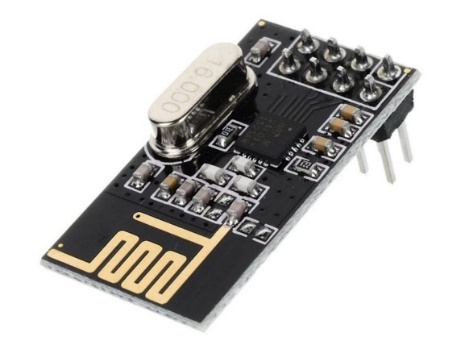
*а)**б)*

Рисунок 1.7 – Радиомодуль NRF24L01: *а* – без внешней антенны,

*б* – с внешней SMA антенной

HC-12 – приёмопередающий модуль связи на базе чипа SI4482, который может использоваться для беспроводной связи между устройствами на расстоянии до 1 км. Модуль использует частоту 433 или 868 МГц и имеет максимальную скорость передачи данных до 9.6 Кбит/с. HC-12 поддерживает несколько режимов работы, включая режим сна, режим передачи данных и режим настройки.

Модуль имеет 100 радиоканалов, которые могут быть использованы для обеспечения безопасности передачи данных и снижения влияния помех. Также в модуле есть встроенная антенна, но для повышения качества связи можно использовать внешнюю антенну.

HC-12 работает с помощью простого UART интерфейса, что делает его легко интегрируемым с микроконтроллерами и другими устройствами. Все настройки модуля могут быть изменены с помощью AT-команд, что упрощает процесс настройки и интеграции. В модуле используется модуляция частотной/частотно-сдвиговой ключевой (FSK) – это метод модуляции, при котором информационный сигнал изменяет частоту несущей в зависимости от состояния модулирующего сигнала.



Рисунок 1.8 – Радиомодуль HC-12

Для получения информации о данных модулях использовалась техническая спецификация [14, 16] и источники [15, 17].

Как видно из сравнения характеристик и особенностей работы радиомодулей, модель NRF24L01 превосходит радиомодуль HC-12 по таким важным параметрам, как скорость передачи, количество радиоканалов и рабочая частота. Использование NRF24L01 также упростит работу пользователя с радиомодулем из-за его популярности и простоты использования.

### **1.4.2 Обзор модулей гироскопа и акселерометра**

Гироскоп и акселерометр – это инерционные датчики, которые используются в БПЛА для измерения угловых скоростей и ускорений соответственно. Гироскоп позволяет определить угловые скорости вокруг его осей, акселерометр позволяет измерять линейные ускорения устройства, что важно для управления дроном и поддержания его ориентации и стабилизации в пространстве.

Для сравнения были выбраны наиболее популярные модули. Результаты сравнения представлены в таблице 1.6.

Таблица 1.6 – Сравнение модулей с гироскопом и акселерометром

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметры сравнения** | **GY-521** | **GY-91** | **GY-9255** |
| Напряжение питания | 3.3 – 5 В | 3.3 – 5 В | 3.3 – 5 В |
| Потребляемый ток | до 4 мА | до 10 мА | до 6 мА |
| Диапазон измерений акселерометра | ± 2 ± 4 ± 8 ± 16 g | ± 2 ± 4 ± 8 ± 16 g | ± 2 ± 4 ± 8 ± 16 g |
| Диапазон измерений гироскопа | ± 250 ±500  ±1000 ±2000 °/с | ± 250 ±500  ±1000 ±2000 ° / с | ± 250 ±500  ±1000 ±2000 ° / с |
| Интерфейсы | I2C, SPI | I2C, SPI | I2C, SPI |
| Рабочая температура | от -40 до +85 ºС | от -40 до +85 ºС | от -40 до +85 ºС |
| Размеры | 13 мм × 20 мм | 21 мм × 15 мм | 15 мм × 25 мм |

Модуль GY-521 – это инерционный модуль, который используется для измерения ускорения, угловой скорости и магнитного поля в трех измерениях. Он основан на чипе MPU-6050, который включает в себя трёхосевой гироскоп, трёхосевой акселерометр и дополнительно имеет магнитометр HMC5883L, что позволяет определять ориентацию объекта в пространстве и его движение.

Микросхема MPU6050 имеет 16-битный аналого-цифровой преобразователь (АЦП), который обеспечивает точность измерений. Модуль имеет встроенный фильтр Калмана низких частот для уменьшения шума и стабилизации данных.

Обмен данными между модулем GY-521 и микроконтроллером осуществляется по шине I2C. Необходимое напряжение питания для модуля MPU6050 регулирует встроенный стабилизатор напряжения 3.3V.

На рисунке 1.9 представлено графическое изображение модуля.

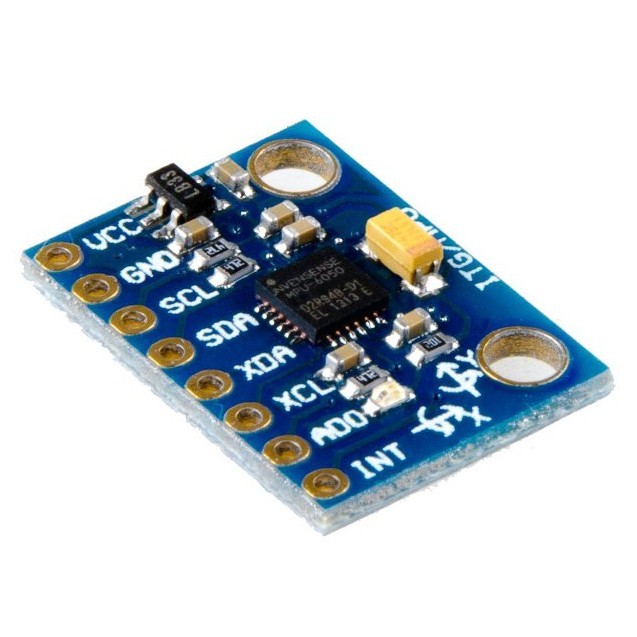


Рисунок 1.9 – Модуль гироскопа и акселерометра GY-521 (MPU6050)

Модуль GY-91 – это небольшой электронный модуль, предназначенный для измерения параметров окружающей среды, состоящий из трёх основных компонентов: MPU-9250, BMP280 и L883. Он объединяет в себе несколько датчиков: трёхосевой акселерометр, трёхосевой гироскоп, магнитометр и барометр. Акселерометр измеряет ускорение по трем осям и позволяет определить направление силы тяжести. Гироскоп измеряет угловую скорость вращения вокруг трех осей и позволяет определить изменение ориентации объекта. Магнитометр измеряет магнитное поле и позволяет определить ориентацию в пространстве относительно магнитного поля Земли. Барометр измеряет атмосферное давление и позволяет определить высоту над уровнем моря. Использование модуля GY-91 позволяет сэкономить место в готовом устройстве, а также его конечную стоимость, если устройство предполагает использование акселерометра, гироскопа и барометра.

Модуль GY-91 подключается к микроконтроллеру или компьютеру через интерфейс I2C или SPI и может быть использован с различными платформами, такими как Arduino, Raspberry Pi и другие.

Изображение модуля GY-91 представлено на рисунке 1.10.

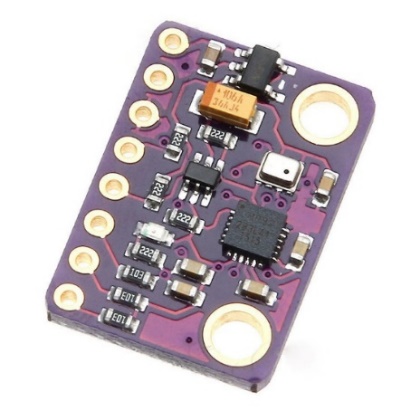


Рисунок 1.10 – Модуль гироскопа, акселерометра и барометра

GY-91 (MPU9250 + BMP280)

Модуль GY9255 основан на микросхеме MPU-9255, которая представляет собой мульти-чип, состоящий из двух чипов, интегрированных в общий корпус. Первая часть модуля состоит из трёхосевого гироскопа и трёхосевого акселерометра. Вторая часть модуля, собранная на чипе AK8963, представляет собой магнитометр. Модуль не требует сложной калибровки с использованием сложных и дорогостоящих приборов. MPU-9255 может использоваться для сопряжения с несколькими цифровыми датчиками (например, датчики давления и другие).

GY9255 оснащен интерфейсами I2C и SPI, которые позволяют передавать 16-битные данные с АЦП. Модуль включает в себя поддержку для автоматического распознавания активности и обеспечивает высокую точность при низком энергопотреблении.

Изображение модуля GY-9255 представлено на рисунке 1.11.

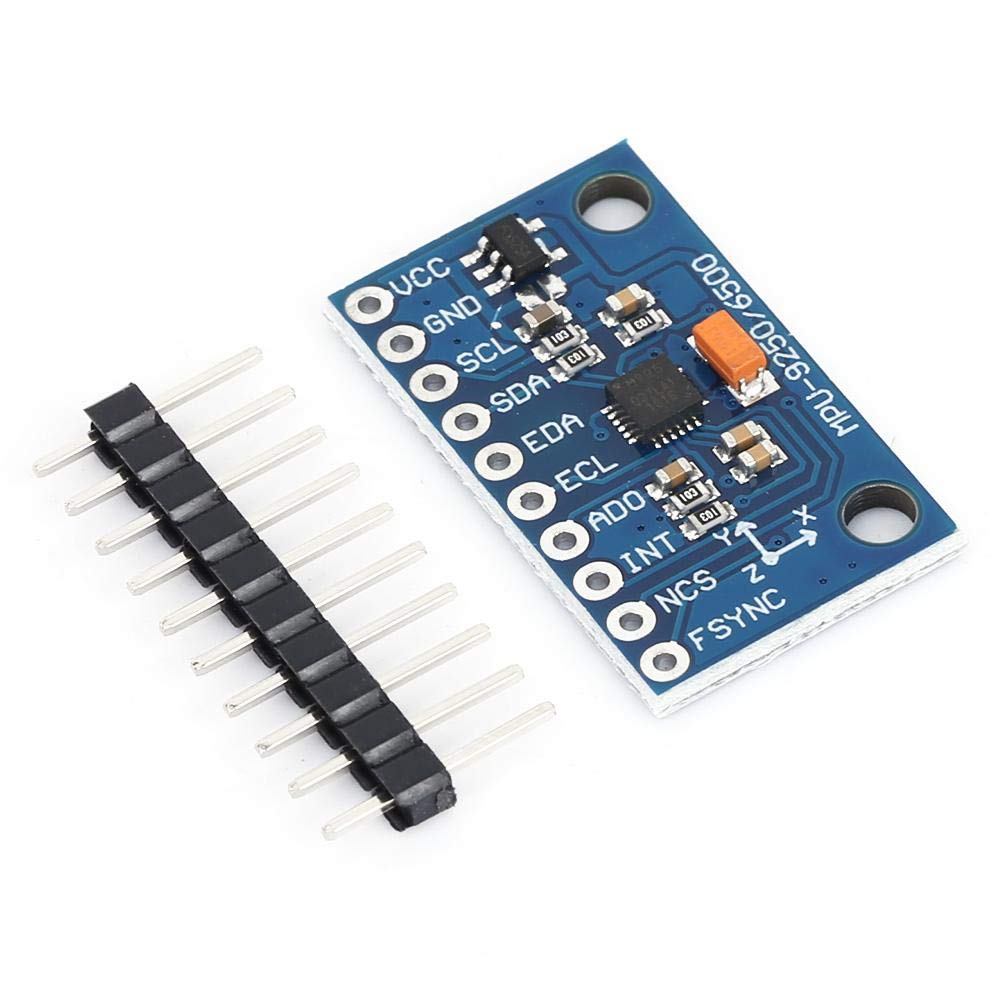


Рисунок 1.11 – Модуль гироскопа и акселерометра GY-9255 (MPU9255)

Для получения информации о данных модулях использовалась техническая спецификация [22] и источники [18, 19, 20, 21].

Как видно из сравнения, все рассмотренные модули имеют схожие характеристики. Отличительным является модуль GY-91, который помимо акселерометра и гироскопа также включает в себя барометр, что является важным параметром, если у микроконтроллера недостаточно выходов для подключения отдельного модуля барометра.

### **1.4.3 Обзор барометров**

Барометр в БПЛА используется для измерения атмосферного давления и высоты полёта, что обеспечивает функцию удержания высоты. При наличии барометра дрон способен зависать в одной точке на определённой высоте. В таблице 1.7 приведены сравнительные характеристики наиболее популярных модулей барометров.

Таблица 1.7 — Сравнение барометров

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметры сравнения** | **GY-68** | **GY-91** | **GY-BM** |
| Напряжение питания | 1.8 – 3.6 В | 3 – 5 В | 1.7 – 3.6 В |
| Потребляемый ток | до 5 мА | до 10 мА | до 2 мА |
| Диапазон измерений барометра | 300 – 1100 гПа  (от +9000 до -500 м) | 300 – 1100 гПа  (от +9000 до -500 м) | 300 – 1100 гПа  (от +9000 до -500 м) |
| Точность измерений барометра | ±1 гПа | ±1 гПа | ±1 гПа |
| Интерфейсы | I2C | I2C, SPI | I2C, SPI |
| Рабочая температура | от -40 до +85 ºС | от -40 до +85 ºС | от -40 до +85 ºС |
| Размеры | 13 мм × 10 мм | 21 мм × 15 мм | 11 мм × 15 мм |

**GY-68** – это компактный датчик атмосферного давления и температуры на основе датчика BMP180, который построен на магниторезистивной технологии, производства Bosch. Он предназначен для измерения атмосферного давления и температуры в реальном времени.

Модуль имеет интерфейс I2C, который позволяет легко подключить его к микроконтроллеру или другому устройству. Он также оснащён встроенным конвертером аналого-цифрового сигнала (ADC), который обеспечивает точное измерение атмосферного давления и температуры.

GY-68 имеет высокую точность измерения атмосферного давления (до 0.01 гПа) и температуры (до 0.1 градуса Цельсия). Диапазон измерения давления составляет от 300 до 1100 гПа, а диапазон измерения температуры от -40 до +85 градусов Цельсия.

Изображение модуля GY-68 представлено на рисунке 1.12.

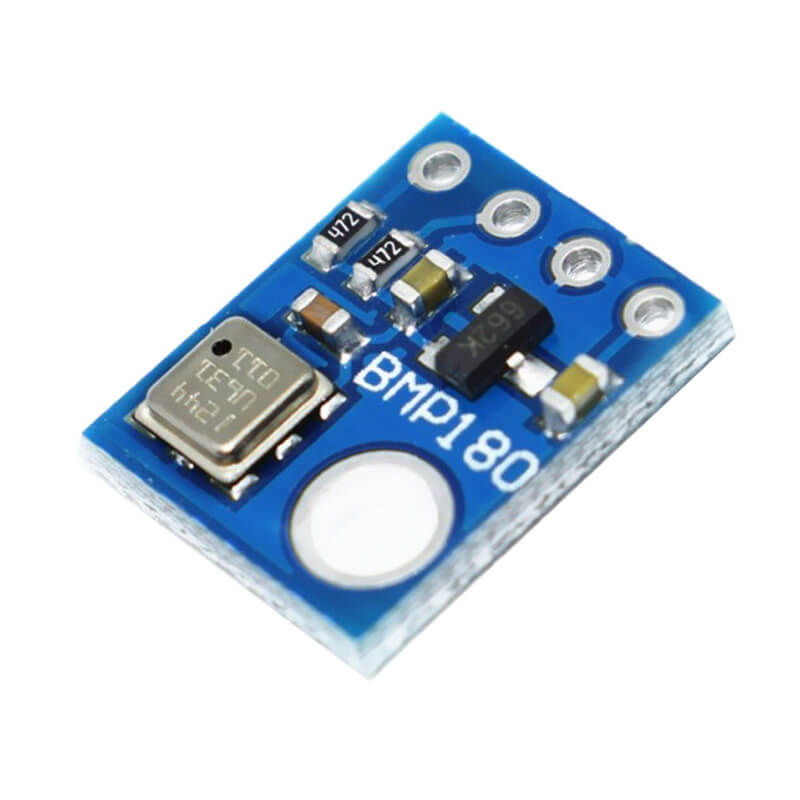


Рисунок 1.12 – Модуль барометра GY-68 (BMP180)

GY-91 – это многофункциональный модуль, который был описан в разделе 1.4.2.

Дальнейшим развитием GY-68 является модуль GY-BM, который представляет из себя высокоточный цифровой измеритель атмосферного давления на базе чипа BMP280 от фирмы BOSH. После изготовления каждый датчик проходит индивидуальную калибровку в заводских условиях. Его малые размеры, низкое энергопотребление и высокая измерительная способность позволили завоевать популярность среди множества разработчиков Arduino-проектов. Модуль BMP280 был разработан фирмой как более технологичная модель своего предшественника BMP180. Данная модификация, в отличие от своего младшего брата, предоставляет пользователю целых 2 последовательных интерфейса обмена данными (SPI и I2C), а также 2 режима работы: режим низкого энергопотребления и режим высокой производительности.

На рисунке 1.13 представлено изображение модуля GY-BM.

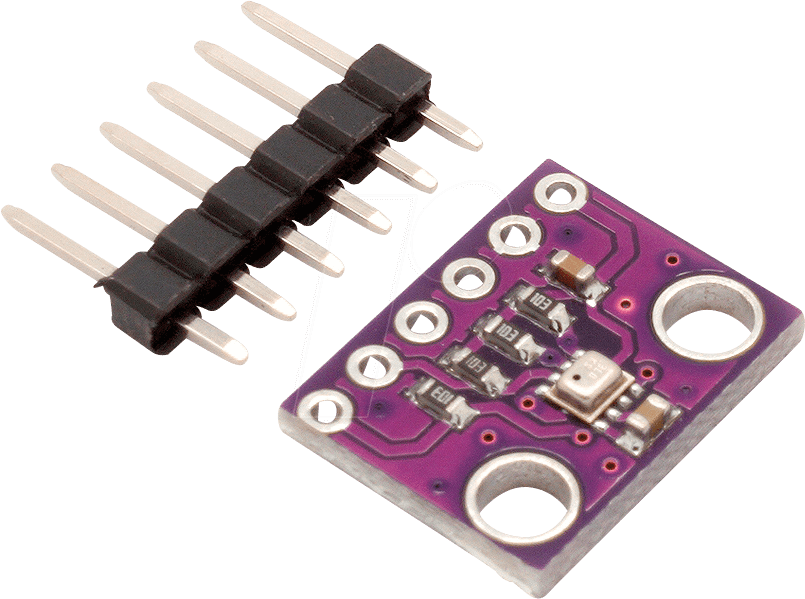


Рисунок 1.13 – Модуль барометра GY-BM (BMP280)

Для получения информации о модулях барометров использовалась техническая спецификация [24] и источники [19, 20, 23, 25].

После рассмотрения модулей барометров видно, что все они имеют схожие характеристики. Модуль GY-BM является более новой моделью модуля GY-68, поэтому является более предпочтительным. Модуль GY-91 совмещает в себе барометр, акселерометр и гироскоп, что даёт ему преимущество при установке на небольшом устройстве.

### **1.4.4 Обзор модулей GPS**

GPS (система глобального позиционирования) – спутниковая система навигации, обеспечивающая измерение расстояния, времени и определение местоположения во всемирной системе координат.

Определение местоположения устройства происходит путём измерения моментов времени приёма синхронизированного сигнала от навигационных спутников антенной потребителя. Для определения трёхмерных координат GPS-приёмнику нужно принимать сигнал четырех или более GPS-спутников.

Модуль GPS необходим для удержания позиции квадракоптера в пространстве. Это значит, что, если ветер его сдует на некоторое расстояние в сторону (обычно это сантиметры или десятки сантиметров), бортовая электроника выполнит возврат летательного аппарата в ту самую точку где он находился первоначально. Эта функция очень упрощает управление квадрокоптером и считается необходимой для качественной видеосъёмки.

Кроме этого, дроны, оснащённые GPS, как правило способны выполнять дальние полёты по заранее заданной программе, без участия оператора и не требуя управления пультом.

В таблице 1.8 приведены сравнительные характеристики наиболее популярных модулей GPS.

Таблица 1.8 — Сравнение модулей GPS

|  |  |  |
| --- | --- | --- |
| **Параметры сравнения** | **NEO-6M** | **ATGM336H** |
| Прием сигнала | BDS / GPS / GLONASS / GALILEO / QZSS / SBAS | BDS / GPS / GLONASS / GALILEO / QZSS / SBAS |
| Напряжение питания | 3 – 5 В | 5 В | |
| Максимальный потребляемый ток | 45 мА | 25 мА | |
| Время холодного старта | 38 с | 35 с | |
| Время горячего старта | 1 с | 1 с | |
| Точность позиционирования | до 5 м | < 2 м | |
| Частота обновления | 5 Гц | 1 – 10 Гц | |
| Чувствительность | 162 дБ | 162 дБ | |
| Максимальная высота | 18000 м | 18000 м | |
| Скорость подключения по UART | 9600 бод | 115200 бод | |
| Интерфейс | UART | UART | |
| Рабочая температура | от -40 до +85 ºС | от -40 до +85 ºС | |
| Размеры | 30 мм × 23 мм | 10.1 мм × 9.7 мм | |
| Вес комплекта | 18 г | 21 г | |

Для приёма GPS сигнала и определения местоположения устройства возможно использование GPS модуля NEO-6M. Данный модуль является популярным выбором для любительских устройств. Основой модуля является чип NEO-6M-0-001, который позволяет отслеживать до 22 спутников одновременно на 50 каналах, обладает высокой чувствительностью и энергопотреблением всего 45 мА. Определение местоположения происходит до 5 раз в секунду с точностью в 2.5 м.

Для подключения модуль использует UART на скоростях от 4800 до 230400 бод. Напряжение питания варьируется в диапазоне от 3.3 до 5 в, что позволяет подключить его без дополнительных микросхем сдвига логического уровня как к Arduino, так и к Raspberry Pi.

Изображение модуля NEO-6M представлено на рисунке 1.14.

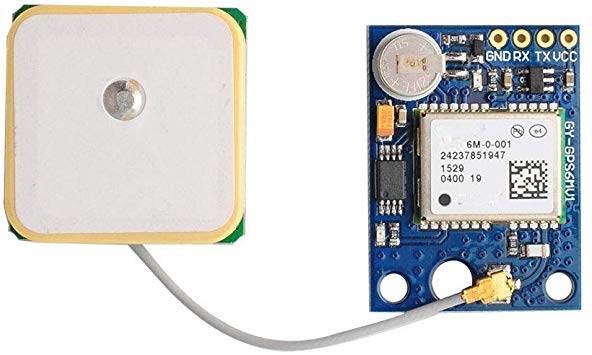


Рисунок 1.14 – Модуль GPS NEO-6M

Модуль GPS ATGM336H – GPS-модуль, позволяющий получать координаты устройств с помощью системы глобального позиционирования. Модуль построен на базе GPS-чипа AT6558, имеет встроенную память и резервную батарею для хранения настроек.

AT6558 – это спутниковый навигационный модуль с 32 каналами отслеживания, которые могут одновременно принимать до шести спутников используемые для навигационных систем. Данный модуль обладает высокой чувствительность и низким энергопотреблением. Подходит для автомобильной и переносной навигации. Изображение модуля GPS ATGM336H представлено на рисунке 1.15.



Рисунок 1.15 – Модуль GPS ATGM336H

Для получения информации о GPS модулях использовалась техническая спецификация [26, 28] и источники [27, 29].

Были рассмотрены модули GPS NEO-6M и GPS ATGM336H и их характеристики. Как видно, модуль GPS ATGM336H имеет большую частоту обновления, скорость подключения по UART и точность позиционирования, чем, несомненно, превосходит свой аналог GPS NEO-6M. Модуль GPS ATGM336H компенсирует свои недостатки меньшим потреблением тока. По всем остальным параметрам модули имеют схожие характеристики.

### **1.4.5 Обзор двигателей**

Двигатели БПЛА – это устройства, которые создают подъёмную силу, необходимую для полёта летательного аппарата. В зависимости от модели и назначения, двигатели могут иметь различные характеристики, такие как: размер, мощность, тип и другие. Наиболее распространёнными типами двигателей для дронов являются электрические двигатели, которые, в свою очередь, подразделяются на коллекторные (щёточные) и бесколлекторные (бесщёточные) [30]. Основное отличие коллекторного двигателя от бесколлекторного заключается в его конструкции.

Двигатель коллекторного типа состоит из корпуса, внутри которого есть плюс и минус (одна фаза). Его основная задача – преобразование электроэнергии в механическую энергию или наоборот. Он неподвижен и имеет щеточно-коллекторный узел. Коллектор представляет собой набор контактов, которые расположены на вращающемся элементе (роторе). Щётка – скользящий контакт, который расположен на неподвижном элементе (статоре). В движение приводится ротор с помощью щёток, подающих постоянный ток на обмотку. Направление вращения зависит от полярности.

К плюсам коллекторных двигателей можно отнести: небольшой вес и габариты, низкая цена, простота ремонта и обслуживания. Из минусов можно выделить: низкий коэффициент полезного действия (КПД), малая скорость вращения, перегрев, маленький ресурс, хрупкость.

Бесколлекторный электромотор состоит из вращающегося ротора с постоянными магнитами и неподвижного статора с электромагнитами. Смена направления вращения производится путём смены полярности. Такие электромоторы обладают разным числом полюсов, и чем их больше, тем медленнее, но со значительным усилием, вращается ротор.

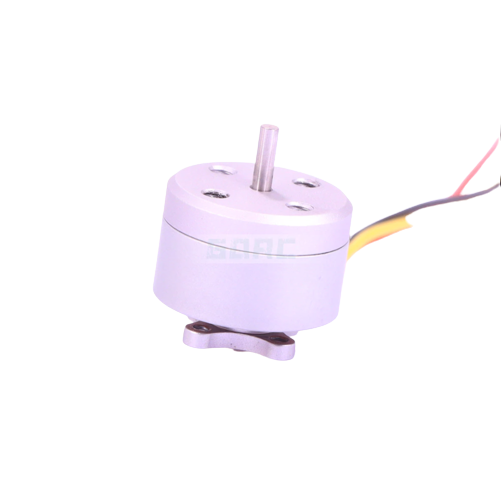
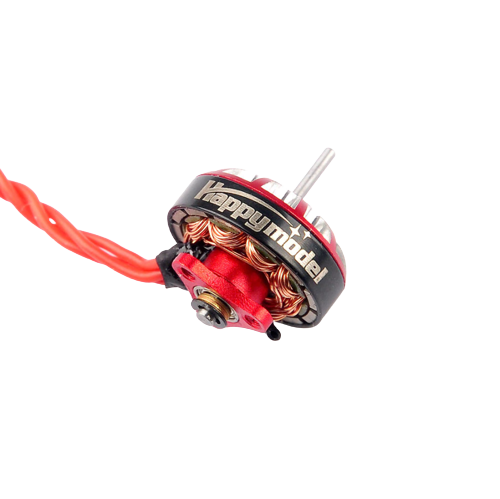
Бесколлекторные двигатели имеют следующие плюсы: большая скорость вращения, износоустойчивость, стойкость к внешним воздействиям, высокий КПД, долговечность. По сравнению с коллекторными двигателями, бесколлекторные являются более тихими, а также меньше весят и нагреваются. Из недостатков можно выделить: высокая цена, сложность ремонта.

Коллекторные двигатели применяются обычно на дронах начального уровня. В дронах средней и премиальной категории преимущественно устанавливаются двигатели бесколлекторного типа.

Для сравнения были выбраны двигатели бесколлекторного типа, результаты сравнения представлены в таблице 1.9. На рисунке 1.16 приведено графическое изображение сравниваемых двигателей.

Таблица 1.9 — Сравнение двигателей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Параметры сравнения** | **LD-Power 1106 4200KV** | **EMAX 1104 4300KV** | **HappyModel EX1102 9000KV** | **SKYSTARS KOKO 1404 3000KV** | **EMAX RS1408 2300KV** |
| Тип | бесколлекторный | бесколлекторный | бесколлекторный | бесколлекторный | бесколлекторный |
| Диаметр вала | 1.5 мм | 1.5 мм | 1.5 мм | 1.5 мм | 1.5 мм |
| Количество оборотов на 1 вольт | 4200 | 4300 | 9000 | 3000 | 2300 |
| Питание | 2S – 4S LiPo | 2S – 3S LiPo | 1S – 2S LiPo | 3S – 4S LiPo | 4S LiPo |
| Максимальный ток | 17.5 А | 12 А | 6.5 А | 20 А | 30 А |
| Максимальная мощность | 120 Вт | 80 Вт | 70 Вт | 130 Вт | 200 Вт |
| Максимальная тяга | 190 г | 120 г | 40 г | 280 г | 560 г |
| Рекомендуемый пропеллер | 2 – 3 дюйма | 2 – 3 дюйма | 1.9 – 2.5 дюйма | 3 – 4 дюйма | 2 – 4 дюйма |
| Размер | 11 × 6 мм | 11 × 4 мм | 11 × 2 мм | 11 × 4 мм | 11 × 8 мм |
| Вес | 7.3 г | 5.5 г | 2.5 г | 12.8 г | 12 г |

*а) б)* * в)* **

*г)* *д)*

Рисунок 1.16 – Двигатели: *а* – LD-Power 1106 4200KV, *б* – EMAX 1104 4300KV, *в* – HappyModel EX1102, *г* – SKYSTARS KOKO 1404 3000KV,

*д* – EMAX RS1408

Для получения информации о рассмотренных двигателях использовались источники [31, 33, 35], а также товары [32, 34, 36, 37, 38].

### **1.4.6 Обзор регуляторов оборотов двигателей**

Electronic Speed Controller (ESC) – это электронный регулятор скорости, который управляет скоростью вращения двигателей [39]. Это важная функция, которая позволяет управлять полётом БПЛА, изменяя скорость вращения двигателей в зависимости от требований. ESC принимают сигналы от полётного контроллера и передают эти сигналы каждому из двигателей, чтобы контролировать скорость и направление их вращения. ESC также обеспечивает защиту от перегрузок и коротких замыканий, чтобы защитить двигатели и другие компоненты системы от повреждений

Принцип работы ESC заключается в преобразовании постоянного тока, поступающего от аккумулятора, в переменный ток, который подаётся на двигатель. Для управления скоростью вращения двигателя, ESC использует модуляцию ширины импульсов (ШИМ), где длина импульса управляет скоростью вращения двигателя. Когда ESC получает сигнал от приёмника или полётного контроллера, он преобразует этот сигнал в сигнал ШИМ, который отправляется на двигатель. Сигнал ШИМ состоит из серии коротких импульсов, которые имеют разную длину, соответствующую желаемой скорости вращения двигателя. ESC также может иметь встроенные функции защиты, такие как защита от перегрузки, защита от короткого замыкания, защита от низкого заряда аккумулятора и другие. Кроме того, ESC может иметь настройки, позволяющие пользователю изменять настройки скорости, торможения и другие параметры.

Важно отметить, что каждый тип двигателя может требовать различных настроек ESC. Поэтому, при выборе ESC необходимо убедиться, что он совместим с используемым типом двигателя и сигналами управления, которые будут использоваться в системе [40].

Первый параметр, на который нужно обратить внимание при выборе регулятора – это максимальный ток, он измеряется в амперах. Моторы потребляют энергию при вращении, если им нужен ток больше, чем может выдать регулятор, то регулятор начнёт греться и в итоге выйдет из строя. При использовании регуляторов с большим запасом по току мы получаем дополнительный вес, увеличенный размер и, возможно, большую цену. Достоинство такого подхода – меньше шанс выхода регуляторов из строя, а эффективность работы регуляторов немного выше.

Ещё одна вещь, о которой стоит задуматься – какой ток может выдать аккумулятор. Большинству дронов достаточно регуляторов на 20 А, так как четырехбаночные аккумуляторы ёмкостью 1300 – 1500 мА\*ч не выдают большую мощность продолжительное время (то есть 20А регулятор выдерживает подобные нагрузки).

Emax BlHeli ESC – это электронный регулятор скорости, разработанный компанией Emax для использования в различных типах летательных аппаратов, таких как квадрокоптеры, самолёты и мультикоптеры. Он основан на программном обеспечении BlHeli, который предоставляет возможность настройки и управления различными параметрами регулятора.

Особенности регулятора Emax BlHeli ESC включают в себя:

1. Быстродействие: высокая частота обновления до 500 Гц, что обеспечивает быструю реакцию на изменения сигналов управления и плавное управление двигателями.

2. Надёжность: встроенная защита от перегрузки, короткого замыкания, перегрева и низкого уровня напряжения аккумулятора.

3. Настройка: возможность изменения настроек, таких как максимальная скорость, торможение, стартовый режим и другие параметры, позволяющие пользователю настроить регулятор под свои потребности и требования.

4. Поддержка различных типов двигателей: Emax BlHeli ESC поддерживает различные типы двигателей, в том числе бесколлекторные моторы (BLDC) и моторы постоянного тока (DC).

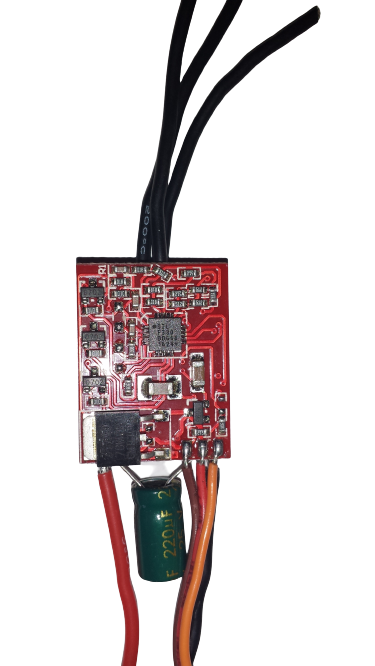
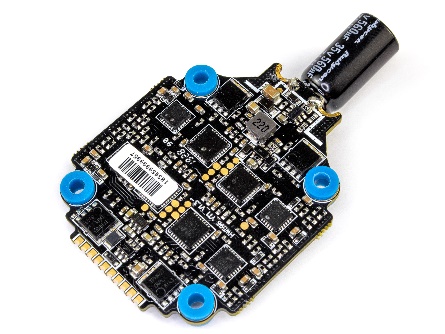
Среди аналогов Emax BlHeli ESC можно выделить:

1. Hobbywing XRotor Micro [41]: поддерживает множество настроек и функций, имеет высокую частоту обновления до 600 Гц и защиту от различных сбоев.

2. Flycolor Raptor [42]: компактный регулятор с поддержкой функции OneShot125 и возможностью настройки различных параметров.

3. T-Motor V2 [43]: регулятор с высоким качеством сборки, возможностью настройки и защитой от перегрузки, короткого замыкания и других сбоев.

Все рассмотренные регуляторы классифицируются по максимальному току. Графическое изображение регуляторов представлено на рисунке 1.17.

*а)**б)* 

*а)*  *б)* 

Рисунок 1.17 – Регуляторы оборотов: *а* – Emax BlHeli ESC,

*б* – Hobbywing XRotor Micro, *в* – Flycolor Raptor, *г* – T-Motor V2

# **2 РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ**

Изучив предметную область разрабатываемой системы, были разработаны основные требования, которые должны быть выполнены при реализации дипломного проекта. Для упрощения разработки системы разобьём её на структурные блоки.

## **2.1 Описание основных блоков устройства**

В результате разделения летательного устройства были определены следующие блоки:

− блок обработки информации контроллера полёта;

− блок определения ориентации в пространстве;

− блок определения высоты;

− блок приёма и передачи радиосигнала контроллера полёта;

− блок индикации;

− блок управления двигателями;

− блок питания контроллера полёта.

Устройство дистанционного управления подразделяется на следующие блоки:

− блок обработки информации устройства управления;

− блок управления летательным аппаратом;

− блок взаимодействия с настройками управления;

− блок приёма и передачи радиосигнала устройства управления;

− блок отображения информации;

− блок питания устройства управления.

Структурная схема, иллюстрирующая перечисленные блоки и связи между ними, приведена на чертеже ГУИР.400201.022 Э1.

Блоки были выделены таким образом, чтобы каждый из них выполнял определённую задачу и обеспечивали корректную работу системы в целом. Далее рассмотрим принцип работы и задачи каждого из перечисленных выше блоков, а также их взаимодействие между собой.

## **2.2 Блок обработки информации контроллера полёта**

Блок обработки информации контроллера полёта – это основной блок разрабатываемого летательного устройства, центральным звеном которого является микроконтроллер. Данный блок производит большую часть вычислений и связывает между собой другие блоки. Микроконтроллер предназначен для сбора и обработки данных, получаемых с других блоков контроллера полёта, а также управления этими блоками в реальном времени.

Блок обработки информации контроллера полёта выполняет следующий ряд задач, связанных с передачей данных и сбором информации:

1. Получение и обработка данных, поступающих с блока радиопередачи контроллера полёта.

2. Сбор и обработка данных, полученных с блока определения ориентации в пространстве.

3. Сбор и обработка данных, поступающих с блока определения высоты.

4. Взаимодействие с блоком управления двигателей.

5. Передача управляющих сигналов на блок индикации.

В общем случае, блок обработки информации на БПЛА отвечает за последовательный сбор данных с датчиков устройства и получения команд с блока радиопередачи. После этого блок производит все необходимые расчёты и преобразования, а затем передаёт управляющие сигналы на блок управления двигателями и блок индикации.

В задачи рассматриваемого блока также входит предварительная конфигурация и инициализация других блоков, таких как: блок приёма и передачи радиосигнала, блок определения ориентации в пространстве, блок определения высоты, блок управления двигателями.

Питание, необходимое для включения и стабильной работы блока обработки информации, будет поступать от блока управления двигателями.

## **2.3 Блок определения ориентации в пространстве**

Блок определения ориентации в пространстве использует инерциальные датчики для измерения скорости, ускорения и угловой скорости, чтобы определить положение и ориентацию БПЛА в пространстве. Данный блок является наиболее важным для стабилизации устройства так как на основе измеряемых в этом блоке параметров система корректирует работу дрона, чтобы он не отклонялся от заданного направления.

Блок содержит инерциальные датчики, такие как трёхосевой акселерометр и гироскоп, которые работают вместе и обеспечивают точное измерение движения и ориентации летательного устройства.

Данные, получаемые в блоке определения ориентации в пространстве, поступают на блок обработки информации контроллера полёта и обрабатываются с помощью специальных алгоритмов, таких как фильтр Калмана.

Конфигурируется блок с помощью блока обработки информации контроллера полёта при запуске БПЛА с целью минимизации погрешностей и обеспечения стабильной работы. Для этого вычисляются отклонения по трём осям устройства в неподвижном состоянии.

## **2.4 Блок определения высоты**

Функцией данного блока является измерение высоты, на которой находится летательное устройство в реальном времени, что позволяет ему правильно позиционировать себя относительно земли и обеспечивает безопасные и эффективные полёты и посадку. На основе этих измерений выполняется команда «Удержания высоты», при которой устройство зависает на одной, заданной высоте.

При проектировании БПЛА могут быть использованы различные методы для определения высоты. Один из таких методов – это измерение атмосферного давления, которое уменьшается с увеличением высоты над уровнем моря.

Для измерения давления и определения высоты будет использован барометрический датчик. Измеряемые параметры в блоке определения высоты передаются блоку обработки информации контроллера полёта, который их анализирует и вносит необходимые изменения в работу дрона.

## **2.5 Блок приёма и передачи радиосигнала контроллера полёта**

Устройство дистанционного управления должно взаимодействовать с летательным устройством. Для этого необходимо организовать связь посредством радиосигнала, что предполагает разработку блока приёма и передачи радиосигнала.

Радиопередача между дроном и устройством управления позволяет оператору дистанционно управлять полётом летательного устройства, получать данные и контролировать его положение в режиме реального времени. Данный блок должен быть установлен как на устройстве управления, так и на самом БПЛА. Между двумя блоками радиопередачи настраивается соединение на заданном радиоканале, что позволяет им обмениться информацией.

Конфигурируется рассматриваемый блок с помощью блока обработки информации. Данные, которые необходимо отправить на блок приёма и передачи радиосигнала устройства управления, поступают от блока обработки информации контроллера полёта. Если блок получает данные от блока приёма и передачи радиосигнала устройства управления, то он передаёт их на блок обработки информации.

## **2.6 Блок индикации**

Блок индикации необходим для информирования пользователя о состоянии работы устройства. Он состоит из пары светодиодов, каждый из которых имеет свой цвет. Комбинация светодиодов, а также время их мигания, позволяет сформировать различные сообщения пользователю.

Работа с блоком индикации осуществляется через блок обработки информации контроллера полёта, который отдаёт необходимые команды по включению/отключению нужного светодиода, тем самым подавая на блок индикации необходимое питание.

## **2.7 Блок управления двигателями**

Блок управления двигателями отвечает за контроль и регулирование работы двигателей. Он является одним из ключевых в работе БПЛА так как обеспечивает перемещение устройства в воздушном пространстве, позволяет ему подняться и держаться над землёй.

Блок управления двигателями в разрабатываемом комплексе представляет собой четыре бесколлекторных мотора и регулятор оборотов для каждого из моторов.

Данный блок управляется блоком обработки информации контроллера полёта, который передаёт управляющие ШИМ сигналы для запуска того или иного двигателя. Питание, необходимое для работы двигателей, поступает от регуляторов оборотов, а к ним от – блока питания.

## **2.8 Блок питания контроллера полёта**

Блок питания необходим для обеспечения устройства электроэнергией. Питание является важной составляющей любого устройства, поскольку без электроэнергии оно не может работать. Надлежащее питание обеспечивает работу всех компонентов устройства и позволяет ему выполнять свои функции. В зависимости от типа устройства и его конструкции, могут использоваться различные источники питания, такие как аккумуляторные баттареи, электрическая сеть, солнечные баттареи и так далее.

В качестве источника питания будут использоваться аккумуляторные баттареи так как разрабатываемый дрон должен функционировать без прямого подключения к электрической сети.

Блок питания, установленный на БПЛА, должен подавать разное питание на блок обработки информации и блок управления двигателями. Блок управления двигателями потребляет больше энергии, поэтому питание будет поступать на этот блок, а в затем идти на блок обработки информации и другие блоки летательного устройства.

## **2.9 Блок обработки информации устройства управления**

Блок обработки информации устройства управления – это основной блок разрабатываемого устройства управления, центральным звеном которого является микроконтроллер. Данный блок необходим для связи всех блоков устройства, выполнения вычислений и преобразований. Микроконтроллер предназначен для сбора и обработки данных, получаемых с других блоков устройства управления, а также отправки этих данных на блок приёма и передачи радиосигнала.

Блок обработки информации устройства управления выполняет следующий ряд задач, связанных с передачей данных и сбором информации:

1. Сбор и обработка данных, поступающих с блока управления летательным аппаратом.

2. Обработка данных, поступающих с блока взаимодействия с настройками управления.

3. Передача данных на блок отображения информации.

4. Передача данных на блок приёма и передачи радиосигнала.

Блок обработки информации на устройстве управления отвечает за последовательный сбор данных с блока управления летательным аппаратом и блока взаимодействия с настройками управления. После этого блок производит все необходимые расчёты и преобразования, а затем передаёт управляющие сигналы на блок приёма и передачи радиосигнала, а также блок отображения информации.

В задачи рассматриваемого блока также входит предварительная конфигурация и инициализация других блоков устройства управления.

Питание, необходимое для включения и стабильной работы блока обработки информации, будет поступать от блока питания устройства управления.

## **2.10 Блок управления летательным аппаратом**

Блок управления летательным аппаратом является одним из наиболее важных в разрабатываемом устройстве управления так как с помощью него пользователь задаёт необходимые команды, которые должен выполнять БПЛА.

Блок представляет собой пару джойстиков, один из которых отвечает за поднятие летательного устройства на определённую высоту, другой – задаёт направление движения. В состав блока также могут входить дополнительные тумблеры или кнопки, с помощью которых пользователь может задать необходимые команды для управления дроном.

Все изменения в блоке управления летательным аппаратом передаются на блок обработки информации устройства управления с помощью аналоговых и цифровых сигналов.

## **2.11 Блок взаимодействия с настройками управления**

Блок взаимодействия с настройками управления косвенно связан с блоком управления летательным аппаратом так как позволяет изменять параметры управления. Блок помогает пользователю настроить управление под себя и скорректировать значения, которые будут переданы на летательное устройство.

Блок представлен набором кнопок и тумблеров, каждый из которых отвечает за свой параметр.

Все изменения в блоке взаимодействия с настройками управления передаются на блок обработки информации устройства управления, где в последствии сохраняются в память устройства и применяются при обработке данных.

## **2.12 Блок приёма и передачи радиосигнала устройства управления**

Блок приёма и передачи радиосигнала устройства управления выполняет функции, аналогичные блоку приёма и передачи радиосигнала контроллера полёта БПЛА, описанные в пункте 2.5.

Рассматриваемый блок устанавливается на устройстве управления и необходим для передачи команд летательному устройству по радиосвязи. Данные для передачи поступают от блока обработки информации устройства управления.

## **2.13 Блок отображения информации**

Блок отображения информации представляет собой ЖКИ-экран, устанавливаемый на устройство управления. Главная функция блока – это информирование пользователя о текущих параметрах. Блок позволяет пользователю узнать состояние заряда устройства, посмотреть данные, поступающие с блока управления летательным аппаратом, а также увидеть ошибки, при их наличии.

Вся отображаемая информация поступает с блока обработки данных устройства управления в реальном времени.

## **2.14 Блок питания устройства управления**

Блок питания устройства управления выполняет функции, аналогичные блоку питания контроллера полёта БПЛА, описанные в пункте 2.8.

На устройстве управления необходимо предусмотреть преобразование входного напряжения до двух значений: 3.3В и 5В так как большинство элементов работают именно с этими значениями напряжения. В качестве источника питания будут использоваться аккумуляторные батареи.

# **3 РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ СХЕМЫ**

В предыдущем разделе (см. раздел 2 Разработка структурной схемы) была спроектирована структура разрабатываемого комплекса и приведено описание основных блоков системы. В данном разделе блоки будут описаны с точки зрения разработки функций, которые реализуются в данном дипломном проекте. Функциональная схема устройства представлена на чертеже ГУИР.400201.022 Э2.

**3.1 Блоки летательного устройства**

Летательное устройство включает в себя следующие блоки:

− блок обработки информации контроллера полёта;

− блок определения ориентации в пространстве;

− блок определения высоты;

− блок приёма и передачи радиосигнала контроллера полёта;

− блок индикации;

− блок управления двигателями;

− блок питания контроллера полёта.

### **3.1.1 Блок обработки информации контроллера полёта**

Блок обработки информации контроллера полёта – это основной блок разрабатываемого летательного устройства, центральным звеном которого является микроконтроллер. По результатам анализа в пункте 1.3 была выбрана модель микроконтроллера ATmega328P-AU.

С точки зрения основных функциональных возможностей, микроконтроллер включает в себя [44]:

1. Центральный процессор (CPU). ATmega328P-AU имеет 8-битный RISC-процессор с тактовой частотой до 20 МГц. Он также поддерживает инструкции, которые позволяют быстро выполнять операции с памятью и передавать управление другим частям программы.

Преимущественно CPU состоит из арифметико-логического устройства (АЛУ), общих регистров назначения (РОН), регистров статуса, регистра инструкций, счётчика команд и декодера команд.

2. Встроенная память. Микроконтроллер имеет 32 Кбайт флэш-памяти для хранения программного кода, 2 Кбайт оперативной памяти (SRAM) и 1 Кбайт EEPROM для хранения данных.

Оперативная память используется для временного хранения данных и инструкций во время работы микроконтроллера. SRAM может хранить переменные, стек вызовов, временные данные и другую информацию, необходимую для выполнения программы или алгоритма.

Постоянная FLASH память хранит все исполняемые инструкции разработанного программного кода и неизменяемые данные. Постоянная память может быть перезаписана с помощью программного кода или с помощью внешнего программатора с использованием интерфейса JTAG.

3. Входы/выходы (I/O). ATmega328P-AU имеет 23 программируемых входа/выхода (GPIO), каждый из которых может быть настроен как вход или выход. GPIO могут использоваться для подключения различных периферийных устройств, таких как датчики, дисплеи и другие. Они предназначены для простых в выполнении функций, таких как подача сигнала enable, переключения ключей, подача импульсов. На входах/выходах можно задавать необходимость подтягивающих резисторов. В большинстве I/O разделяют функции с другими подсистемами микроконтроллера.

4. Аналого-цифровой преобразователь (АЦП). Микроконтроллер имеет 10-битный АЦП с 8 входами, который позволяет преобразовывать различные значения напряжения в цифровой формат для их контроля и анализа.

Сигнал, поступивший на вход АЦП захватывается с помощью специальной схемы. Далее управляющий автомат АЦП выдаёт значение, поступающее на вход цифро-аналогового преобразователя. Входной сигнал АЦП и выходной сигнал внутреннего ЦАП сравниваются с помощью компаратора и в зависимости от результата сравнения подбирается новое цифровое значение, пока оно не приблизится к уровню, совпадающему с измеряемым сигналом. АЦП имеет несколько каналов, которые измеряют несколько разных входных напряжений циклически последовательно их переключая.

Максимальный и минимальный уровень напряжения, которому соответствует максимальное и минимальное цифровое значение соответственно определяется с помощью опорных напряжений. Поэтому на вход АЦП должно поступать напряжение, которое меньше опорного.

5. Таймеры/счётчики. Таймеры и счётчики могут использоваться для измерения времени, генерации сигналов, таких, как ШИМ-сигнал, и других задач. ATmega328P-AU имеет три 8-битных таймера/счётчика и один 16-битный таймер/счётчик. Каждый из них имеет свои режимы работы и может быть настроен на генерацию прерываний, что позволяет создавать различные события в процессе работы микроконтроллера.

Для настройки таймеров в микроконтроллере ATmega328P-AU используются регистры управления таймерами.

6. ШИМ-генераторы. Микроконтроллер имеет 6 каналов ШИМ-генераторов, которые могут использоваться для управления яркостью светодиодов, скоростью двигателей и других задач. Более подробно ШИМ рассмотрено в пункте 3.3.6.

7. Коммуникационные интерфейсы. ATmega328P-AU поддерживает различные коммуникационные интерфейсы, такие как UART, SPI и I2C, что позволяет ему общаться с другими устройствами. Используемые в проекте интерфейсы рассмотрены в пункте 3.3.

Блок-схема микроконтроллера представлена на рисунке 3.1.

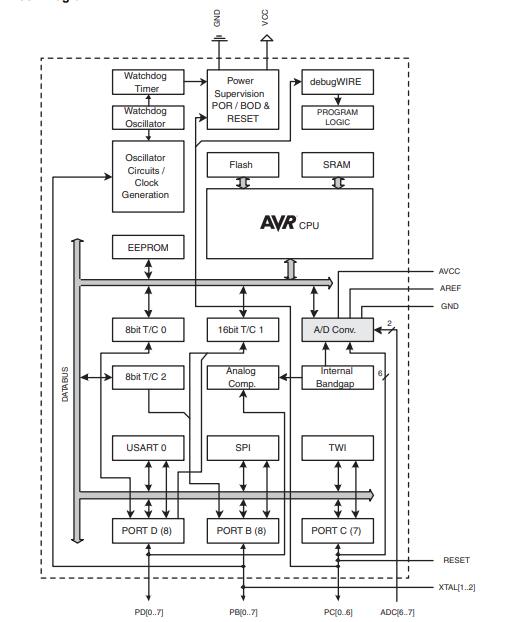


Рисунок 3.1 – Блок-схема микроконтроллера ATmega328P-AU

### **3.1.2 Блок определения ориентации в пространстве**

Блок определения ориентации в пространстве представляет собой совокупность инерционных датчиков: трёхосевой акселерометр и гироскоп. Исходя из обзора модулей акселерометра и гироскопа, представленного в пункте 1.4.2, для проекта была выбрана модель GY-521.

Модуль GY-521 реализует следующие функциональные возможности:

1. Измерение ускорения. Модуль GY-521 может измерять ускорение в трёх направлениях (по осям x, y и z) благодаря установленному на модуль трёхосевому акселерометру.

2. Измерение скорости вращения. Модуль GY-521 имеет гироскоп, который может измерять угол поворота или угловую скорость вращения объекта в трёх направлениях.

3. Функция DMP (Digital Motion Processor). Модуль GY-521 имеет встроенный цифровой процессор движения (DMP), который может выполнять ряд задач, таких как обработка данных с датчиков и расчёт ориентации объекта. Это может упростить процесс программирования и обработки данных.

4. Интерфейсы коммуникации. Модуль GY-521 поддерживает несколько интерфейсов коммуникации, включая I2C и SPI, что позволяет подключать его к широкому диапазону устройств и микроконтроллеров.  
 5. Переменная скорость передачи данных. Модуль GY-521 позволяет изменять скорость передачи данных между модулем и микроконтроллером, что может быть полезно для оптимизации производительности и энергопотребления.

Общение между блоком определения ориентации в пространстве и микроконтроллером осуществляется по интерфейсу I2C, описанному в разделе 3.3.1.

Схема модуля GY-521 [45] представлена на рисунке 3.2.

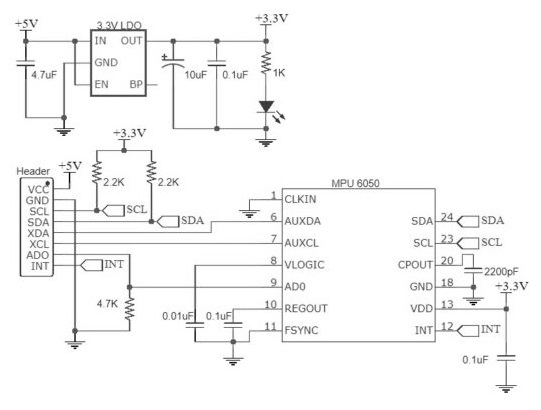


Рисунок 3.2 – Схема модуля GY-521

### **3.1.3 Блок определения высоты**

Блок определения высоты представлен в виде барометра. В качестве барометра был выбран модуль GY-68, исходя из рассмотренных характеристик в пункте 1.4.3.

Основные функциональные возможности модуля GY-68 включают:

1. Измерение атмосферного давления. Модуль GY-68 оснащён датчиком давления, который может измерять атмосферное давление в диапазоне от 300 до 1100 гПа. Измерение происходит с точностью до 0,02 гПа, что позволяет получить точные данные о текущем давлении и выполнить расчёты по определению высоты.

2. Измерение температуры. Модуль включает датчик температуры, который может измерять температуру в диапазоне от -40 до +85 градусов Цельсия. Измерение происходит с точностью до 0,5 градусов Цельсия.

3. Калибровка. Модуль GY-68 может быть откалиброван для повышения точности измерений. Калибровка происходит с использованием специальных алгоритмов, которые позволяют устранить ошибки, связанные с факторами окружающей среды.

4. Интерфейс I2C. Модуль GY-68 может быть легко подключен к микроконтроллеру через интерфейс I2C, что обеспечивает быстрый и простой способ получения данных от датчика. Более детально интерфейс I2C описан в разделе 3.3.1.

Схема модуля GY-68 [46] представлена на рисунке 3.3.

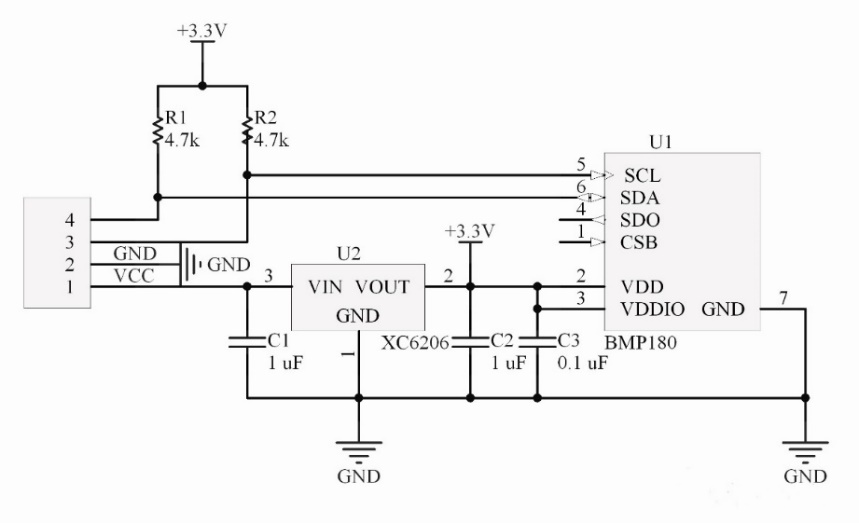


Рисунок 3.3 – Схема модуля GY-68

### **3.1.4 Блок приёма и передачи радиосигнала контроллера полёта**

Блок приёма и передачи радиосигнала контроллера полёта представляет собой радиомодуль, который позволяется передавать данные, используя радиосигнал. С точки зрения передачи радиосигнала, рассматриваемый блок реализует ведомое устройство. После рассмотрения радиомодулей, представленных на рынке, в пункте 1.4.1, была выбрана модель NRF24L01.

Блок-схема радиомодуля NRF24L01 [47] представлена на рисунке 3.4.

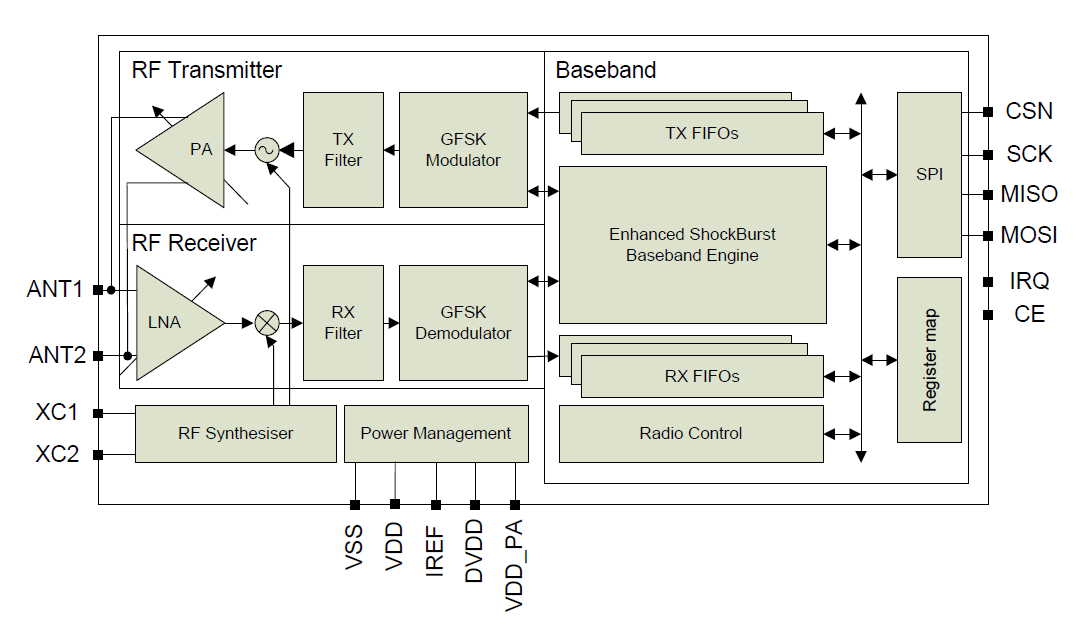


Рисунок 3.4 – Блок-схема радиомодуля NRF24L01

Некоторые из функциональных возможностей модуля NRF24L01 включают:

1. Беспроводная связь. Модуль позволяет обмениваться данными между двумя или более устройствами на расстоянии до 100 метров без использования проводов.

2. Конфигурация каналов. Модуль NRF24L01 имеет 126 различных каналов для беспроводной связи, что позволяет избежать помех от других беспроводных устройств, работающих на той же частоте.

3. Шифрование данных. NRF24L01 поддерживает шифрование данных для защиты от несанкционированного доступа к передаваемым данным.

4. Поддержка интерфейса SPI. Модуль поддерживает протокол SPI, что позволяет подключать его к микроконтроллеру, используя интерфейс протокола. Более детально интерфейс SPI описан в пункте 3.4.1.

Обмен данными между блоком приёма и передачи радиосигнала контроллера полёта и микроконтроллером осуществляется по интерфейсу SPI, описанному в пункте 3.3.2.

Передача данных между радиомодулями NRF24L01 осуществляется по протоколу ESB, который описан в пункте 3.3.4.

Модуль изменяет передаваемый сигнал, используя GFSK модуляцию, рассмотренную в пункте 3.3.5.

Схема радиомодуля NRF24L01 [48] представлена на рисунке 3.5.

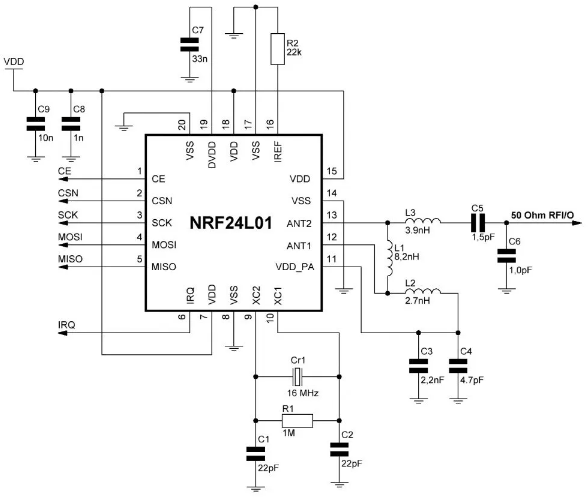


Рисунок 3.5 – Схема радиомодуля NRF24L01

### **3.1.5 Блок индикации**

Блок индикации представлен в виде двух светодиодов, устанавливаемых на летательное устройство. Светодиод (LED) – это электронный компонент, который преобразует электрическую энергию в световую энергию. В разрабатываемом комплексе блок выполняет функцию информирования о подключении БПЛА к устройству управления, а также отображения наличия ошибок в работе, если они присутствуют.

Светодиоды отличаются цветом свечения и, соответственно, длинной излучаемой волны, а также входным напряжением [49]. На рисунке 3.6 представлено распределение длины световой волны в зависимости от цвета светодиода.



Рисунок 3.6 – Цвет в зависимости от длины световой волны

Предполагается использовать светодиоды разного цвета, например: зелёный и красный.

### **3.1.6 Блок управления двигателями**

Блок управления двигателями состоит из самих двигателей и регуляторов оборотов двигателей. Всего в блоке устанавливается четыре бесколлекторных двигателя и четыре регулятора. Каждый двигатель подключается к отдельному регулятору скорости.

Двигатели выполняют функцию поднятия летательного устройства над землёй и позволяют ему перемещаться в воздушном пространстве. После рассмотрения нескольких моделей двигателей в пункте 1.4.5, была выбрана модель LD-Power 1106 4200KV.

Регулятор оборотов должен уметь взаимодействовать с выбранной моделью двигателя и необходим для обеспечения плавного старта и корректной работы мотора. В связи с этим, после рассмотрения моделей регуляторов в пункте 1.4.6 была выбрана Emax BlHeli ESC.

С точки зрения выполняемых функций, Emax BlHeli ESC обладает следующими возможностями [50]:

1. Работа с различными типами моторов, включая бесколлекторные двигатели.

2. Поддержка автоматической настройки. ESC может автоматически настраиваться под подключённый мотор, что упрощает процесс установки.

3. Защита от перегрева. Регулятор имеет встроенную внутреннюю защиту от перегрева, что позволяет предотвратить повреждения от высоких температур.

4. Обеспечение плавного старта двигателей. ESC поддерживает функцию «безболезненного старта», что уменьшает износ моторов и улучшает качество полёта.

Регулятор оборотов двигателей включает в себя:

1. Микроконтроллер. Встроенный в ESC микроконтроллер работает под управлением прошивки, например, BlHeli. Он управляет работой и параметрами регулятора.

2. MOSFET-транзисторы. Представляют собой полевой транзистор с изолированным затвором. Он работает как ключ: драйвер затвора подаёт сигнал – MOSFET открывается и подаёт ток на определённую область статора, там самым, заставляя магниты вращать колокол двигателя из-за того, что обмотка (статор) попеременно меняет напряжение в определённых частях так, чтобы магниты начинали притягиваться в определённый момент времени к определённой части статора – таким образом, колокол двигателя начинает вращаться. Все это происходит много раз за доли секунды.

3. Драйвер затвора. Драйвер затвора управляет работой MOSFET-транзисторов, о которых описано выше. Он определяет, когда полевому транзистору открыться, а когда закрыться.

4. Другие элементы, такие как: фильтрующие конденсаторы, датчик тока, стабилизатор напряжения.

На рисунке 3.7 приведено графическое изображение взаимодействия регулятора скорости и подключённого к нему двигателя.

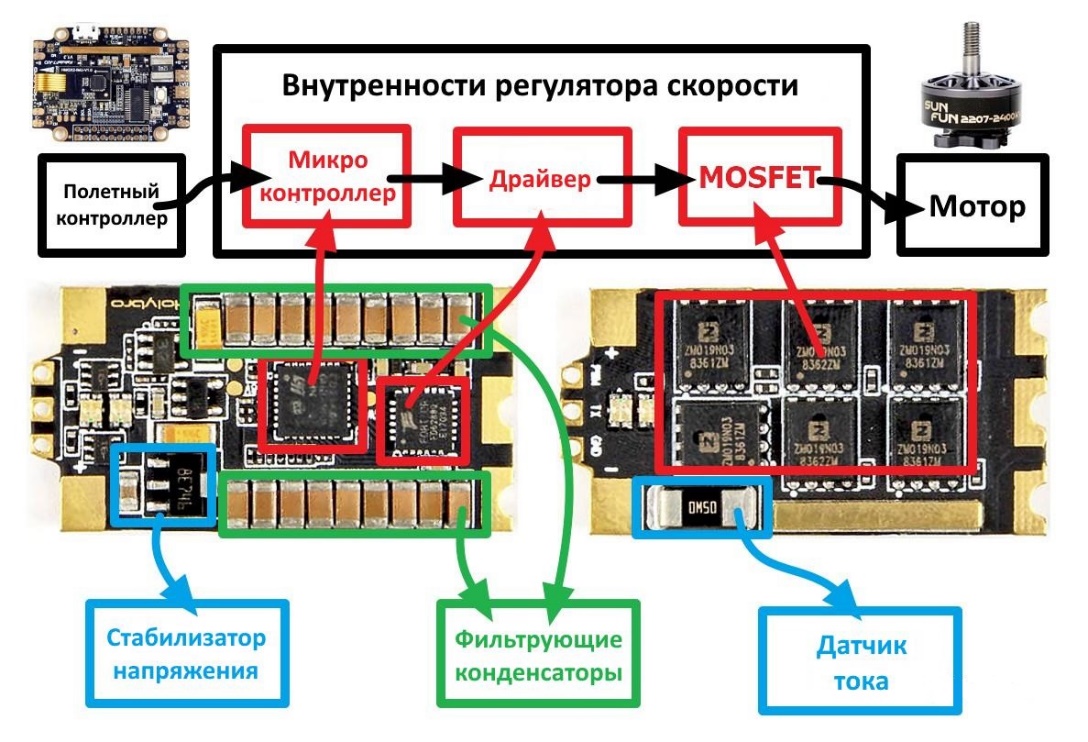


Рисунок 3.7 – Взаимодействие регулятора скорости и двигателя

### **3.1.7 Блок питания контроллера полёта**

Данный функциональный блок обеспечивает питанием датчики и элементы, установленные на летательное устройство. Блок состоит из аккумулятора, который выдаёт напряжение 7.4 В. В блоке также присутствуют дополнительные стабилизаторы и элементы, обеспечивающие преобразование напряжения до нужных значений и контролирующие правильность подключения питания.

Большинство датчиков и элементов, устанавливаемых на БПЛА, работают от напряжения 3.3 В и 5 В., следовательно, питание, которое поступает на эти элементы, должно быть преобразовано до 5 В и 3.3 В – эту функцию выполняют стабилизаторы напряжения.

Питание с блока поступает не напрямую, а проходит через блок управления двигателями. Напряжение, получаемое от аккумулятора (7.4 В), изначально идёт на четыре установленных регулятора скорости так как это основной потребитель энергии всей схемы. На ESC установлен преобразователь напряжения до 5 В, поэтому микроконтроллер и остальные элементы питаются от стабилизаторов напряжения, подключённых к выходу одного из регуляторов.

**3.2 Блоки устройства управления**

Устройство дистанционного управления подразделяется на следующие блоки:

− блок обработки информации устройства управления;

− блок управления летательным аппаратом;

− блок взаимодействия с настройками управления;

− блок приёма и передачи радиосигнала устройства управления;

− блок отображения информации;

− блок питания устройства управления.

### **3.2.1 Блок обработки информации устройства управления**

Блок обработки информации устройства управления представлен в виде микроконтроллера ATmega328P-AU и имеет схожий функционал с блоком обработки информации контроллера полёта, описанного в пункте 3.1.

Дополнительно, на устройстве управления к блоку обработки информации подключается внешний EEPROM, расширитель цифровых портов и преобразователь интерфейса USB в UART.

Внешний EEPROM предназначен для хранения данных во внешней памяти, которая может быть доступна микроконтроллеру или другому устройству через определённый интерфейс [51]. В реализуемом комплексе внешняя память подключается к микроконтроллеру посредством I2C интерфейса, описанного в пункте 3.3.1.

Функциональные возможности внешней EEPROM включают следующее:

1. Чтение и запись данных. Подключённый микроконтроллер может записывать данные в память и читать их из неё. Это позволяет сохранять данные между сеансами работы устройства.

2. Безопасное хранение данных. Внешняя память обычно имеет механизмы защиты данных, такие как контроль чётности или аппаратную защиту от записи, которые помогают защитить хранимые данные от ошибок и несанкционированного доступа.

3. Устойчивость к воздействию окружающей среды. EEPROM обычно имеет улучшенную устойчивость к температурным и электромагнитным воздействиям.

4. Большой объем памяти. По сравнению с встроенной в микроконтроллер EEPROM, внешняя память имеет достаточный объем для хранения большого количества данных.

Устанавливаемый на устройство управления расширитель портов MCP23017 взаимодействует с микроконтроллером по I2C интерфейсу и позволяет управлять до 16 цифровыми входами/выходами (I/O) с помощью всего двух выводов микроконтроллера. Главная функция, которую выполняет расширитель портов – увеличение количества контактов ATmega328P-AU.

MCP23017 позволяет управлять каждым входом отдельно, задавая необходимую конфигурацию, а также генерировать прерывания при изменении состояния входов, что позволяет микроконтроллеру быстро реагировать на изменения [52]. Блок-схема MCP23017 представлена на рисунке 3.8.

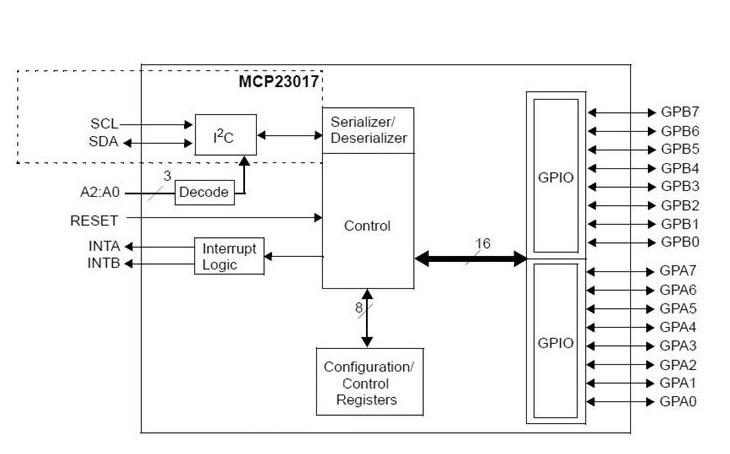


Рисунок 3.8 – Блок-схема расширителя портов MCP23017

Преобразователь интерфейса USB в UART устанавливается на устройство управления и представляет собой микросхему CH340G – универсальный USB-конвертер последовательного интерфейса (UART), который позволяет передавать данные между компьютером и другими электронными устройствами через USB-порт [53].

Микросхема CH340G связана с микроконтроллером и с её помощью будет осуществляться программирование устройства управления через установленный USB-порт, данные которого преобразуются и передаются по UART интерфейсу, с которым работает ATmega328P-AU. Также микросхема имеет встроенный регулятор напряжения, который позволяет питать компоненты устройства от USB-порта.

**3.2.2 Блок управления летательным аппаратом**

Блок управления летательным аппаратом состоит из двух джойстиков, которые позволяют пользователю управлять полётом БПЛА. Один джойстик отвечает за скорость вращения двигателей устройства. Другой джойстик имеет две оси: ось тангажа и ось крена. Они предназначены для управления движением БПЛА вперёд/назад и влево/вправо соответственно.

Все джойстики подключаются напрямую к аналоговым входам микроконтроллера и питаются от напряжения 5 В.

### **3.2.3 Блок взаимодействия с настройками управления**

Данный блок состоит из 8 тактовых кнопок, двух микротумблеров на 2 позиции и двух микротумблеров на 3 позиции. Кнопки подключаются к одному аналоговому входу микроконтроллера и, в зависимости от поступающего сигнала, микроконтроллер определяет, какая кнопка была нажата/отжата. Все микротумблеры связываются с микроконтроллером через подключение к расширителю портов, описанному в пункте 3.2.1.

Задача рассматриваемого блока – обеспечение возможности пользователю физически изменять или сохранять настройки разрабатываемого устройства. Каждая кнопка будет отвечать за определённое действие, как, например: смещение центра джойстика по крену на 1, смещение центра джойстика по тангажу на 1, изменение диапазона скорости, сохранение настроек.

### **3.2.4 Блок приёма и передачи радиосигнала устройства управления**

Блок приёма и передачи радиосигнала устройства управления представлен в виде радиомодуля и имеет такой же функционал, что и блок приёма и передачи радиосигнала контроллера полёта, описанный в пункте 3.1.4.

С точки зрения передачи радиосигнала, данный блок реализует ведущее устройство.

### **3.2.5 Блок отображения информации**

Блок отображения информации устанавливается на устройство управления и необходим для вывода данных пользователю. Рассматриваемый блок состоит из графического жидкокристаллического дисплея с разрешением 160 x 128 пикселей и диагональю 1.77 дюйма, использующий контроллер ST7735S. Он обладает множеством функциональных возможностей, таких как:

1. Отображение изображений, графики текста и символов.

2. Поддержка 18-битного цвета, что позволяет отображать до 262144 оттенков цвета.

3. Режим монохромного отображения.

4. Наличие встроенного контроллера с памятью для хранения изображений и графики.

Принцип работы модуля LCD 160x128 1.77 ST7735S заключается в следующем:

1. Контроллер микросхемы ST7735S получает информацию о том, какие пиксели нужно отобразить на дисплее, и генерирует соответствующие сигналы для управления дисплеем.

2. Контроллер отправляет данные о цвете пикселей и их координатах на дисплейный модуль.

3. Каждый пиксель состоит из трёх цветовых составляющих – красной, зелёной и синей (RGB). Контроллер дисплея сочетает эти цвета в нужной пропорции, чтобы создать желаемый цвет пикселя.

4. Когда все пиксели были обновлены на дисплее, контроллер дисплея генерирует сигнал окончания вывода и переходит в режим ожидания новых данных.

Питается модуль от напряжения 3.3 В и подключается к микроконтроллеру посредством SPI интерфейса [54].

### **3.2.6 Блок питания устройства управления**

Данный функциональный блок обеспечивает питанием модули и элементы, установленные на устройство управления. Блок состоит из аккумулятора, который выдаёт напряжение 3.7 В. Так как большинство элементов работают от напряжения 3.3 В и 5 В, то питание аккумулятора подключается к повышающему DC-DC преобразователю до 5 В. Напряжение, получаемое от преобразователя, поступает на стабилизаторы напряжения на 3.3 В и 5 В. Элементы схемы питаются уже от стабилизаторов. Дополнительно устанавливается диод Шоттки, который контролирует полярность подключения питания.

В блоке также предусмотрена возможность зарядки аккумулятора устройства управления. Эту функцию выполняет подключаемая к аккумулятору микросхема TP4056 – контроллер заряда Li-ion аккумуляторов [55]. Принцип её работы состоит в следующем:

1. Контроллер заряда измеряет напряжение на аккумуляторе и определяет, нужно заряжать аккумулятор или нет.

2. Если аккумулятор необходимо зарядить, контроллер заряда настраивает ток зарядки. Это делается путём установки напряжения на выходе регулировки тока зарядки.

3. Контроллер заряда следит за процессом зарядки и переключает зарядное устройство с постоянного тока на пульсирующий ток, когда аккумулятор приближается к полной зарядке. Это позволяет зарядить аккумулятор полностью и увеличить его срок службы.

4. Контроллер заряда обеспечивает защиту от перегрузки и короткого замыкания, чтобы предотвратить повреждение аккумулятора или других компонентов, подключённых к нему.

Схема контроллера заряда представлена на рисунке 3.9.

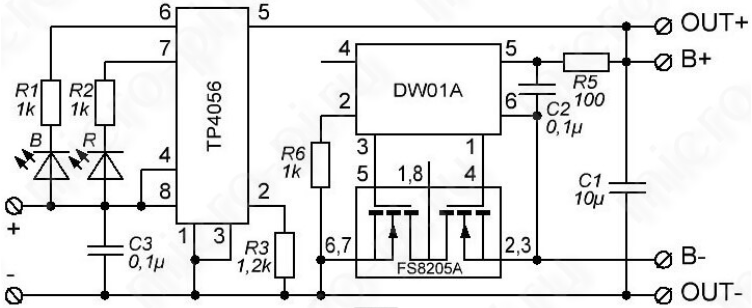


Рисунок 3.9 – Схема контроллера заряда TP4056

**3.3 Интерфейсы и протоколы**

Данный пункт описывает, как различные компоненты и устройства общаются друг с другом. Будут рассмотрены интерфейсы и протоколы, которые используются в разрабатываемом комплексе. Помимо этого, также будет рассмотрена широко-импульсная модуляция и GFSK модуляция.

### **3.3.1 Интерфейс I2C**

Интерфейс I2C представляет собой последовательную асимметричную шину для связи между интегральными схемами внутри электронных приборов, которая использует две двунаправленные линии связи: SDA (Data) и SCL (Clock).

SDA – это линия передачи данных, которая используется для передачи битов данных в обоих направлениях между микроконтроллером и устройством [56]. SCL – это линия синхронизации тактовых импульсов, которая контролирует скорость передачи данных и синхронизирует передачу данных между устройствами.

Диаграмма передачи данных по интерфейсу I2C представлена на рисунке 3.10.

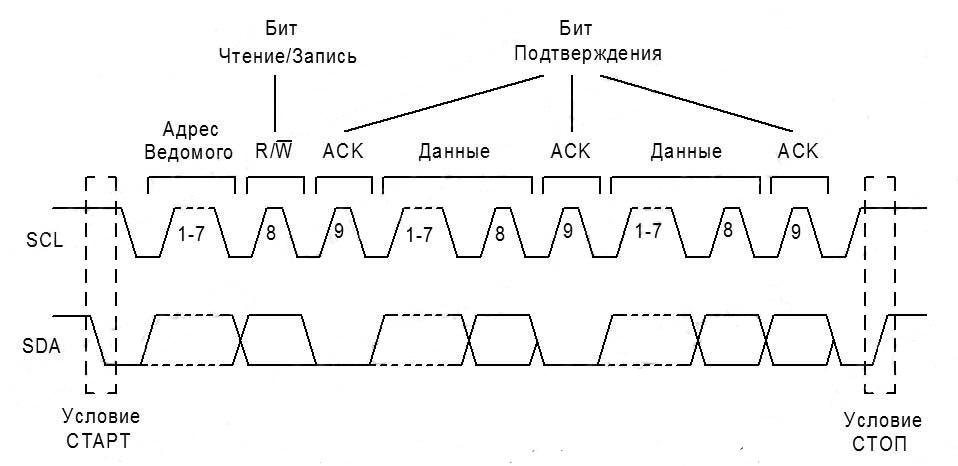


Рисунок 3.10 – Диаграмма передачи данных по интерфейсу I2C

Всего на одной шине I2C может быть до 127 устройств. Каждое устройство распознается по уникальному адресу и может работать как передатчик или приёмник, в зависимости от назначения. Кроме того, при передаче данных, устройства могут быть классифицированы как ведущие и ведомые. Ведущий – это устройство, которое инициирует передачу данных и вырабатывает сигналы синхронизации. При этом любое адресуемое устройство считается ведомым по отношению к ведущему. Исходя из спецификации работы шины, в каждый отдельный момент в шине может быть только один ведущий, а именно то устройство, которое обеспечивает формирование сигнала SCL шины. Ведущий может выступать как в роли ведущего-передатчика, так и ведущего-приёмника.

Процедура обмена начинается с того, что ведущий формирует состояние «СТАРТ» – генерация перехода сигнала линии SDA из высокого состояния в низкое при высоком уровне на линии SCL. Этот переход воспринимается устройствами, подключёнными к шине, как признак начала процедуры обмена. Процедура обмена завершается тем, что ведущий формирует состояние «СТОП» – переход состояния линии SDA из низкого состояния в высокое при высоком уровне на линии SCL. Состояния всегда вырабатываются ведущим. Считается, что шина занята после фиксации состояния «СТАРТ». Шина считается освободившейся через некоторое время после фиксации состояния «СТОП».

Данные по шине I2C передаются 8-битными словами. После каждого слова передаётся один бит подтверждения приёма принимающей стороной, выставляемый на шину SDA.

Изначально шиной поддерживался 7-битный адрес и скорость передачи в 100 кбит/с. Позже спецификация была дополнена и теперь возможна 10-битная адресация и скорость до 400 кбит/с.

### **3.3.2 Интерфейс SPI**

SPI – последовательный синхронный интерфейс для передачи данных в режиме полного дуплекса, предназначенный для обеспечения простого и недорогого высокоскоростного сопряжения микроконтроллеров и периферии.

В SPI используются четыре цифровых сигнала:

1. MOSI – выход ведущего, вход ведомого. Служит для передачи данных от ведущего устройства ведомому.

2. MISO – вход ведущего, выход ведомого. Служит для передачи данных от ведомого устройства ведущему.

3. SCLK или SCK – последовательный тактовый сигнал. Служит для передачи тактового сигнала для ведомых устройств.

4. CS или SS – выбор ведомого устройства (сигнал является инверсным, активный уровень – низкий).

Передача осуществляется пакетами. Длина пакета, как правило, составляет 1 байт (8 бит), при этом известны реализации SPI с иной длиной пакета, например, 4 бита. Ведущее устройство инициирует цикл связи установкой низкого уровня на выводе выбора подчинённого устройства (SS), с которым необходимо установить соединение [57].

В простейшем случае к ведущему устройству подключено единственное ведомое устройство и необходим двусторонний обмен данными. В таком случае используется трёхпроводная схема подключения. Интерфейс SPI позволяет подключать к одному ведущему устройству несколько ведомых устройств, причём подключение может быть осуществлено следующими способами: структура связи «звезда», структура связи «кольцо». На рисунке 3.11 представлены схемы подключения нескольких ведомых устройств.

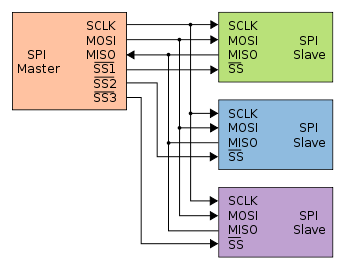
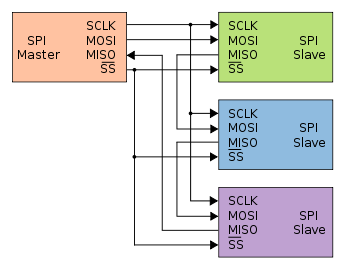
*а)**б)*

Рисунок 3.11 – Схема подключения нескольких ведомых устройств по шине SPI: *а* – структура связи «звезда», *б* – структура связи «кольцо»

В процессе SPI передачи используется классический способ преобразования данных из параллельной формы в последовательную с помощью регистра сдвига (см. рисунок 3.12).

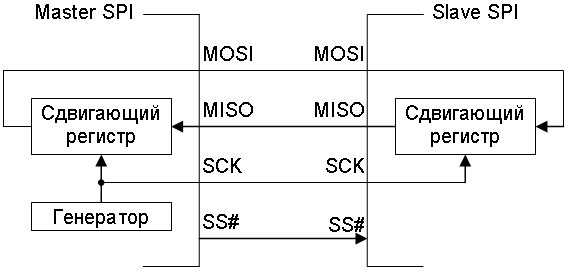


Рисунок 3.12 – Формат передачи данных по интерфейсу SPI

Сначала по шине данных или из некоторого буферного регистра производится параллельная загрузка данных в сдвигающий регистр. Далее, по тактовому сигналу SCLK, формируемому ведущим устройством SPI, выполняется побитовый сдвиг данных в сдвигающем регистре. Данные, выдвигаемые с одной стороны регистра, формируют выходной сигнал устройства; входные данные устройства побитно вводятся с другой стороны регистра. Это одновременно происходит как в ведущем, так и в ведомом устройстве. Тактовый сигнал SCLK, который формируется ведущим устройством, является общим для ведущего и ведомого и синхронизирует сдвиг данных в сдвигающих регистрах и фиксацию информации на двух линиях данных (MOSI – от ведущего, MISO – от ведомого).

### **3.3.3 Интерфейс UART**

Интерфейс UART (Universal Asynchronous Receiver/Transmitter) является одним из наиболее распространённых интерфейсов для связи между устройствами в электронике. Он представляет собой асинхронный последовательный интерфейс, который может использоваться для передачи данных в двух направлениях [58].

UART состоит из двух линий связи: линии передачи (TX) и линии приёма (RX). TX линия передаёт данные в виде последовательности битов, а RX линия принимает эти биты и декодирует их обратно в исходные данные.

Передача данных в UART осуществляется по одному биту в равные промежутки времени. Этот временной промежуток определяется заданной скоростью UART и для конкретного соединения указывается в бодах (что в данном случае соответствует битам в секунду). Существует общепринятый ряд стандартных скоростей: 300; 600; 1200; 2400; 4800; 9600; 19200; 38400; 57600; 115200; 230400; 460800; 921600 бод.

Помимо информационных битов, UART автоматически вставляет в поток синхронизирующие метки, так называемые стартовый и стоповый биты (см. рисунок 3.13). При приёме эти лишние биты удаляются из потока. Обычно стартовый и стоповый биты обрамляют один байт информации (8 бит), при этом младший информационный бит передаётся первым, сразу после стартового. Для формирования временных интервалов передающий и приёмный UART имеют источник точного времени (тактирования).

Принято соглашение, что пассивным (в отсутствие потока данных) состоянием входа и выхода UART является логическая 1. Стартовый бит всегда является логическим 0, поэтому приёмник UART ждёт перепада из 1 в 0 и отсчитывает от него временной промежуток в половину длительности бита (середина передачи стартового бита). Если в этот момент на входе всё ещё 0, то запускается процесс приёма минимальной последовательности. Для этого приёмник отсчитывает 9 битовых длительностей подряд (для 8-битных данных), и в каждый момент фиксирует состояние входа. Первые 8 значений являются принятыми данными, последнее значение – проверочное (стоп-бит). Значение стоп-бита всегда равно 1. Если реально принятое значение иное, UART фиксирует ошибку.

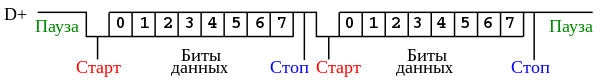


Рисунок 3.13 – Передача данных по интерфейсу UART

Поскольку синхронизирующие биты занимают часть битового потока, то результирующая пропускная способность UART меньше скорости соединения.

Многие реализации UART имеют возможность автоматически проверять целостность данных методом контроля битовой чётности. Включение или выключение этой функции производится записью соответствующего слова инициализации во внутренний регистр управления UART. Когда эта функция включена, минимальная последовательность информационных битов дополняется конечным битом контроля чётности.

### **3.3.4 Протокол радиопередачи ESB**

Протокол радиопередачи Enhanced ShockBurst (ESB) – это высокоэффективный и надёжный протокол беспроводной связи, который был разработан компанией Nordic Semiconductor для использования в радиочастотных системах на базе кристаллов и передатчиков-приёмников. Протокол поддерживает двустороннюю передачу пакетов данных, буферизацию пакетов, подтверждение пакетов и автоматическую повторную передачу потерянных пакетов [59].

Протокол ESB использует технологию частотного расширения спектра (FHSS) для обеспечения надёжной и безопасной передачи данных. FHSS разбивает данные на множество небольших пакетов и передаёт их по разным частотам, изменяя частоту передачи в каждом кадре данных. Это позволяет избежать помех от других устройств и снизить воздействие шума на передачу. Пакеты содержат в себе заголовок, полезную нагрузку и контрольную сумму. Заголовок пакета содержит информацию о типе пакета, идентификаторе передатчика и получателя, а также другую метаинформацию.

Пакеты данных передаются в режиме burst, в котором передатчик передаёт набор пакетов данных подряд, ожидая подтверждения получения каждого пакета. Получатель в свою очередь проверяет целостность принятых данных и посылает обратный сигнал (ACK) в ответ на каждый полученный пакет данных. Если передатчик не получает подтверждения получения пакета данных в течение заданного времени, то он повторно передаёт пакет до тех пор, пока не получит подтверждение. Пример повторной передачи пакетов представлен на рисунке 3.14.

Кроме того, протокол ESB обеспечивает безопасность передачи данных с помощью шифрования AES-128. Протокол ESB поддерживает как режим двунаправленной связи, так и односторонней передачи данных. В режиме двунаправленной связи устройства могут обмениваться данными и подтверждениями. В режиме односторонней передачи данные передаются только от отправителя к получателю без подтверждения получения.

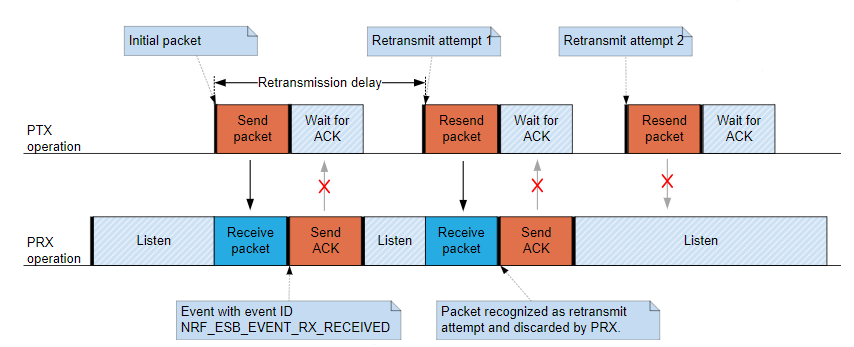


Рисунок 3.14 – Повторная передача пакетов в протоколе ESB

### **3.3.5 GFSK модуляция**

Гауссовская частотная модуляция (GFSK) – вид частотной модуляции, при которой используется фильтр Гаусса для сглаживания частотных перестроек при изменении значения информационного символа. Он основан на частотной модуляции FSK, но с использованием гауссовской формы импульса, которая обеспечивает более эффективное использование спектральной полосы [60].

Принцип работы GFSK заключается в том, что биты данных модулируют фазовый угол несущего сигнала. Если следующий бит данных отличается от предыдущего, то фазовый угол несущего сигнала меняется, что приводит к изменению частоты несущей. Однако вместо резких изменений частоты используется гауссовская форма импульса, что сглаживает изменения частоты и позволяет уменьшить ширину спектра передаваемого сигнала. Сигнал GFSK представлен на рисунке 3.15.

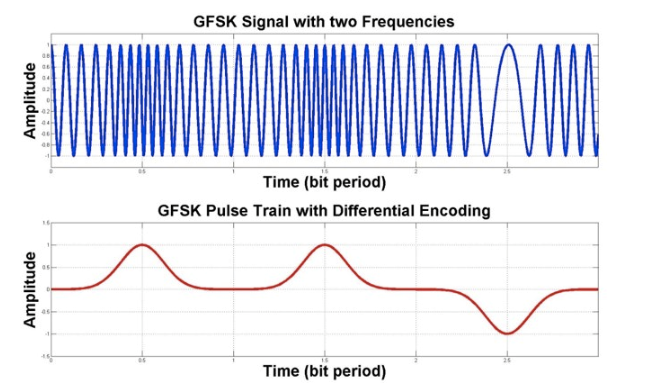


Рисунок 3.15 – Сигнал GFSK

GFSK обладает хорошей помехоустойчивостью и эффективно использует доступную полосу частот, что делает его широко применяемым в беспроводной связи, например, в Bluetooth и некоторых видах радиосвязи.

### **3.3.6 Широтно-импульсная модуляция**

Широтно-импульсная модуляция (ШИМ, англ. PWM) – это метод преобразования сигнала, при котором изменяется длительность импульса (скважность), а частота остаётся константой (см. рисунок 3.16). ШИМ часто используется в электронике для управления мощностью, например, для управления скоростью электрического двигателя или яркостью светодиодной лампы.

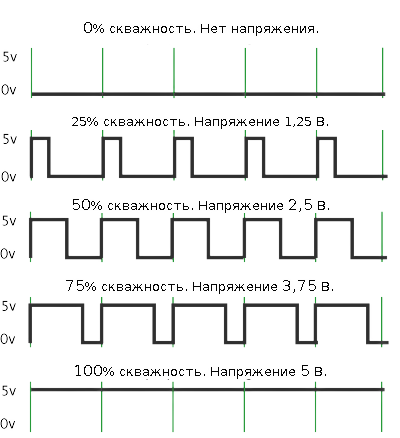


Рисунок 3.16 – Диаграмма ШИМ сигнала

Основной причиной применения ШИМ является стремление к повышению КПД [61]. В ШИМ в качестве ключевых элементов используют транзисторы (или другие полупроводниковые приборы), работающие не в линейном, а в ключевом режиме, то есть транзистор всё время либо разомкнут (выключен), либо замкнут (находится в состоянии насыщения). В первом случае транзистор имеет очень высокое сопротивление, поэтому ток в цепи весьма мал, и, хотя всё напряжение питания падает на транзисторе, выделяемая на транзисторе мощность очень мала. Во втором случае сопротивление транзистора крайне мало, и, следовательно, падение напряжения на нём близко к нулю, при этом выделяемая мощность так же мала. В переходных состояниях (переход ключа из проводящего состояния в непроводящее и обратно) мощность, выделяемая в ключе, значительна, но так как длительность переходных состояний крайне мала по отношению к периоду переключения, то средняя мощность потерь на переключение оказывается незначительной.

# **4 РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ СХЕМЫ**

В этом разделе подробно рассматривается практическая реализация функциональной схемы аппаратных средств устройства. Состав устройства показан на схеме электрической принципиальной (чертёж ГУИР.400201.022 Э3). В разделе описываются элементы представленной схемы, приведены программные фрагменты для взаимодействия с этими элементами, а также выполнены необходимые расчёты.

## **4.1 Блок обработки информации контроллера полёта**

Устанавливаемый на контроллер полёта микроконтроллер ATmega328P-AU представлен на принципиальной схеме микросхемой DD1. Графическое изображение контактов микроконтроллера из технической спецификации [9] представлено на рисунке 4.1.

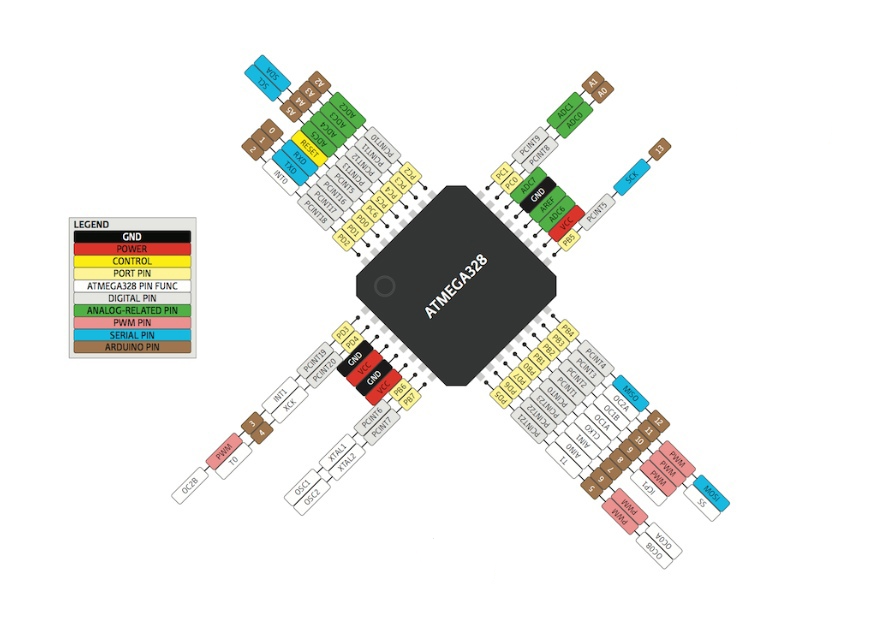


Рисунок 4.1 – Контакты микроконтроллера ATmega328P-AU

### **4.1.1 Подключение питания и внешнего резонатора микроконтроллера**

Для того, чтобы микроконтроллер работал, все его VCC входы (4, 6, 18) подключаются к источнику питания с напряжением +5 В, а все GND (3, 5, 21) – подключаются к общей земле. Помимо этого, к микроконтроллеру необходимо подключить внешний тактовый резонатор, как показано на рисунке 4.2. Тактовый резонатор используется для предоставления точного и стабильного источника тактового сигнала, который управляет внутренними операциями микроконтроллера.

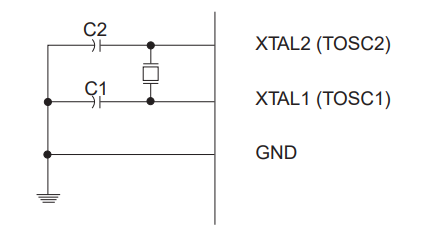


Рисунок 4.2 – Подключение внешнего тактового резонатора к микроконтроллеру ATmega328P-AU

В качестве тактового резонатора используется кварцевый SMD резонатор на 16.0 МГц, который подключается к RB6 и PB7 выходу микроконтроллера. К резонатору дополнительно подключаются С1 и С2 конденсаторы на 22 пФ. Информация о подключаемом резонаторе и конденсаторах взята из технической спецификации [9].

### **4.1.2 Подключение порта для программирования микроконтроллера**

Для программирования ATmega328P-AU необходимо подключить к нему программатор. Программатор – это устройство или инструмент, используемый для записи программного обеспечения в электронные устройства, такие как микроконтроллеры, микросхемы, процессоры и другие программно-управляемые устройства. Программаторы могут также использоваться для считывания программного обеспечения из этих устройств.

Для программирования контроллера полёта будет использоваться внешний программатор CP2102 [62]. CP2102 подключается к персональному компьютеру через USB порт и позволяет обмениваться информацией с микроконтроллером через интерфейс UART, с которым работает ATmega328P-AU. Для этого на плате полётного контроллера предусмотрены выходы для UART интерфейса, которые представлены на принципиальной схеме разъёмом XP2.

Разъём XP2 подключается следующим образом:

− +5V – шина +5 В устройства;

− VCC – общая земля;

− RX – выход PD1 микроконтроллера;

− TX – выход PD0 микроконтроллера;

− Reset – выход PC6 микроконтроллера.

### **4.1.3 Подключение pull-up резисторов**

При подключении устройств к сигнальным линиям SDA, SCL, а также линии Reset могут возникать проблемы, такие как нежелательные переходы уровней сигнала, интерференция и шум. Резистор на сигнальной линии помогает устранить эти проблемы. Такой резистор принято называть pull-up резистором, его номинал, согласно технической спецификации Atmega328P-AU, составляет 10 кОм. Он подключается к линии и к источнику питания +5 В.

Pull-up резисторы представлены на принципиальной схеме элементами R22, R23, R24.

## **4.2 Блок определения ориентации в пространстве**

В состав рассматриваемого блока входит микросхема DD3 – модуль GY-521. Контакты модуля описаны в таблице 4.1.

Таблица 4.1 – Описание контактов модуля GY-521

|  |  |  |
| --- | --- | --- |
| **Выход** | **Название** | **Описание** |
| 1 | VCC | Питание модуля, напряжение 5 В. |
| 2 | GND | Земля. |
| 3 | SCL | Тактовый, синхронизирующий сигнал шины I2C. |
| 4 | SDA | Линия данных I2C. |
| 5 | XDA | Дополнительный I2C интерфейс для подключения внешнего магнитометра. |
| 6 | XCL |
| 7 | AD0 | I2C адрес модуля. По умолчанию AD0 подтянут к земле, поэтому адрес устройства – 0x68. Если соединить AD0 с контактом питания, то адрес изменится на 0x69. |
| 8 | INT | Настраиваемое прерывание. |

### **4.2.1 Подключение модуля GY-521**

Подключение модуля GY-521 к микроконтроллеру производится по шине I2C и не требует дополнительных компонентов. На основе таблицы 4.1 подключение к микроконтроллеру DD1 выполняется следующим образом:

− VCC – шина +5 В устройства;

− GND – общая земля;

− SCL – выход PC5 микроконтроллера;

− SDA – выход PC4 микроконтроллера;

− XDA – не подключается (отсутствует внешний магнитометр);

− XCL – не подключается (отсутствует внешний магнитометр);

− AD0 – не подключается (адрес модуля 0x68);

− INT – выход PD2 микроконтроллера.

### **4.2.2 Программное управление модулем GY-521**

Так как работа с модулем GY-521 осуществляется посредствам шины I2C, то для работы необходимо подключить и инициализировать встроенную библиотеку Wire.h, как показано ниже:

#include <Wire.h> //подключение библиотеки для работы с I2C

void setup() {

…

Wire.begin(); //инициализация и подключение к шине I2C

…

}

Для определения адреса I2C, по которому подключён модуль GY-521, реализован следующий метод:

//метод проверяет подключение модуля GY-521 по указанному адресу

byte search\_gy521(int address) {

//начать передачу данных I2C по указанному адресу

Wire.beginTransmission(address);

Wire.write(0x75); //

Wire.endTransmission(); //завершить передачу

//запрашиваем данные у модуля по адресу

Wire.requestFrom(address, 1);

timer = millis() + 100; //устанавливаем таймер

//ждем пока не получим данные от модуля

while (Wire.available() < 1 && timer > millis());

lowByte = Wire.read(); //читаем старший байт данных

return lowByte;

}

Модуль GY-521 может иметь адрес 0x68 или адрес 0x69. Для того, чтобы проверить какой из этих адресов используется, выполняется вызов вышеописанного метода:

Serial.println(F("Searching for MPU-6050 on address 0x68"));

delay(1000);

if (search\_gy521(0x68) == 0x68) { //

Serial.println(F("MPU-6050 found on address 0x68"));

gy521\_address = 0x68;

}

if (gy521\_address == 0) {

Serial.println(F("Searching for MPU-6050 on address 0x69"));

delay(1000);

if (search\_gy521(0x69, 0x75) == 0x68) {

Serial.println(F("MPU-6050 found on address 0x69"));

gy521\_address = 0x69;

}

}

//если удалось определить адрес – сохраняем в памяти

if (gy521\_address != 0) {

EEPROM.write(5, gy521\_address);

}

Для конфигурации модуля реализован метод setup\_gy521(). Метод задаёт все необходимые регистры и проверяет их правильность.

void setup\_gy521() {

//установка регистра PWR\_MGMT\_1 для активации модуля

Wire.beginTransmission(gy521\_address);

Wire.write(0x6B);

Wire.write(0x00);

Wire.endTransmission();

//установка регистра GYRO\_CONFIG для диапазона 500dps

Wire.beginTransmission(gy521\_address);

Wire.write(0x1B);

Wire.write(0x08);

Wire.endTransmission();

//установка регистра ACCEL\_CONFIG для точности +/- 8g

Wire.beginTransmission(gy521\_address);

Wire.write(0x1C);

Wire.write(0x10);

Wire.endTransmission();

//проверка правильности установки регистра GYRO\_CONFIG

Wire.beginTransmission(gy521\_address);

Wire.write(0x1B);

Wire.endTransmission();

Wire.requestFrom(gy521\_address, 1);

while (Wire.available() < 1);

if (Wire.read() != 0x08) {

digitalWrite(RED\_LED\_PIN, HIGH);

while (1) delay(10);

}

//установка регистра CONFIG для н.ч. фильтра на 43 Гц

Wire.beginTransmission(gy521\_address);

Wire.write(0x1A);

Wire.write(0x03);

Wire.endTransmission();

}

После конфигурации модуля GY-521, его необходимо откалибровать. Для этого выполняется цикл, внутри которого происходит получение данных с модуля. Цикл выполняется 2000 раз и после окончания высчитывается среднее отклонение. Важно при калибровке поместить устройство на ровную поверхность и обеспечить неподвижность. Ниже представлен код, иллюстрирующий калибровку.

Serial.println(F("Calibrating GY-521, this will take +/- 8 seconds"));

Serial.print(F("Please wait"));

for (cal\_int = 0; cal\_int < 2000; cal\_int++) {

if (cal\_int % 100 == 0) Serial.print(F("."));

read\_gy521();

gy521\_roll\_cal += acc\_axis[1];

gy521\_pitch\_cal += acc\_axis[2];

gy521\_yaw\_cal += acc\_axis[3];

delay(4);

}

gy521\_roll\_cal /= 2000;

gy521\_pitch\_cal /= 2000;

gy521\_yaw\_cal /= 2000;

//Вывод результатов калибровки

Serial.println(F(""));

Serial.print(F("Axis 1 offset="));

Serial.println(gy521\_roll\_cal);

Serial.print(F("Axis 2 offset="));

Serial.println(gy521\_pitch\_cal);

Serial.print(F("Axis 3 offset="));

Serial.println(gy521\_yaw\_cal);

Serial.println(F(""));

//сохраняем результаты калибровки в памяти устройства

EEPROM.write(1, gy521\_roll\_cal);

EEPROM.write(2, gy521\_pitch\_cal);

EEPROM.write(3, gy521\_yaw\_cal) ;

При получении данных с модуля, необходимо обратиться к регистру 0x3B и прочитать 14 байт информации в формате двухбайтовых чисел со знаком (значения в диапазоне от -32768 до 32767). Первые 8 байт хранят в себе информацию с акселерометра, следующие 2 байта информацию о температуре и последние 8 – информацию с гироскопа. Для чтения данных модуля используется метод, описанный ниже.

//чтение данных с модуля GY-521

void read\_ gy521(){

Wire.beginTransmission(gy521\_address);

Wire.write(0x3B); //отправляем запрос на получение данных

Wire.endTransmission();

Wire.requestFrom(gy521\_address, 14); //читаем 14 байт

//ждем, пока все 14 байт не будут прочитаны

while(Wire.available() < 14);

acc\_axis[1] = Wire.read()<<8|Wire.read();

acc\_axis[2] = Wire.read()<<8|Wire.read();

acc\_axis[3] = Wire.read()<<8|Wire.read();

temperature = Wire.read()<<8|Wire.read();

gyro\_axis[1] = Wire.read()<<8|Wire.read();

gyro\_axis[2] = Wire.read()<<8|Wire.read();

gyro\_axis[3] = Wire.read()<<8|Wire.read();

}

## **4.3 Блок определения высоты**

Блок определения высоты представлен в виде микросхемы DD4 – модуль GY-68. Контакты модуля описаны в таблице 4.2.

Таблица 4.2 – Описание контактов модуля GY-68

|  |  |  |
| --- | --- | --- |
| **Выход** | **Название** | **Описание** |
| 1 | VIN | Питание модуля, напряжение 5 В. |
| 2 | GND | Земля. |
| 3 | SCL | Тактовый, синхронизирующий сигнал шины I2C. |
| 4 | SDA | Линия данных I2C. |

### **4.3.1 Подключение модуля GY-68**

Подключение модуля GY-68 к микроконтроллеру производится по шине I2C и не требует дополнительных компонентов. На основе таблицы 4.2 подключение выполняется следующим образом:

− VCC – шина +5 В устройства;

− GND – общая земля;

− SCL – выход PC5 микроконтроллера DD1;

− SDA – выход PC4 микроконтроллера DD1.

### **4.3.2 Программное управление модулем GY-68**

Для работы с модулем GY-68 используется встроенная библиотека Wire.h и дополнительно библиотека Adafruit\_BMP085.h [72], с помощью которой можно без особых усилий инициализировать модуль и прочитать из него данные. Ниже приведены фрагменты кода, которые позволяют управлять модулем.

Adafruit\_BMP085 bmp; //создание объекта для работы с модулем

//инициализация модуля GY-68

void setup\_gy68() {

//поиск модуля по адресу 0x77 и его инициализация

if (!bmp.begin()) {

digitalWrite(RED\_LED\_PIN, HIGH);

while (1) delay(10);

}

}

//чтение данных с модуля GY-68

void read\_gy68() {

bmp.readTemperature(); //получение текущей температуры

bmp.readPressure(); //получение текущего давления

bmp.readAltitude(); //получение текущей высоты

}

## **4.4 Блок приёма и передачи радиосигнала контроллера полёта**

Блок представлен микросхемой DD5 – радиомодуль NRF24L01. Работа с модулем осуществляется по шине SPI, описание контактов модуля представлено в таблице 4.3.

Таблица 4.3 – Описание контактов модуля NRF24L01

|  |  |  |
| --- | --- | --- |
| **Выход** | **Название** | **Описание** |
| 1 | VIN | Питание модуля, напряжение 3.3 В. |
| 2 | GND | Земля. |
| 3 | CE | Разрешение передачи данных. |
| 4 | CSN | Выбор SPI устройства. |
| 5 | SCK | Тактовый, синхронизирующий сигнал шины SPI. |
| 6 | MOSI | Линия передачи данных SPI. |
| 7 | MISO | Линия приёма данных SPI. |
| 8 | IRQ | Настраиваемое прерывание. |

### **4.4.1 Подключение модуля NRF24L01**

Согласно технической спецификации [63], при подключении питания к модулю, необходимо между контактами GND и VCC установить конденсатор ёмкостью 100 мкФ. Конденсатор необходим для снижения помех. На принципиальной схеме конденсатор представлен элементом С14.

Согласно таблице 4.3, подключение NRF24L01 к микроконтроллеру DD1 выполняется следующим образом:

− VIN – шина +3.3 В устройства;

− VCC – общая земля;

− CE – выход PD7 микроконтроллера;

− CSN – выход PB0 микроконтроллера;

− SCK – выход PB5 микроконтроллера;

− MOSI – выход PB3 микроконтроллера;

− MISO – выход PB4 микроконтроллера;

− IRQ – не подключается (выход на землю).

### **4.4.2 Программное управление модулем NRF24L01**

Для взаимодействия с модулем NRF24L01 используется две библиотеки: nRF24L01.h и RF24.h [69].

В начале программы необходимо объявить объект модуля, адрес канала, по которому будет осуществляться передача радиосигнала и структуру для передачи. Важно, что адрес канала и структура должны быть одинаковыми для летательного аппарата и устройства управления. Ниже представлен фрагмент кода, описывающий объявление:

static const uint8\_t PROGMEM CE\_NRF = 7;

static const uint8\_t PROGMEM CS\_NRF = 8;

RF24 radio(CE\_NRF, CS\_NRF); //объект NRF (необходимо указать номер контакта микроконтроллера для CE и CS

static const uint64\_t PIPE = 0xE8E8F0F0E1LL; //канал

//структура передачи данных

struct MyData {

uint16\_t throttle;

uint16\_t yaw;

uint16\_t pitch;

uint16\_t roll;

uint16\_t aux1;

uint16\_t aux2;

};

MyData data; //объект структуры

Для начала обмена информацией по радиоканалу, необходимо провести конфигурацию модуля. Фрагмент кода, описывающий выполняемые настройки:

radio.setAutoAck(0); //отключение автоматического подтверждения

radio.setRetries(0, 15); //задаем одну попытку для передачи и ожидание в 3750 миллисекунд в случае неудачи

radio.setPayloadSize(sizeof(MyData)); //задаем размер данных

radio.setChannel(0x60); //выбираем 60й канал

//устанавливаем скорость передачи в 1 Мбит/сек

radio.setDataRate(RF24\_1MBPS);

//устанавливаем максимальную мощность передатчика

radio.setPALevel(RF24\_PA\_HIGH);

//задаем 1 канал для передачи по адресу PIPE

radio.openReadingPipe(1, PIPE);

//включение радиомодуля и инициализация настроек

radio.powerUp();

//переключаем модуль в режим приёма данных

radio.startListening();

После выполнения вышеописанных действий, модуль готов к приёму информации по радиоканалу. Для того, чтобы прочитать данные, выполняется следующий код:

//проверяем, что есть данные для чтения

if (radio.available()) {

//читаем полученные данные

radio.read(&data, sizeof(MyData));

throttle = map(data.throttle, 0, 258, 1000, 2000);

yaw = map(data.yaw, 0, 258, -500, 500);

pitch = map(data.pitch, 0, 258, -500, 500);

roll = map(data.pitch, 0, 258, -500, 500);

}

## **4.5 Блок индикации**

Блок индикации представлен двумя светодиодами VD1 и VD3, подключаемых анодом (+) к выходам микроконтроллера PD3 и PD4 соответственно. Также необходимо ограничить поступающий к светодиодам ток путём установки ограничивающих резисторов. Для этого необходимо рассчитать сопротивление устанавливаемых резисторов.

### **4.5.1 Расчёт номинала ограничивающих резисторов**

В проекте используются стандартные SMD светодиоды с рабочим напряжением 3.3 В и силой тока 20 мА. Источник питания – микроконтроллер, который выдаёт напряжение 5 В. Напряжение, которое должен взять на себя резистор рассчитывается по формуле (4.1).

По закону Ома, используя формулу (4.2), рассчитаем значение сопротивления.

Таким образом, получаем сопротивление ограничивающего резистора 85 Ом. Далее рассчитаем мощность, которую при этом резистору придётся рассеивать по формуле (4.3).

Исходя из расчётов получаем, что при мощности менее 34 мВт резистор перегорит. Чтобы продлить срок службы светодиодов, в проекте устанавливаются ограничивающие резисторы номиналом 100 Ом.

Существует два варианта подключения ограничивающего резистора к светодиоду, которые представлены на рисунке 4.3.

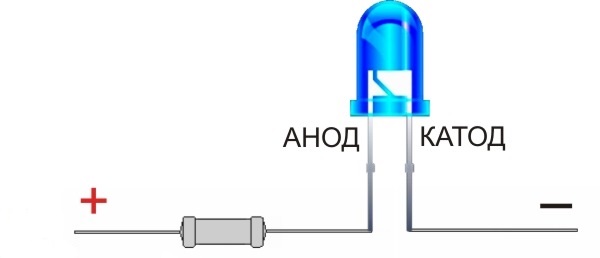
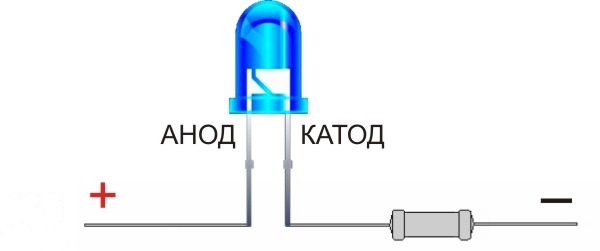
*а)**б)*

Рисунок 4.3 – Варианты подключения светодиода

Было принято решение подключить ограничивающие резисторы R6 и R7 к отрицательному контакту (-) светодиодов.

### **4.5.2 Программное управление светодиодами**

Для управления светодиодами в Arduino необходимо установить порты, к которым они подключены, на вывод, как показано ниже.

pinMode(LED\_PIN, OUTPUT); //LED\_PIN – номер выхода светодиода

Для того, чтобы включить/выключить светодиод используется функция digitalWrite(LED\_PIN, HIGH) и digitalWrite(LED\_PIN, LOW); соответственно.

## **4.6 Блок управления двигателями**

Двигатели представлены на принципиальной схеме элементами DD11-DD14, регуляторы оборотов двигателей – элементами DD7-DD10.

В двигателях трёхфазное питание, поэтому они имеют три выхода. Электродвигатель работает на принципе переменного тока, где магнитное поле вращается вокруг статора. Электрический ток поступает на каждую из трех фаз по очереди и обеспечивает фазовую разницу в электрическом сигнале, что позволяет магнитному полю вращаться. Графическое изображение выходов двигателей представлено на рисунке 4.4.

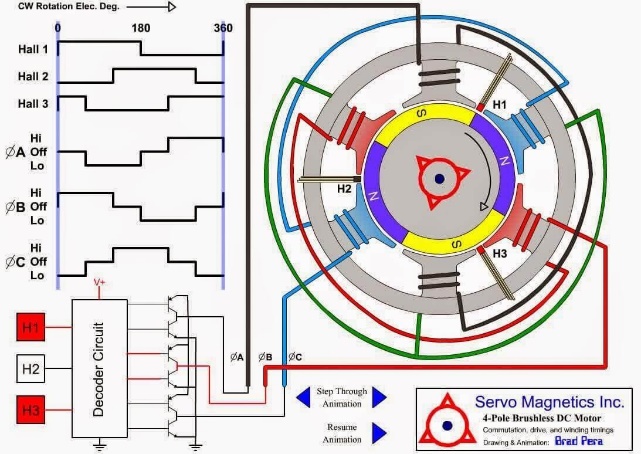


Рисунок 4.4 – Выходы двигателей

### **4.6.1 Подключение регуляторов оборотов и двигателей**

Регуляторы оборотов подключаются напрямую к двигателям, используя 3 выхода. Выход H1 регуляторов заводится на вход A двигателя, H2 – на вход B двигателя, H3 – на вход C двигателя.

Входы IN– всех ESC подключаются к общей земле, входы IN+ – к аккумуляторной батареи с напряжением питания 7.4 В. К DD1 микроконтроллеру регуляторы подключаются через вход A0: DD7 подключается к PB2 (10) выходу микроконтроллера, DD8 – к PD5 (5) выходу микроконтроллера, DD9 – к PD6 (6) выходу микроконтроллера, DD10 – к PB1 (9) выходу микроконтроллера.

Напряжение (+5 В) с выхода OUT+ одного из регуляторов (DD10) поступает в блок питания, где в последующем идёт на все элементы полётного контроллера. В других трёх регуляторах выход OUT+ никуда не подключается.

### **4.6.2 Программное управление двигателями**

Управление двигателями осуществляется через регистры микроконтроллера. Для конфигурирования контактов, которые подключены к ESC, используются следующие команды:

//Настройка выходов ESC как OUTPUT

DDRD |= B01000000; //контакт PD6(6)

DDRD |= B00100000; //контакт PD5(5)

DDRB |= B00000100; // контакт PB2(10)

DDRB |= B00000010; // контакт PB1(9)

ESC управляются с помощью ШИМ сигнала, поэтому для запуска двигателей необходимо подавать сигнал на протяжении от 1000 мс до 2000 мс. В этом случае 1000 мс означает, что двигатели будут полностью остановлены, 2000 мс – двигатели будут работать на полной мощности. Программно это реализуется следующим образом:

loop\_timer = micros(); //получаем текущее время

//начинаем подавать сигнал на ESC

PORTD |= B01000000; //подаем сигнал на ESC PD6(6)

PORTD |= B00100000; //подаем сигнал на ESC PD5(5)

PORTB |= B00000100; //подаем сигнал на ESC PB2(10)

PORTB |= B00000010; //подаем сигнал на ESC PB1(9)

//вычисление длительности сигнала каждого ESC

//esc – заданная скорость (от 1000 до 2000), loop\_timer – текущее время

timer\_channel\_1 = esc\_1 + loop\_timer;

timer\_channel\_2 = esc\_2 + loop\_timer;

timer\_channel\_3 = esc\_3 + loop\_timer;

timer\_channel\_4 = esc\_4 + loop\_timer;

esc\_loop\_timer = micros();

//перестаем подавать сигнал на ESC по истечении его времени

while (timer\_channel\_1 > esc\_loop\_timer || timer\_channel\_2 > esc\_loop\_timer || timer\_channel\_3 > esc\_loop\_timer || timer\_channel\_4 > esc\_loop\_timer) {

esc\_loop\_timer = micros();

if (timer\_channel\_1 <= esc\_loop\_timer) PORTB &= B11111011;

if (timer\_channel\_2 <= esc\_loop\_timer) PORTB &= B11111101;

if (timer\_channel\_3 <= esc\_loop\_timer) PORTD &= B11011111;

if (timer\_channel\_4 <= esc\_loop\_timer) PORTD &= B10111111;

}

## **4.7 Блок питания контроллера полёта**

Блок питания контроллера полёта включает в себя аккумулятор GB1, диод Шоттки VD5 и стабилизатор напряжения DA1.

В качестве аккумулятора используется батарея Li-Ion 803063 7.4 В, которая подключается к регуляторам оборотов двигателей DD7-DD10.

На контакте OUT+ одного из регуляторов устанавливается диод Шоттки SS14, выход которого подключается к шине питания +5 В контроллера полёта.

К шине +5 В подключается стабилизатор напряжения AMS1117 через вход IN. Вход GND стабилизатора подключается к общей земле. Выход OUT подключается к общей шине +3.3 В устройства.

## **4.8 Блок обработки информации устройства управления**

Устанавливаемый на устройство управления микроконтроллер ATmega328P-AU представлен на принципиальной схеме микросхемой DD2. Микроконтроллер питается от напряжения +5 В и к нему подключается тактовый резонатор ZQ2 и конденсаторы C3 и С4, схема подключения которых описана в пункте 4.1.1. Дополнительно к питанию микроконтроллера подключается два фильтрующих конденсатора C5 и C6 на 0.1 мкФ.

### **4.8.1 Подключение внешнего EEPROM**

Внешний EEPROM подключается к микроконтроллеру DD2 по I2C интерфейсу и представлен микросхемой DA3. Описание контактов микросхемы представлено в таблице 4.4.

Таблица 4.4 – Описание контактов микросхемы AT24C256C

|  |  |  |
| --- | --- | --- |
| **Выход** | **Название** | **Описание** |
| 1 | A0 | Адресный вход. Используется при подключении нескольких AT24C256C. |
| 2 | A1 |
| 3 | A2 |
| 4 | GND | Земля. |
| 5 | SDA | Линия данных I2C. |
| 6 | SCL | Тактовый, синхронизирующий сигнал шины I2C. |
| 7 | WP | Вход для защиты записи. Если подключен к земле – запись разрешена. |
| 8 | VCC | Питание модуля, напряжение 5 В. |

Исходя из таблицы 4.4, подключение микросхемы DA3 осуществляется следующим образом:

− A0 – не подключается (выход на землю);

− A1 – не подключается (выход на землю);

− A2 – не подключается (выход на землю);

− GND – общая земля;

− SDA – выход PC4 микроконтроллера DD2;

− SCL – выход PC5 микроконтроллера DD2;

− WP – общая земля;

− VCC – шина +5 В устройства.

Дополнительно к линиям SDA и SCL подключаются pull-up резисторы R18-R19 номиналом 4.7 кОм для снижения помех.

### **4.8.2 Подключение расширителя портов**

Расширитель портов MCP23017 представлен на принципиальной схеме микросхемой DD18. Описание контактов представлено в таблице 4.5.

Таблица 4.5 – Описание контактов микросхемы MCP23017

|  |  |  |
| --- | --- | --- |
| **Выход** | **Название** | **Описание** |
| 1 | A2 | Адресный вход. Используется при подключении нескольких MCP23017. |
| 2 | A1 |
| 3 | A0 |
| 4 | RES | Аппаратный сброс MCP23017. Активный низкий уровень. |
| 5 | SO | Линия данных SPI. |
| 6 | CS | Выбор SPI устройства. |
| 7 | SDA | Линия данных I2C. |
| 8 | SCL | Тактовый, синхронизирующий сигнал шины I2C. |
| 9, 12 | GND | Земля. |
| 10, 11 | VCC | Питание модуля, напряжение 5 В. |
| 13 | ITB | Настраиваемое прерывание контактов GPB. |
| 14 | ITA | Настраиваемое прерывание контактов GPA. |
| 16, 18, 20, 22, 24, 26, 28, 30 | GPA0-GPA7 | Цифровые входы/выходы для подключения дополнительных устройств. |
| 15, 17, 19, 21, 23, 25, 27, 29 | GPB0-GPB7 |

Подключение MCP23017 выполняется следующим образом:

− A0-A2 – не подключается (выход на землю);

− RES – не подключается (выход на +5 В для предотвращения срабатывания сброса). Дополнительно устанавливается резистор R28 на 10 кОм для ограничения тока и защиты от повреждений входа;

− SO, CS – не подключаются;

− SDA – выход PC4 микроконтроллера DD2;

− SCL – выход PC5 микроконтроллера DD2;

− GND – общая земля;

− VCC – шина +5 В устройства;

− ITB, ITA – не подключаются;

− GPB1, GPB2 – подключение к микротумблеру TB1 на 3 позиции;

− GPB4, GPB7 – подключение к микротумблеру TB2 на 3 позиции;

− GPB3 – подключение к микротумблеру TB3 на 2 позиции;

− GPB6 – подключение к микротумблеру TB3 на 2 позиции;

− GPA0-GPA6 – не подключаются;

− GPA7 – подключение к пьезодинамику.

− GPB0, GPB5 – не подключаются.

### **4.8.3 Подключение преобразователя интерфейса USB в UART**

Преобразователь интерфейса CH340G USB в UART представлен на принципиальной схеме микросхемой DA14, для подключения используется техническая спецификация [64]. Контакты микросхемы описаны в таблице 4.6.

Таблица 4.6 – Описание контактов микросхемы CH340G

|  |  |  |
| --- | --- | --- |
| **Выход** | **Название** | **Описание** |
| 1 | GND | Земля. |
| 2 | TXD | Выходной контакт, передающий данные от микросхемы CH340G к внешнему устройству. |
| 3 | RXD | Входной контакт, принимающий данные от внешнего устройства к микросхеме CH340G. |
| 4 | V3 | Внутреннее опорное напряжение для USB интерфейса. При питании 3,3 В вывод должен быть соединен с Vcc. При напряжении питания 5 В, к нему необходимо подключить относительно земли блокировочный конденсатор. |
| 5 | UD+ | Контакт USB D+ положительного питания. |
| 6 | UD- | Контакт USB D- отрицательного питания. |
| 7 | X1 | Входной контакт кварцевого резонатора или генератора сигналов (1 МГц – 12 МГц). |
| 8 | X0 | Выходной контакт кварцевого резонатора или генератора сигналов (1 МГц – 12 МГц). |
| 9 | VCC | Питание модуля, напряжение 5 В. |
| 10 | R232 | Включение инверсии входа RXD. Активный уровень – высокий. Вход имеет внутренний резистор, подключенный к земле. |
| 11 | RTS | Контакт UART запрос на передачу данных от CH340G. |
| 12 | DTR | Контакт UART готовности CH340G к обмену данными. |
| 13 | DCD | Контакт UART обнаружения подключения устройства. |
| 14 | RI | Оповещение приемника о наличии входящего сигнала прерывания на линии связи. |
| 15 | DSR | Готовность подключенного устройства к обмену данными. |
| 16 | CTS | Запрос подключенного устройства на передачу данных. |

На основе таблицы 4.6 и технической спецификации [64], подключение модуля выполняется следующим образом:

− GND – общая земля;

− TXD – выход PD0 микроконтроллера DD2;

− RXD – выход PD1 микроконтроллера DD2;

− V3 – не подключается;

− UD+ – выход D+ разъёма XP3;

− UD- – выход D- разъёма XP3;

− X1 – вход кварцевого резонатора ZQ3;

− X0 – выход кварцевого резонатора ZQ3;

− VCC – шина +5 В устройства;

− R232 – не подключается;

− RTS – не подключается;

− DTR – выход PC6 микроконтроллера DD2;

− DCD – не подключается;

− RI – не подключается;

− DSR – не подключается;

− CTS – не подключается.

К входам X0 и X1 микросхемы CH340G подключается кварцевый резонатор ZQ3, а также конденсаторы C18 и C19 ёмкостью 22 пФ согласно спецификации, как показано на рисунке 4.5. Дополнительно устанавливается резистор R29 сопротивлением 1 МОм для защиты от статического заряда и электромагнитных помех.

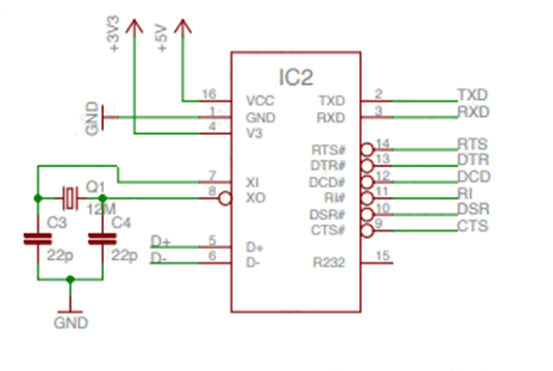


Рисунок 4.5 – Подключение кварцевого резонатора к CH340G

На шины питания заводятся фильтрующие конденсаторы C10-C12 ёмкостью 0.1 мкФ и конденсатор C5 ёмкостью 0.01 мкФ.

Согласно технической спецификации, на линии TXD и RXD необходимо установить резисторы сопротивлением 1 кОм или 2.2 кОм для защиты от перенапряжения в случае, если устройство и микросхема CH340G пытаются передавать данные одновременно. В проекте используются резисторы R12 и R13 сопротивлением 1 кОм. Для визуализации передачи данных по линиям TXD и RXD, к ним подключаются светодиоды VD2 и VD4, а также ограничивающие резисторы R11 и R10 для этих светодиодов.

На выход +5 В разъёма XP3 устанавливается самовосстанавливающийся предохранитель F1 номиналом 500 мА и конденсатор C17 ёмкостью 22 пФ для защиты от перегрузки, короткого замыкания и фильтрации электромагнитных помех.

### **4.8.4 Программное управление внешним EEPROM**

Для управления внешний EEPROM используется встроенная библиотека Wire.h. Для начала работы необходимо в setup() прописать инициализацию шины I2C, вызвав метод Wire.begin();

Для записи и чтения данных из внешней EERPOM используются методы, описанные ниже.

//метод позволяет записать данные в EEPROM. addr – адрес, по которому выполняется запись, val – данные для записи

void WriteEEPROM(int addr, uint8\_t val) {

//начало передачи данных. 0x50 – адрес EEPROM

Wire.beginTransmission(0x50);

Wire.write((int)(addr >> 8)); //запись старшего байта адреса

Wire.write((int)(addr & 0xFF)); //запись младшего байта адреса

Wire.write(val); //запись значения

Wire.endTransmission(); //завершение передачи

}

//метод позволяет прочитать данные из EEPROM. addr – адрес по которому выполняется чтение

uint8\_t ReadEEPROM(int addr) {

//начало передачи данных. 0x50 – адрес EEPROM

Wire.beginTransmission(0x50);

Wire.write((int)(addr >> 8)); //запись старшего байта адреса

Wire.write((int)(addr & 0xFF)); //запись младшего байта адреса

Wire.endTransmission(); //завершение передачи

Wire.requestFrom(0x50, 1); //запрос на байт информации

return Wire.read(); //чтение данных

}

### **4.8.5 Программное управление расширителем портов**

Для работы с MCP23017 используется встроенная библиотека Wire.h и подключаемая библиотека Adafruit\_MCP23017 [74].

В начале работы программы выполняется объявление объекта модуля и инициализация расширителя портов и используемых выходов, как показано ниже.

Adafruit\_MCP23017 mcp;

//инициализация модуля по стандартному адресу 0x20

mcp.begin();

//настройка INPUT входов

mcp.pinMode(15, INPUT); //настраиваем GPB4 как INPUT

mcp.pinMode(12, INPUT); //настраиваем GPB7 как INPUT

mcp.pinMode(11, INPUT); //настраиваем GPB3 как INPUT

mcp.pinMode(14, INPUT); //настраиваем GPB5 как INPUT

mcp.pinMode(9, INPUT); //настраиваем GPB2 как INPUT

mcp.pinMode(10, INPUT); //настраиваем GPB1 как INPUT

//настройка подтягивающих резисторов на входах

mcp.pullUp(15, HIGH);

mcp.pullUp(12, HIGH);

mcp.pullUp(11, HIGH);

mcp.pullUp(14, HIGH);

mcp.pullUp(9, HIGH);

mcp.pullUp(10, HIGH);

//настройка OUTPUT выходов

mcp.pinMode(7, OUTPUT); //настраиваем GPA7 как OUTPUT

Для чтения данных с заданного контакта используется метод mcp.digitalRead(PIN);, где PIN – номер контакта.

Для записи 0 и 1 на заданный контакт используется метод mcp.digitalWrite(), как показано ниже:

mcp.digitalWrite(PIN, LOW);

mcp.digitalWrite(PIN, HIGH);

## **4.9 Блок управления летательным аппаратом**

Блок состоит из двух джойстиков DD16 и DD17. Графическое изображение выходов джойстиков представлено на рисунке 4.6.



Рисунок 4.6 – Выходы джойстиков

### **4.9.1 Подключение джойстиков**

Подключение джойстиков к микроконтроллеру DD2 выполняется следующим образом:

− VCCY – шина +5 В устройства;

− Y – для DD16 – вход PC1 микроконтроллера, для DD17 – вход PC3 микроконтроллера;

− GNDY – общая земля;

− VCCX – шина +5 В устройства;

− X – для DD16 – вход PC0 микроконтроллера, для DD17 – вход PC2 микроконтроллера;

− GNDX – общая земля.

### **4.9.2 Программное управление джойстиками**

Для получения информации с джойстиков используются стандартные средства языка C AVR. Все контакты микроконтроллера, к которым подключены джойстики настраиваются на вход, как показано ниже:

pinMode(A0, INPUT); //выход X джойстика DD16

pinMode(A1, INPUT); //выход Y джойстика DD16

pinMode(A2, INPUT); //выход X джойстика DD17

pinMode(A3, INPUT); //выход Y джойстика DD17

Так как джойстики подключены к аналоговым входам микроконтроллера, то для чтения показаний используется метод analogRead(PIN);, где PIN – выход джойстика, с которого необходимо прочитать значение.

## **4.10 Блок взаимодействия с настройками управления**

Блок взаимодействия с настройками управления представлен на принципиальной схеме кнопками SW2-SW9.

### **4.10.1 Подключение кнопок**

Все кнопки подключаются к одному аналоговому входу ADC7 микроконтроллера DD2. Для подключения 8 кнопок к одному аналоговому входу реализуется делитель напряжения [65], как показано на рисунке 4.7.

Для реализации делителя напряжения, как показано на рисунке 4.7, подключаются резисторы R25, R7, R14, R15, R16, R17, R20, R21 и R26. Резистор R25 ёмкостью 10 кОм представляет собой pull-up резистор и необходим для уменьшения помех.

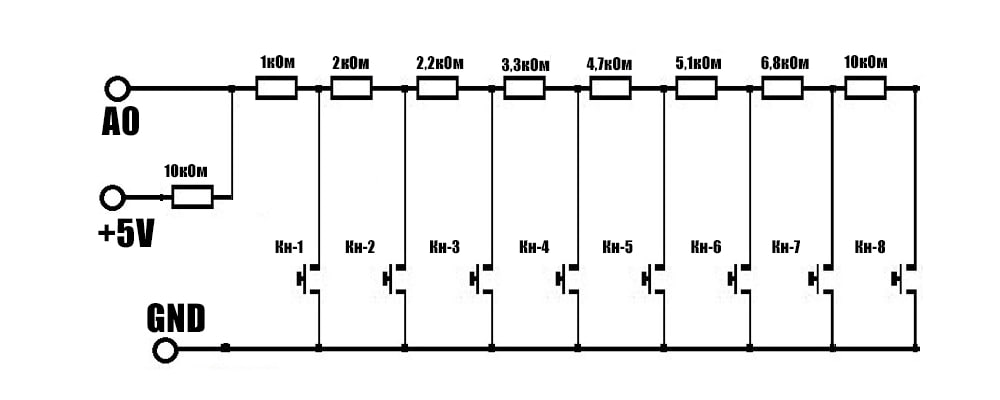


Рисунок 4.7 – Делитель напряжения для подключения кнопок

### **4.10.2 Программное управление кнопками**

Вход, к которому подключены кнопки, настраивается на вход микроконтроллера:

pinMode(A7, INPUT); //настраиваем ADC7 как INPUT

Для чтения значения аналогового сигнала используется следующий метод:

button\_read = analogRead(A7);

После получения значения с контакта, к которому подключены кнопки, определяется какая кнопка была нажата путем сравнения с диапазоном значений каждой кнопки:

if(button\_read < 120 && button\_read > 50) {

//нажата кнопка SW2

}

if(button\_read < 260 && button\_read > 200) {

//нажата кнопка SW3

}

if(button\_read < 380 && button\_read > 320) {

//нажата кнопка SW4

}

if(button\_read < 500 && button\_read > 430) {

//нажата кнопка SW5

}

if(button\_read < 550 && button\_read > 500) {

//нажата кнопка SW6

}

if(button\_read < 650 && button\_read > 580) {

//нажата кнопка SW7

}

if(button\_read < 730 && button\_read > 680) {

//нажата кнопка SW8

}

if(button\_read < 800 && button\_read > 750) {

//нажата кнопка SW9

}

## **4.11 Блок приёма и передачи радиосигнала устройства управления**

Блок представлен микросхемой DD6 – радиомодуль NRF24L01. Описание модуля представлено в пункте 4.4.

### **4.11.1 Подключение модуля NRF24L01**

Подключение NRF24L01 устройства управления к микроконтроллеру DD2 выполняется по аналогии с подключением радиомодуля контроллера полёта, описанного в пункте 4.4.1. Между контактами GND и VCC устанавливается конденсатор С15 ёмкостью 100 мкФ и дополнительно фильтрующий конденсатор С13 ёмкостью 0.1 мкФ.

### **4.11.2 Программное управления модулем NRF24L01**

В устройстве управления NRF24L01 является передатчиком информации. Инициализация и настройка модуля описана в пункте 4.4.2. Для того, чтобы настроить модуль на передачу информации, необходимо заменить конфигурацию radio.startListening(); на radio.stopListening();. Во всем остальном настройка идентична.

Для того, чтобы передать данные, используется метод radio.write(&data, sizeof(MyData));, где: data – объект данных, которые необходимо передать; sizeof(MyData) – размер передаваемых данных.

## **4.12 Блок отображения информации**

Блок отображения информации представлен в виде ЖК дисплея LCD 160x128 1.77 ST7735S – элемент DD15. Выходы дисплея представлены в таблице 4.7.

Таблица 4.7 – Описание выходов LCD 160x128 1.77 ST7735S

|  |  |  |  |
| --- | --- | --- | --- |
| **Выход** | **Название** | **Альтернативное название** | **Описание** |
| 1 | GND |  | Земля. |
| 2 | VCC | VIN | Питание дисплея, напряжение 5 В. |
| 3 | SCL | SCK | Тактовый, синхронизирующий сигнал. |
| 4 | SDA | MOSI, D0 | Данные. |
| 5 | RST | RESET | Аппаратный сброс дисплея. Активный низкий уровень. |
| 6 | DC | RS, A0 | Выбор передачи данных или команд: при низком уровне – команда, при высоком – данные. |
| 7 | CS | SS | Выбор экрана (используется при подключении нескольких дисплеев). Активный низкий уровень. |
| 8 | BL | LEDA, VDD33 | Подсветка дисплея. Напряжение питания 3.3 В. |

### **4.12.1 Подключение дисплея**

Дисплей подключается к микроконтроллеру по шине SPI. В нашем случае шина уже занята подключением модуля радиопередачи, описанного в пункте 4.11. Atmega328P-AU позволяет программно настроить протокол SPI на других выходах, поэтому подключение дисплея осуществляется следующим образом:

− GND – общая земля;

− VCC – шина +5 В устройства;

− SCL – выход PD7 микроконтроллера;

− SDA – выход PD6 микроконтроллера;

− RST – выход PD5 микроконтроллера;

− DC – выход PD4 микроконтроллера;

− CS – выход PD3 микроконтроллера;

− BL – шина +3.3 В устройства.

### **4.12.2 Программное управление дисплеем**

Для работы с дисплеем используются подключаемые библиотеки #include <LCDWIKI\_GUI.h> и #include <LCDWIKI\_SPI.h> [73].

Все выходы микроконтроллера по умолчанию установлены на OUTPUT, поэтому их конфигурация не обязательна, но при желании это можно сделать вызовом метода:

pinMode(PIN, OUTPUT); //PIN – номер выхода микроконтроллера

Далее необходимо создать объект дисплея, передав в него номера соответствующих выходов микроконтроллера, которые подключены к экрану:

//создание объекта дисплея: модель дисплея (ST7735S),

//выход CS, выход DC, не задается (подключен к 3.3 В),

//выход SDA, выход RST, выход SCL, выход BL

LCDWIKI\_SPI lcd(MODEL, CS, DC, -1, SDA, RST, SCL, BL);

После создания объекта дисплея, выполняется его инициализация:

//инициализация дисплея

lcd.Init\_LCD();

lcd.Fill\_Screen(0x0); //заполнить экран черным цветом

lcd.Set\_Rotation(3); //задать угол отрисовки

lcd.Set\_Text\_Size(1);

lcd.Set\_Text\_Mode(1);

//задать желтый цвет для отрисовки элементов

lcd.Set\_Draw\_color(255,255,0);

//отрисовать желтый прямоугольник по контуру экрана

lcd.Draw\_Rectangle(0 0, lcd.Get\_Display\_Width()-1, lcd.Get\_Display\_Height()-1);

//задать черный цвет для отрисовки элементов

lcd.Set\_Draw\_color(0,0,0);

//задать желтый цвет для текста

lcd.Set\_Text\_colour(255, 100, 0);

В проекте используются методы, представленные ниже. Описание методов можно найти в источнике [73].

//задает цвет выводимо текста

lcd.Set\_Text\_colour();

//заполнение области цветом

lcd.Fill\_Rectangle();

//вывод строки

lcd.Print\_String();

//высота дисплея в пикселях

lcd.Get\_Display\_Height();

//длина дисплея в пикселях

my\_lcd.Get\_Display\_Width()

Для полной очистки экрана выполняется следующий код:

lcd.Fill\_Screen(0x0); //заполнить экран черным цветом

//задать желтый цвет для отрисовки элементов

lcd.Set\_Draw\_color(255,255,0);

//отрисовать желтый прямоугольник по контуру экрана

lcd.Draw\_Rectangle(0 0, lcd.Get\_Display\_Width()-1, lcd.Get\_Display\_Height()-1);

## **4.13 Блок питания устройства управления**

Питание устройства управления поступает от аккумуляторной батареи Li-Ion 18650 3.7 В, представленной на принципиальной схеме элементом GB2. Аккумулятор подключается к микросхеме DD19 – контроллер заряда 03962A. В контроллере заряда присутствует порт microUSB (XP1), с помощью которого происходит зарядка установленного аккумулятора.

В блоке питания устанавливается переключатель SB1, который представляет собой ключ для отключения и включения питания устройства управления. Выход OUT+ микросхемы 03962A заводится на PB5 вход микроконтроллера DD2 и необходим для программного отслеживания заряда аккумулятора.

Питание, которое поступает от контроллера заряда, подаётся на повышающий DC-DC преобразователь BS01 – микросхема DD20. Выход OUT+ преобразователя подключается к шине +5 В устройства.

К шине +5 В подключается стабилизатор напряжения AMS1117 через вход IN. Вход GND стабилизатора подключается к общей земле. Выход OUT подключается к общей шине +3.3 В устройства. На выходе OUT также устанавливается фильтрующий конденсатор C9 ёмкостью 0.1 мкФ.

Дополнительно в блоке устанавливаются диоды Шоттки VD6 и VD7 для контроля полярности подключения питания. Также к шине +5 В подключается фильтрующий конденсатор C20, ёмкостью 1000 мкФ.

# **5 МОДЕЛИРОВАНИЕ**

Моделирование и тестирование разработанной системы являются неотъемлемыми этапами в разработке любого технического проекта, которые можно разделить на 2 этапа – моделирование аппаратных средств и тестирование программного обеспечения. В этом разделе будет описан каждый из этих этапов. Схема программы представлена на чертеже ГУИР.400201.022 ПД.

## **5.1 Тестирование аппаратных средств**

В данном подразделе будут рассмотрены функциональные тесты для проверки правильности работы аппаратных средств устройства.

### **5.1.1 Тестирование блока питания котроллера полёта**

Задачей блока является обеспечение питанием компонентов летательного аппарата, поэтому проверяемый функционал это:

1. Защита устройства от подачи обратного напряжения питания.

2. Защита устройства от протекания тока обратно в источник питания.

3. Защита устройства от скачков напряжения на источнике питания.

Функциональные тесты для проверки правильности работы блока представлены в таблице 5.1.

Таблица 5.1 – Функциональные тесты для блока питания контроллера полёта

|  |  |
| --- | --- |
| **Содержание теста** | **Ожидаемый результат** |
| Подача напряжения +7.4 В на вход питания устройства | Напряжение должно поступить на вход всех ESC, а также напряжение +5 В и +3.3 В должно поступить на вход компонентов устройства |
| Подача напряжения +12 В на вход питания устройства | Напряжение должно поступить на вход всех ESC, а также напряжение +5 В и +3.3 В должно поступить на вход компонентов устройства |
| Подача отрицательного напряжения на вход питания устройства | Компоненты устройства должны быть обесточены |
| Соединение линии питания устройства с землей, подача напряжения +5 В на вход микропроцессорного устройства | Линия питания устройства разорвана |

### **5.1.2 Тестирование блока управления двигателями**

Задачей блока управления двигателями является запуск и остановка двигателей, поэтому проверяемый функционал это:

1. Инициализация ESC

2. Запуск двигателей на разной мощности.

3. Остановка двигателей.

Функциональные тесты для проверки правильности работы блока представлены в таблице 5.2. Важно при проведении тестов снять с двигателей пропеллеры.

Таблица 5.2 – Функциональные тесты для блока управления двигателями

|  |  |
| --- | --- |
| **Содержание теста** | **Ожидаемый результат** |
| Подача напряжения на вход питания летательного аппарат | Питание поступило на вход всех ESC, о чем свидетельствует звуковой сигнал каждого ESC |
| Подача на ESC ШИМ-сигнала длительностью 1500 мс | Двигатели работают на 50% мощности |
| Подача на ESC ШИМ-сигнала длительностью 2000 мс | Двигатели работают на максимальной мощности |
| Подача на ESC ШИМ-сигнала длительностью 1200 мс | Двигатели работают на 20% мощности |
| Подача на ESC ШИМ-сигнала длительностью 1000 мс | Двигатели остановились |

### **5.1.3 Тестирование блока радиопередачи контроллера полёта и блока радиопередачи устройства управления**

Задачей блоков радиопередачи является обмен информацией между устройством управления и летательным аппаратом, поэтому проверяемый функционал это:

1. Инициализация модулей.

2. Отправка данных от устройства управления на летательный аппарат.

Функциональные тесты для проверки правильности работы блока представлены в таблице 5.3.

Таблица 5.3 – Функциональные тесты для блоков радиопередачи

|  |  |
| --- | --- |
| **Содержание теста** | **Ожидаемый результат** |
| Подача напряжения на вход питания устройства управления | Питание +3.3 В поступило на вход модуля радиопередачи устройства управления |
| Подача напряжения на вход питания летательного аппарата | Питание +3.3 В поступило на вход модуля радиопередачи летательного аппарата |
| Отправка данных с устройства управления | Данные отправлены и успешно приняты летательным аппаратом |

### **5.1.4 Тестирование блока отображения информации**

Задачей блока является обмен вывод цветной информации на ЖК дисплей устройства управления, поэтому проверяемый функционал это:

1. Инициализация экрана.

2. Отображение цветной информации.

Функциональные тесты для проверки правильности работы блока представлены в таблице 5.4.

Таблица 5.4 – Функциональные тесты для блока отображения информации

|  |  |
| --- | --- |
| **Содержание теста** | **Ожидаемый результат** |
| Подача напряжения на вход питания устройства управления | Питание +5 В поступило на вход экрана, о чем свидетельствует включённая подсветка |
| Отправка команды на вывод строки заданного цвета на экран | На экране полностью отобразилась отправленная строка с заданным цветом |
| Отправка команды на вывод цветного изображения на экран | На экране полностью отобразилось заданное изображение |
| Отправка команды на заполнение экрана цветом | Экран заполнился заданным цветом |

### **5.1.5 Тестирование блока управления летательным аппаратом**

Задачей блока является получение команд от пользователя для управления летательным аппаратом, поэтому проверяемый функционал это:

1. Инициализация джойстиков.

2. Изменение показаний джойстиков.

Функциональные тесты для проверки правильности работы блока представлены в таблице 5.5.

Таблица 5.5 – Функциональные тесты для блока управления летательным аппаратом

|  |  |
| --- | --- |
| **Содержание теста** | **Ожидаемый результат** |
| Подача напряжения на вход питания устройства управления | Питание +5 В поступило на вход всех джойстиков |
| Установка всех джойстиков в центр | Значение по осям X и Y обоих джойстиков находится в середине диапазона |
| Перемещение всех джойстиков по оси X в нулевую позицию | Значение по оси X обоих джойстиков равно нулю |
| Перемещение всех джойстиков по оси Y в нулевую позицию | Значение по оси Y обоих джойстиков равно нулю |
| Перемещение всех джойстиков по оси X в максимальную позицию | Значение по оси X обоих джойстиков равно максимальному |
| Перемещение всех джойстиков по оси Y в максимальную позицию | Значение по оси Y обоих джойстиков равно максимальному |

### **5.1.6 Тестирование блока взаимодействия с настройками управления**

Задачей блока является получение команд от пользователя по изменению состояния кнопок и тумблеров устройства управления, поэтому проверяемый функционал это:

1. Проверка работоспособности восьми кнопок SW2-SW9.

2. Проверка работоспособности двух тумблеров на 2 позиции TB3, TB4.

3. Проверка работоспособности двух тумблеров на 3 позиции TB1, TB2.

Функциональные тесты для проверки правильности работы блока представлены в таблице 5.6.

Таблица 5.6 – Функциональные тесты для блока взаимодействия с настройками управления

|  |  |
| --- | --- |
| **Содержание теста** | **Ожидаемый результат** |
| Нажатие кнопок SW2-SW9 с интервалом 5 секунд | Определилось нажатие каждой кнопки |
| Переключение тумблера на 2 позиции TB3 | Определилось переключение тумблера TB3 |
| Переключение тумблера на 2 позиции TB4 | Определилось переключение тумблера TB4 |
| Переключение тумблера на 3 позиции TB1 вперёд | Определилось переключение тумблера TB1 вперёд |
| Переключение тумблера на 3 позиции TB1 назад | Определилось переключение тумблера TB1 назад |
| Переключение тумблера на 3 позиции TB2 вперёд | Определилось переключение тумблера TB2 вперёд |
| Переключение тумблера на 3 позиции TB3 назад | Определилось переключение тумблера TB2 назад |

### **5.1.7 Тестирование блока питания устройства управления**

Задачей блока является обеспечение питанием компонентов устройства управления. Для данного компонента проверяемый функционал это:

1. Защита устройства от подачи обратного напряжения питания.

2. Защита устройства от протекания тока обратно в источник питания.

3. Защита устройства от скачков напряжения на источнике питания.

4. Зарядка устройства управления.

Функциональные тесты для проверки правильности работы блока представлены в таблице 5.7.

Таблица 5.7 – Функциональные тесты для блока питания устройства управления

|  |  |
| --- | --- |
| **Содержание теста** | **Ожидаемый результат** |
| Подача напряжения +3.7 В на вход питания устройства | Напряжение +5 В и +3.3 В должно поступить на вход компонентов устройства |
| Подача напряжения +5 В на вход питания устройства | Напряжение +5 В и +3.3 В должно поступить на вход компонентов устройства |
| Подача отрицательного напряжения на вход питания устройства | Компоненты устройства должны быть обесточены |
| Соединение линии питания устройства с землёй, подача напряжения +5 В на вход микропроцессорного устройства | Линия питания устройства разорвана |
| Подключение блока питания к порту microUSB контроллера заряда | Должен загореться красный светодиод на контроллере заряда и начаться процесс зарядки |

## **5.2 Моделирование программного обеспечения**

В данном подразделе будет рассмотрен алгоритм работы контроллера полёта, а также произведено моделирование программного обеспечения летательного аппарата.

### **5.2.1 Алгоритм работы устройства**

В данном подразделе рассмотрим основной алгоритм работы летательного аппарата. Схема алгоритма работы программы представлена на чертеже ГУИР.400201.022 ПД.

Описание алгоритма по шагам:

Шаг 1. Начало работы программы – включение устройства.

Шаг 2. Подготовка промежуточных данных и флагов – инициализация начальных параметров.

Шаг 3. Конфигурация входов и выходов микроконтроллера.

Шаг 4. Настройка интерфейса I2C – задание скорости тактирования и определение адресов.

Шаг 5. Чтение конфигурации из памяти устройства (EEPROM).

Шаг 6. Проверка правильности данных EEPROM. Если данные корректны, то выполняется переход к шагу 7. Если данные некорректны, то мы остаёмся в шаге 6, пока не будет выполнена настройка и перезапуск устройства.

Шаг 7. Настройка регистров и адреса подключения модуля GY-521.

Шаг 8. Проверка подключения модуля GY-521. Если проверка прошла успешно, то переходим к шагу 9. В ином случае остаёмся в шаге 8 до устранения проблемы с подключением и перезапуска устройства.

Шаг 9. Калибровка ESC – выполняется калибровка всех четырёх ESC.

Шаг 10. Калибровка модуля GY-521.

Шаг 11. Настройка регистров и адреса подключения модуля GY-68.

Шаг 12. Калибровка модуля GY-68.

Шаг 13. Получение показания начального давления с модуля GY-68 и высчитывание высоты.

Шаг 14. Инициализация структуры данных модуля NRF24L01 для передачи/получения данных по радиоканалу.

Шаг 15. Конфигурация модуля NRF24L01 – выполнение всех необходимых настроек и задание параметров модуля радиопередачи.

Шаг 16. Проверка подключения модуля NRF24L01. Если проверка прошла успешно, то переходим к шагу 17. В ином случае остаёмся в шаге 16 до устранения проблемы с подключением и перезапуска устройства.

Шаг 17. Ожидание установки соединения с устройством управления. Данный шаг продолжается до тех пор, пока не будет установлено соединение.

Шаг 18. Ожидание установки нулевой скорости на устройстве управления. Данный шаг продолжается до тех пор, пока не будет выставлена нулевая скорость.

Шаг 20. Получение текущего времени в миллисекундах для дальнейших расчётов.

Шаг 21. Завершение настройки – начало бесконечного цикла.

Шаг 22. Вычисление показаний гироскопа в град./с для ПИД-регуляторов

Шаг 23. Вычисление смещения по данным из EEPROM.

Шаг 24. Расчёт полного вектора акселерометра.

Шаг 25. Вычисление угла крена.

Шаг 26. Вычисление угла тангажа.

Шаг 27. Корректировка показаний гироскопа на основе показаний акселерометра (устранение проблем с плавающим нулем).

Шаг 28. Вычисление поправки угла крена.

Шаг 29. Вычисление поправки угла тангажа.

Шаг 30. Проверяем получение данных от устройства управления. Если данные получены, то переходим к шагу 31. Если нет – переходим к шагу 37.

Шаг 31. Читаем данные от устройства управления.

Шаг 32. Текущее состояние старта «0» и правый джойстик в нижнем правом углу? Если да, то переходим к шагу 33. Если нет, то переходим к шагу 34.

Шаг 33. Устанавливаем состояние старта в «1».

Шаг 34. Текущее состояние старта «1» и правый джойстик находится на начальной позиции (в центре)? Если да, то переходим к шагу 35. Если нет, то переходим к шагу 37.

Шаг 35. Устанавливаем состояние старта в «2» – разблокировка двигателей.

Шаг 36. Сбрасываем параметры ПИД-регулятора при каждой разблокировке двигателей.

Шаг 37. Текущее состояние старта «2» и правый джойстик находится в левом нижнем углу? Если да, то переходим к шагу 38. Если нет, то переходим к шагу 39.

Шаг 38. Устанавливаем состояние старта в «0» – блокировка двигателей.

Шаг 39. Вычисление желаемого значения ПИД-регулятора для крена.

Шаг 40. Вычисление желаемого значения ПИД-регулятора для тангажа.

Шаг 41. Вычисление текущей ошибки для крена. Вычисляется путём разности текущего угла крена и желаемого.

Шаг 42. Вычисление общей ошибки для крена. Вычисляется путём суммирования всех предыдущих ошибок для крена.

Шаг 43. Вычисление выходного значения ПИД-регулятора для крена.

Шаг 44. Модуль корректировки скорости по крену больше 100? Если да, то переходим к шагу 45. Если нет, то переходим к шагу 46.

Шаг 45. Устанавливаем корректировку скорости по крену равной 100, учитывая знак.

Шаг 46. Вычисление текущей ошибки для тангажа. Вычисляется путём разности текущего угла тангажа и желаемого.

Шаг 47. Вычисление общей ошибки для тангажа. Вычисляется путём суммирования всех предыдущих ошибок для тангажа.

Шаг 48. Вычисление выходного значения ПИД-регулятора для тангажа.

Шаг 49. Модуль корректировки скорости по тангажу больше 100? Если да, то переходим к шагу 50. Если нет, то переходим к шагу 51.

Шаг 50. Устанавливаем корректировку скорости по тангажу равной 100, учитывая знак.

Шаг 51. Сохранение в памяти текущей ошибки для крена и тангажа для последующих вычислений ПИД-регуляторов.

Шаг 52. Текущее состояние старта «2» (двигатели разблокированы)? Если нет, то переходим к шагу 53. Если да, то переходим к шагу 54.

Шаг 53. Устанавливаем скорость каждого двигателя на 0, переходим к шагу 58.

Шаг 54. Заданная скорость больше 800? Если да, то переходим к шагу 55. Если нет, то переходим к шагу 56.

Шаг 55. Установить значение скорости равным 800.

Шаг 56. Вычисление скорости каждого двигателя на основе заданной скорости и выходных данных ПИД-регуляторов.

Шаг 57. Ограничиваем скорость каждого двигателя до 1000, если значения оказываются выше.

Шаг 58. Ожидаем, пока пройдёт 400 миллисекунд с момента запуска цикла для правильности расчётов.

Шаг 59. Получаем текущее время в миллисекундах.

Шаг 60. Вычисляем длительность ШИМ-сигнала для каждого двигателя на основу текущего времени и рассчитанной скорости.

Шаг 61. Обновляем скорость каждого двигателя путём подачи ШИМ-сигнала на ESC.

Шаг 62. Читаем данные для модуля GY-521 и GY-68.

Шаг 63. Конец бесконечного цикла, переход к шагу 21.

Шаг 63. Завершение работы программы – выключение устройства.

### **5.2.2 Подключение к шине I2C**

Для проверки подключения устройств к шине I2C реализовано программное обеспечение I2C\_scanner, представленное на DVD-диске «Аппаратно-программный комплекс «Беспилотный летательный аппарат с дистанционным управлением».

Данное программное обеспечение загружается в микроконтроллер и сканирует шину I2C по всем известным адресам. После определения, что по адресу подключено ведомое устройство, на экран выводится информация.

Для контроллера полёта было найдено два устройства, подключённых к шине I2C (см. рисунок 5.1). По адресу 0x68 подключён модуль GY-521, по адресу 0x77 – модуль GY-68.

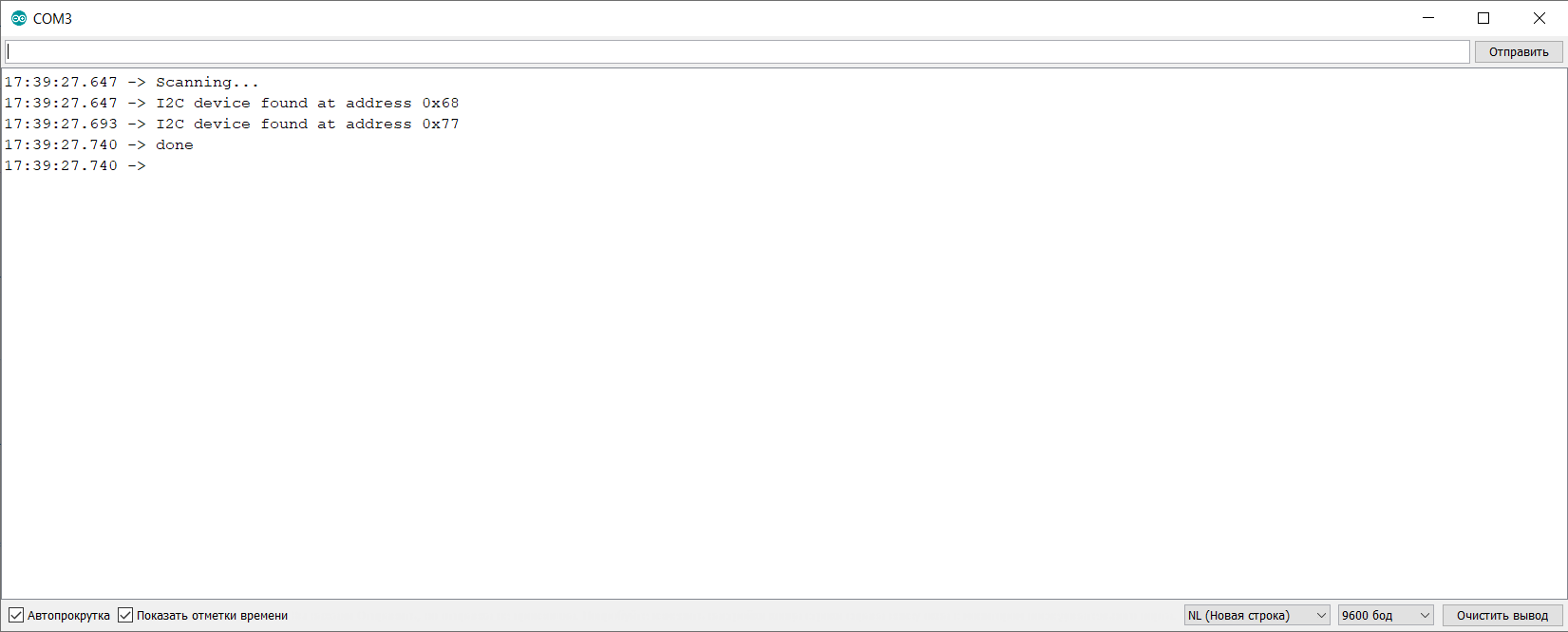


Рисунок 5.1 – Подключённые устройства по шине I2C контроллера полёта

### **5.2.3 Моделирование работы модуля GY-521**

Для моделирования работы модуля GY-521 (MPU6050) разработано программное обеспечение MPU5060\_modulation. Данное программное обеспечение выводит углы модуля по оси x и по оси y в реальном времени и с помощью него можно проверить работу MPU6050.

На рисунке 5.2 представлен график изменения углов осей X и Y. На рисунке 5.3 изображено изменение оси Y без изменения оси X, на рисунке 5.4 – изменение оси X без изменения оси Y. Красный график иллюстрирует ось X, синий – ось Y.

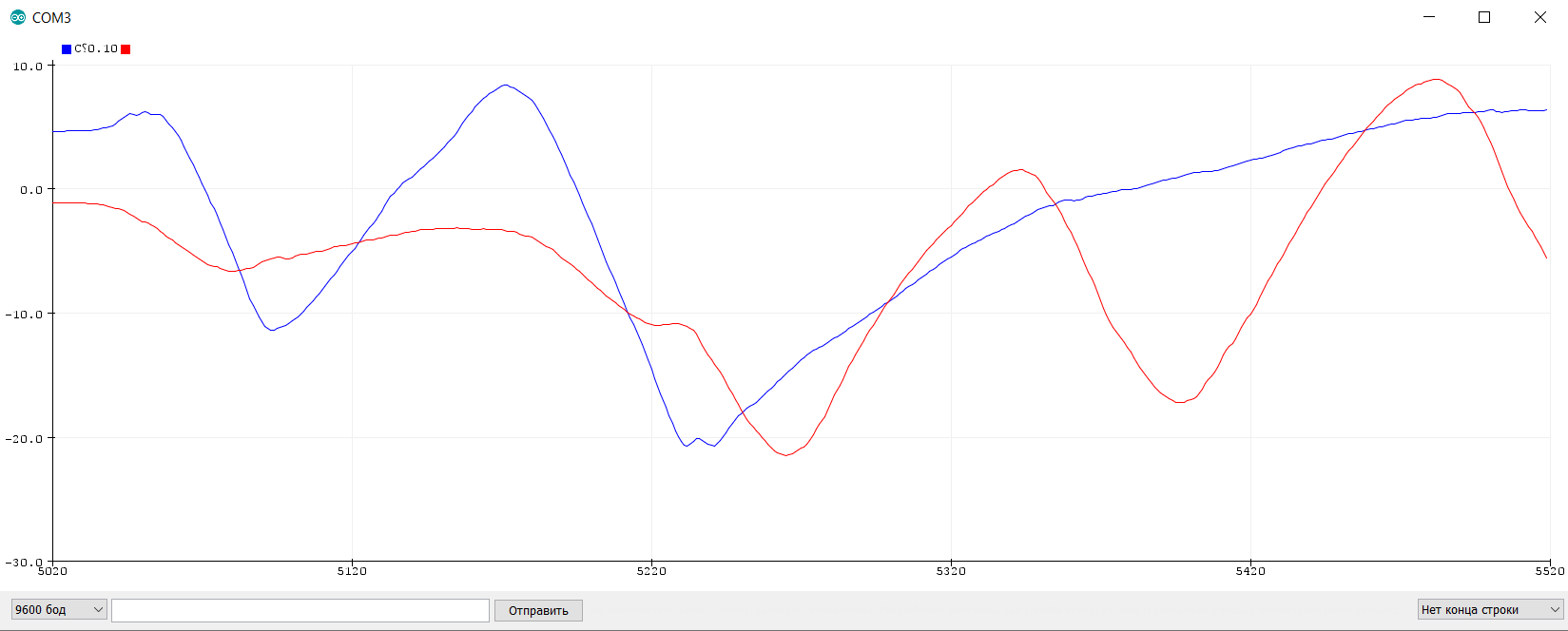


Рисунок 5.2 – Моделирование углов по оси X и y модуля GY-521 (MPU6050)

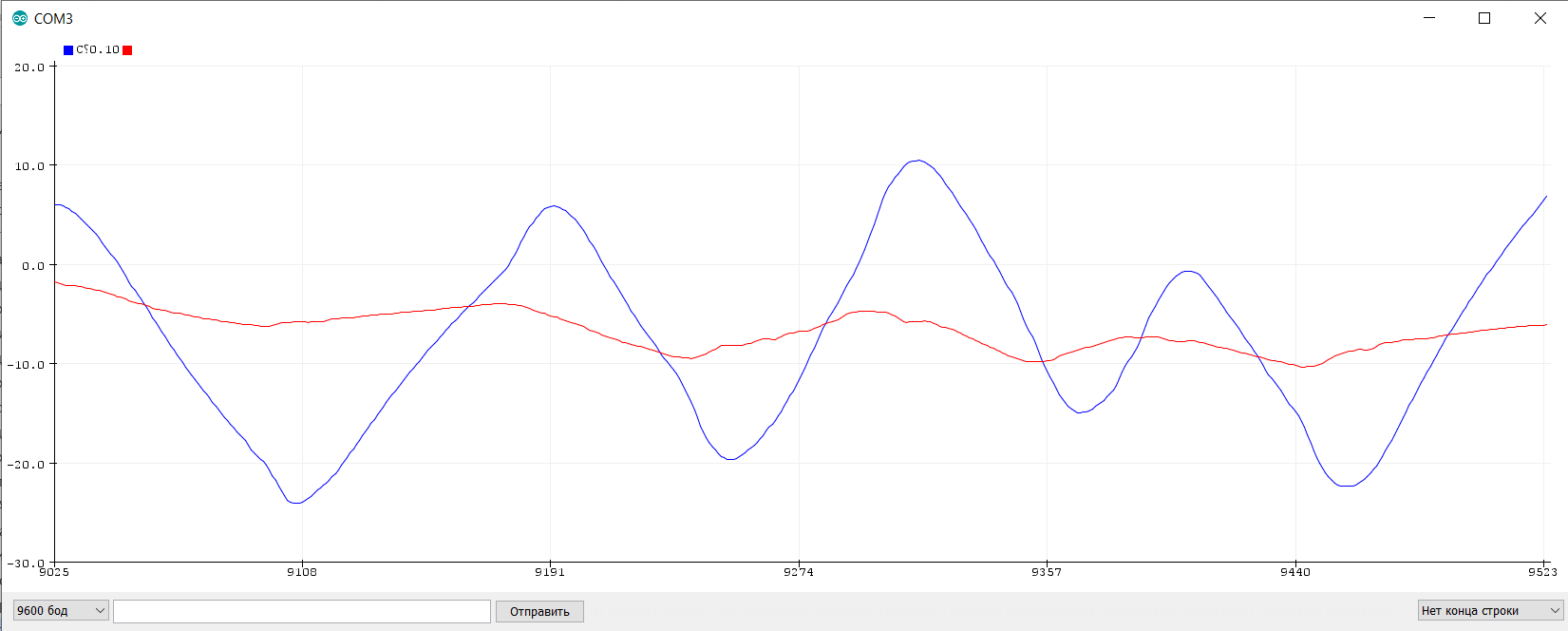


Рисунок 5.3 – Моделирование угла по оси Y модуля GY-521 (MPU6050)

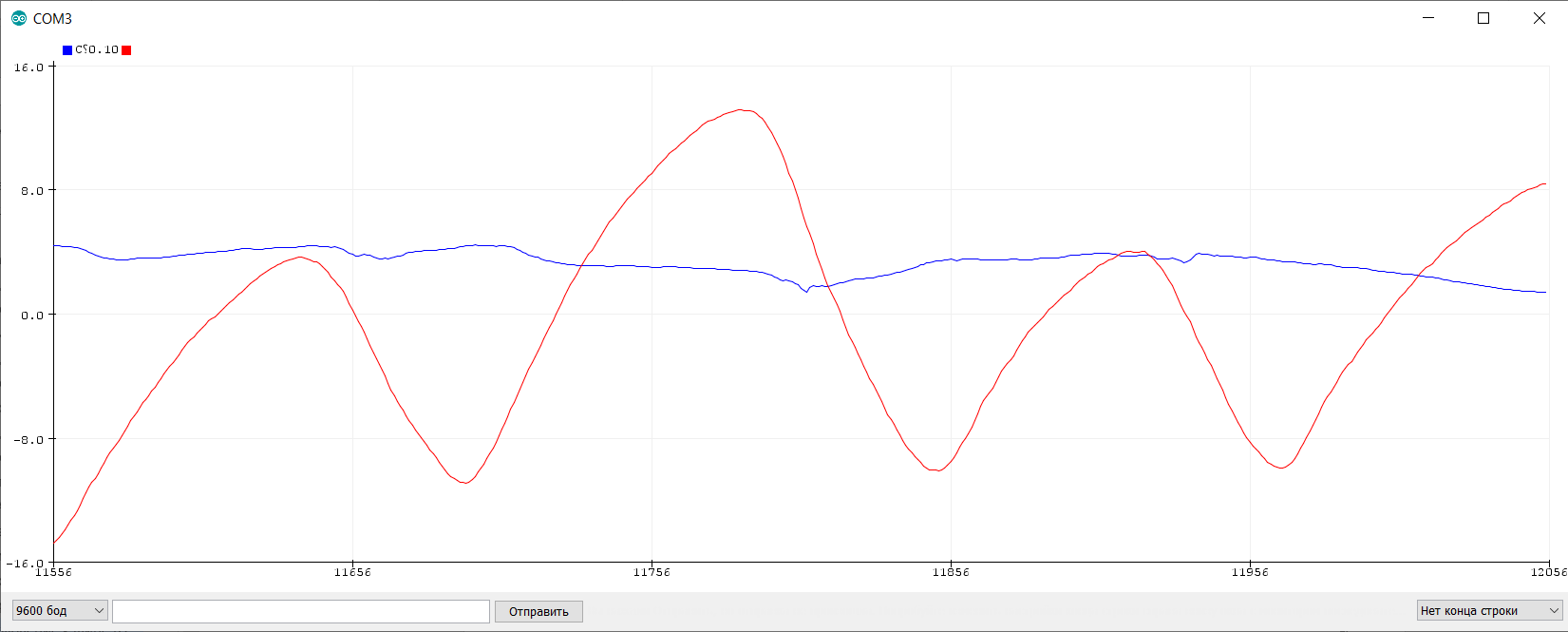


Рисунок 5.4 – Моделирование угла по оси X модуля GY-521 (MPU6050)

### **5.2.4 Моделирование изменение скоростей двигателей**

Для моделирования скоростей двигателей в программное обеспечение контроллера полёта Flight\_controller был добавлен вывод графиков.

На скорость двигателей влияет полученное значение от устройства управления и корректировки ПИД-регуляторов. Результаты модуляции представлены на рисунке 5.5. Синий график отражает изменение скорости переднего левого мотора, красный – переднего правого мотора, зелёный – заднего левого мотора, оранжевый – заднего правого мотора.

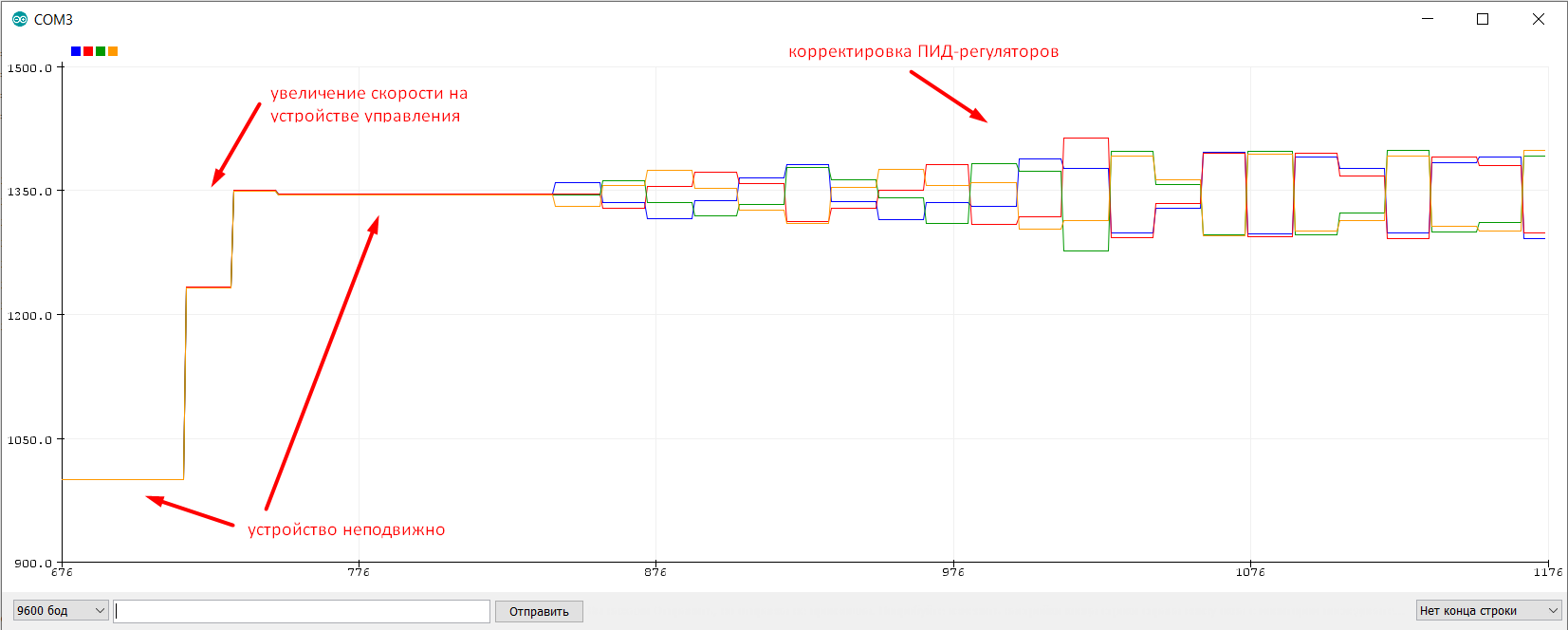


Рисунок 5.5 – Моделирование скоростей двигателей

# **6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

Данный раздел посвящён руководству по сборке, настройке и эксплуатации разработанного комплекса. Здесь даётся описание элементов управления, указываются требования к аппаратному и программному обеспечению, а также описывается процесс предварительной настройки устройства.

## **6.1 Руководство по сборке проекта**

Для сборки проекта необходимо реализовать аппаратную часть комплекса согласно описанным схемам, после чего загрузить разработанное программное обеспечение. Дополнительно на 3D-принтере печатается корпус для удобства эксплуатации и, при желании, устанавливается защита на пропеллеры.

При сборке проекта модуль инерционных датчиков MPU6050 необходимо поместить в центр устройства и обеспечить его неподвижность относительно летательного аппарата для лучшего определения ориентации в пространстве. Направление осей модуля представлено на рисунке 6.1.

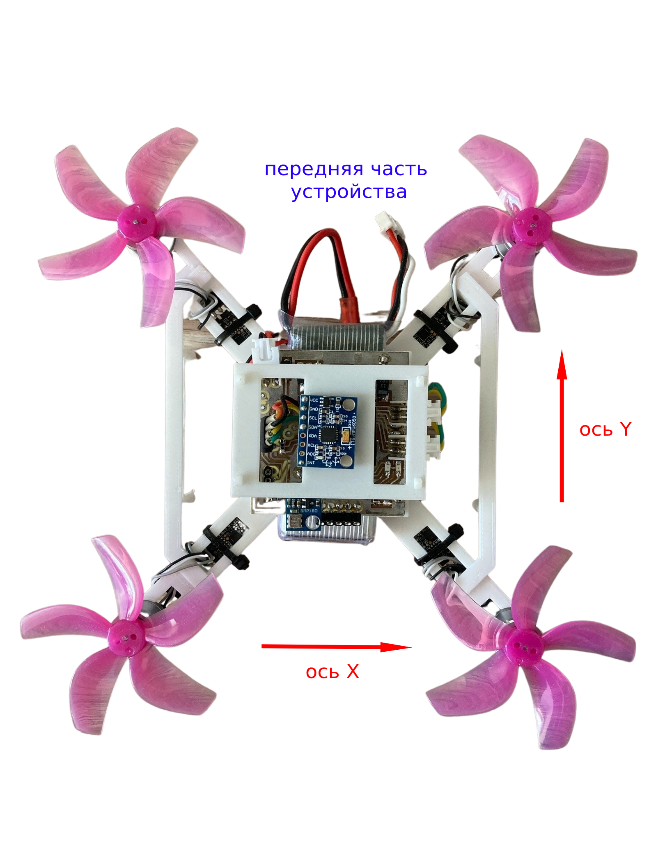


Рисунок 6.1 – Установка модуля MPU6050

### **6.1.1 Требования к аппаратному и программному обеспечению**

Данный проект состоит из летательного аппарата, устройства управления и программного обеспечения для этих устройств. Помимо этого, реализовано программное обеспечение для предварительной настройки контроллера полёта.

Компоненты, необходимые для реализации аппаратной части проекта представлены на принципиальной схеме ГУИР.400201.022 Э3. Помимо этого понадобится внешний программатор CP2102 и кабель с выходом USB Type-B для программирования микроконтроллеров.

Программное обеспечение контроллера полёта использует следующие библиотеки:

− Wire [66];

− SPI [67];

− EEPROM [68];

− nRF24L01/ RF24 [69];

− I2Cdev [70];

− MPU6050 [71];

− Adafruit\_BMP085 [72].

Программное обеспечение устройства управления использует следующие библиотеки:

− Wire [66];

− SPI [67];

− nRF24L01/ RF24 [69];

− LCDWIKI\_GUI/LCDWIKI\_SPI [73];

− Adafruit\_MCP23017 [74].

### **6.1.2 Настройка окружения**

Для написания программного обеспечения была выбрана интегрированная среда разработки Arduino IDE [75].

Первым шагом является создание проекта для выбранного микроконтроллера, в данном проекте – это ATmega328P-AU. Для данного микроконтроллера необходимо выбрать плату Arduino Mini, как показано на рисунке 6.2 и процессор ATmega328P, как показано на рисунке 6.3.

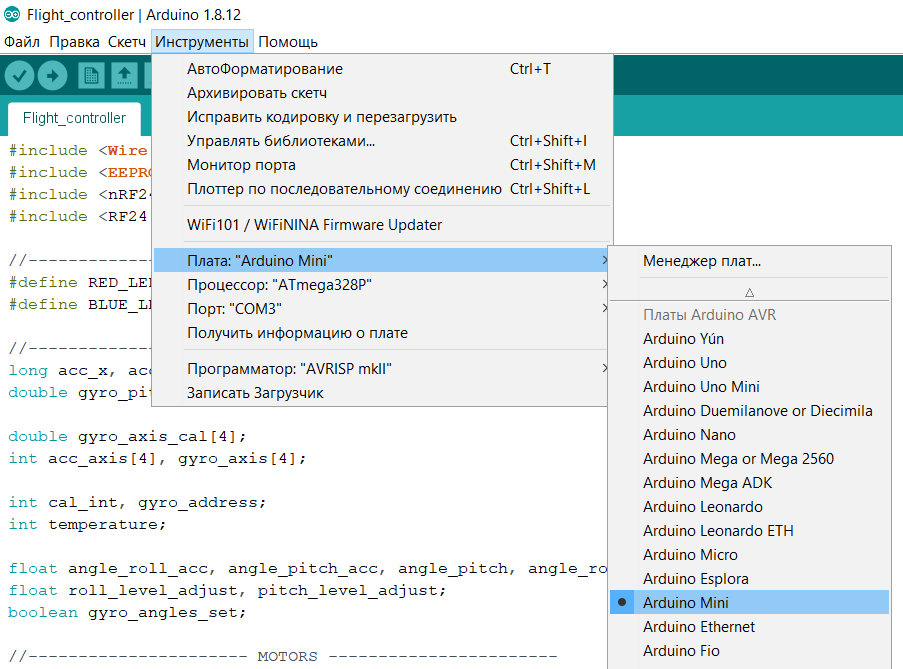


Рисунок 6.2 – Выбор микроконтроллера в Arduino IDE

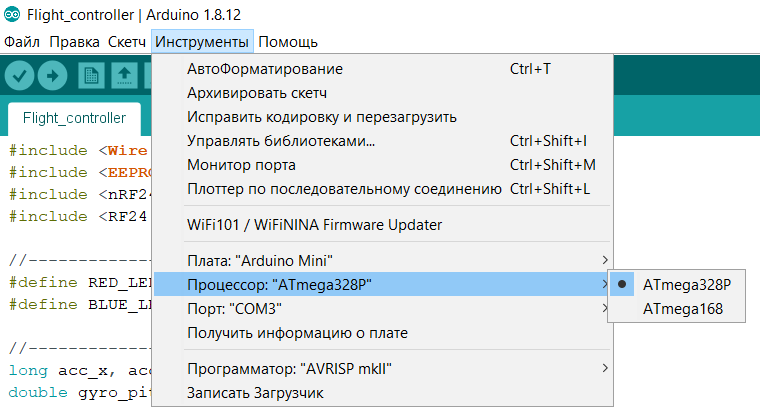


Рисунок 6.3 – Выбор процессора в Arduino IDE

Чтобы запустить проект, необходимо перейти во вкладку Файл и выбрать пункт Открыть, после чего найти в каталоге проект с расширением ino.

### **6.1.3 Загрузка программного обеспечения**

Для загрузки программного обеспечения в микроконтроллер необходимо подключить его к персональному компьютеру, используя внешний программатор CP2102 (для летательного аппарата) или кабель с выходом USB Type-B (для устройства управления), после чего выбрать порт, через который производилось подключение (см. рисунок 6.4).

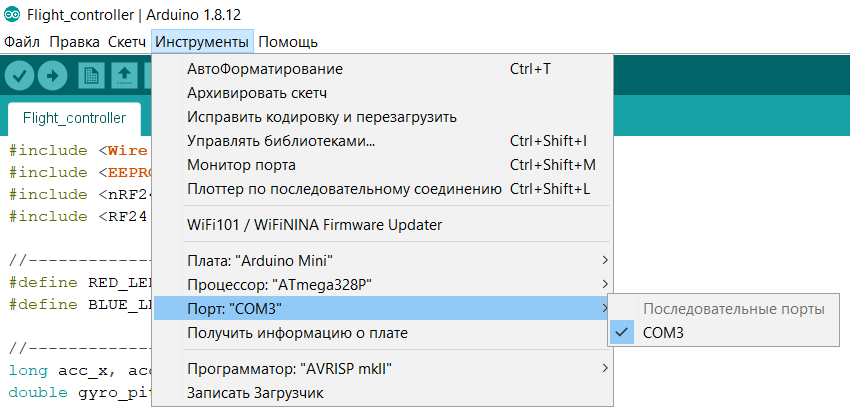


Рисунок 6.3 – Выбор порта в Arduino IDE

После того, как микроконтроллер был успешно подключён, необходимо нажать на кнопку, представленную на рисунке 6.5. Дополнительно программное обеспечение можно проверить на наличие ошибок (см. рисунок 6.5).

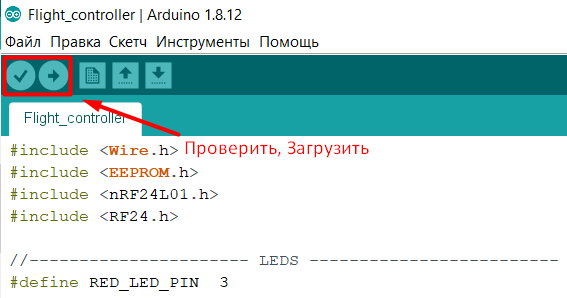


Рисунок 6.5 – Загрузка программного обеспечения на микроконтроллер

Для летательного аппарата необходимо загрузить программное обеспечение Flight\_controller, для устройства управления – Transmitter. Программное обеспечение представлено на DVD-диске «Аппаратно-программный комплекс «Беспилотный летательный аппарат с дистанционным управлением».

Предварительно летательный аппарат должен быть настроен, используя файл Flight\_controller\_setup, как описано в пункте 6.1.4.

### **6.1.4 Настройка контроллера полёта**

Настройка контроллера полёта представляет собой программное обеспечение, которое позволяет настроить шину I2C, проверить подключение устройства управления, а также определить адрес подключения модуля MPU6050 и откалибровать его. Дополнительно программное обеспечение проверяет работу светодиодов и сохраняет всю информацию в памяти устройства для дальнейшего использования. В конце программа проверяет, что данные записались в память корректно.

Для начала настройки контроллера полёта, необходимо загрузить в микроконтроллер файл Flight\_controller\_setup. После того, как программное обеспечение будет успешно загружено, необходимо запустить Монитор порта, как показано на рисунке 6.6. Важно при этом выставить скорость 9600 бод (см. рисунок 6.6).

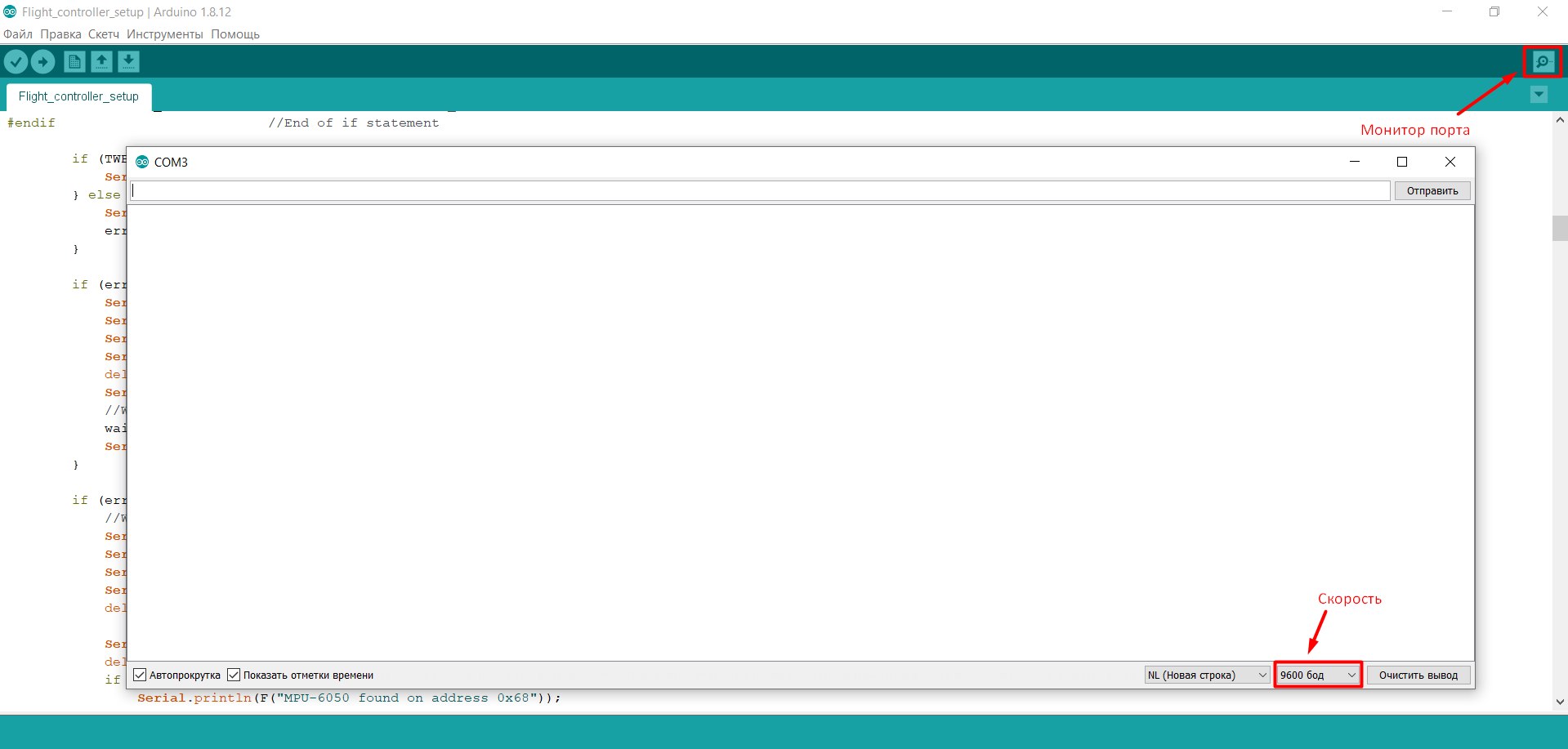


Рисунок 6.6 – Запуск монитора порта в Arduino IDE

После запуска монитора порта начнётся процесс настройки, который состоит из 11 этапов. Если на каком-то из этапов произойдёт ошибка настройки, то данные в память устройства не сохранятся и на экране будет отображено сообщение «The setup is aborted due to an error.» (см. рисунок 6.7.).

1. Инициализация модуля радиопередачи. Данный этап длиться до тех пор, пока модуль NRF24L01 не будет проинициализирован. По завершению на экране выведется сообщение «NRF24L01 initialized successfully» (см. рисунок 6.7.).

2. Инициализация шины I2C. На данном этапе скорость шины устанавливается равной 400 кГц. При успешной инициализации шины на экран выводится сообщение «I2C clock speed is correctly set to 400 kHz» (см. рисунок 6.7.). В противном случае отображается сообщение «I2C clock speed is not set to 400 kHz. (ERROR)» и настройка завершается с ошибкой.

3. Установка соединения с устройством управления. Летательный аппарат пытается подключиться к устройству управления по радиосвязи. Если соединение успешно установлено, то на экран выводится сообщение «Connection with control device successfully established» (см. рисунок 6.8). Если по прошествии 10 секунд связь не будет установлена, то на экране отобразится сообщение «No valid control device signals found!!! (ERROR)» и настройка завершится с ошибкой (см. рисунок 6.7).

4. Определение I2C адреса подключения модуля GY-521 (MPU6050). На этом этапе происходит обращение к модулю MPU6050 по разным адресам шины I2C (согласно технической спецификации – адрес 0x68 и 0x69), после чего ожидание ответа от модуля. Если модуль присылает ответ, то в памяти устройства сохраняется адрес, по которому происходило обращение (см. рисунок 6.9). Если ответ не поступает ни по одному адресу, то выводится сообщение «No MPU6050 module found!!! (ERROR)» (см. рисунок 6.8). В этом случае необходимо проверить подключение модуля к контроллеру полёта.

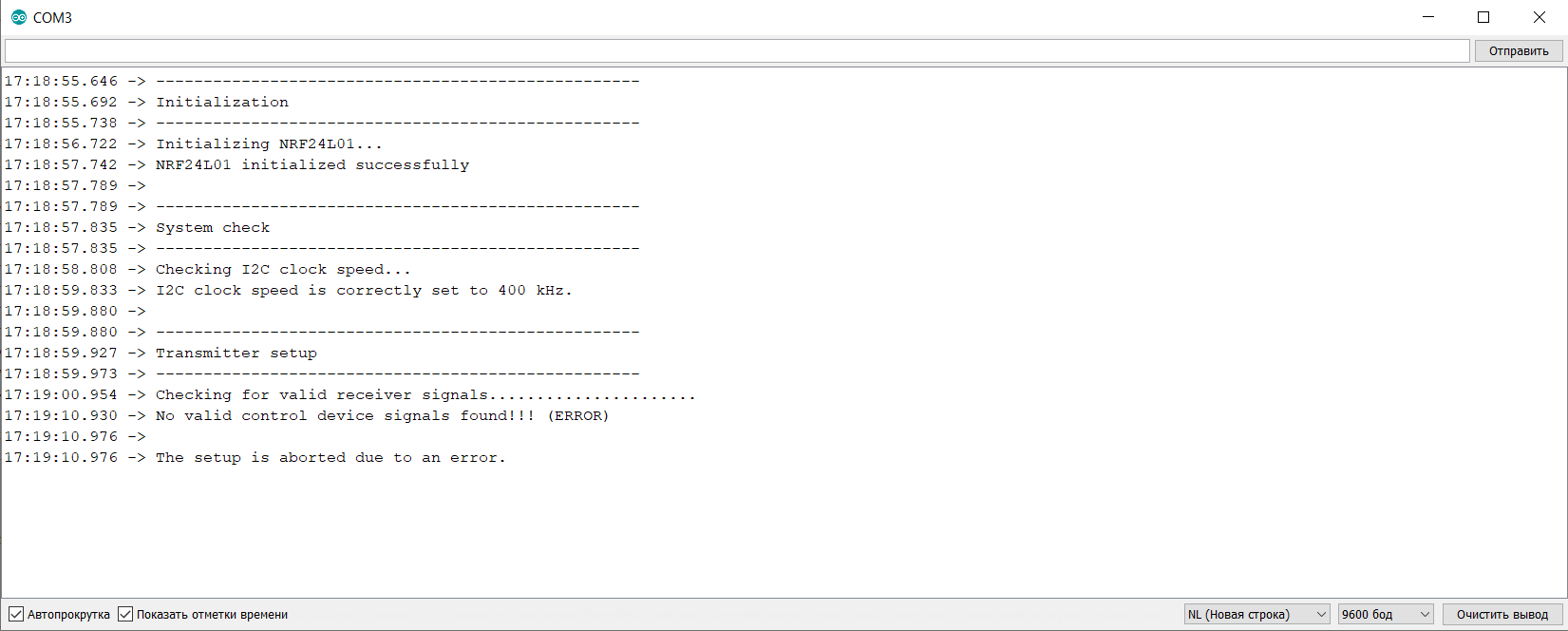


Рисунок 6.7 – Ошибка установки соединения с устройством управления

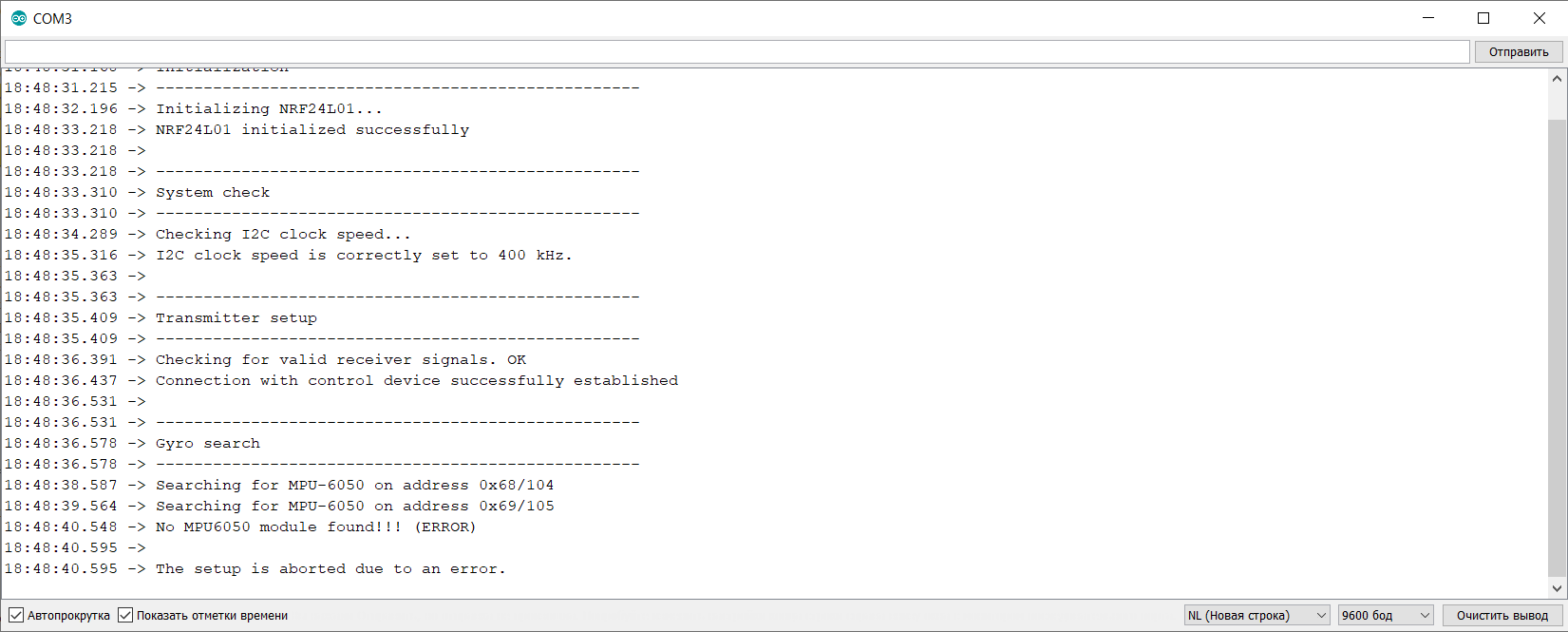


Рисунок 6.8 – Ошибка определения адреса модуля GY-521 (MPU6050)

5. Установка регистров модуля GY-521 (MPU6050). На этом этапе происходит обращение к модулю по ранее определённому адресу и установка всех необходимых регистров для работы с модулем (см. рисунок 6.9). Информация об устанавливаемых регистрах взята из технической спецификации.

6. Калибровка модуля GY-521 (MPU6050). После задания всех необходимых регистров, происходит калибровка модуля. Для этого происходит чтение данных MPU6050 2000 раз с интервалом 5 миллисекунд. Далее вычисляется среднее значение – это начальное отклонение, которое сохраняется в памяти устройства и участвует в дальнейших расчётах.

Результаты калибровки выводятся на экран, как представлено на рисунке 6.9. Вся калибровка длится около 10 секунд и в это время устройство должно быть неподвижно.

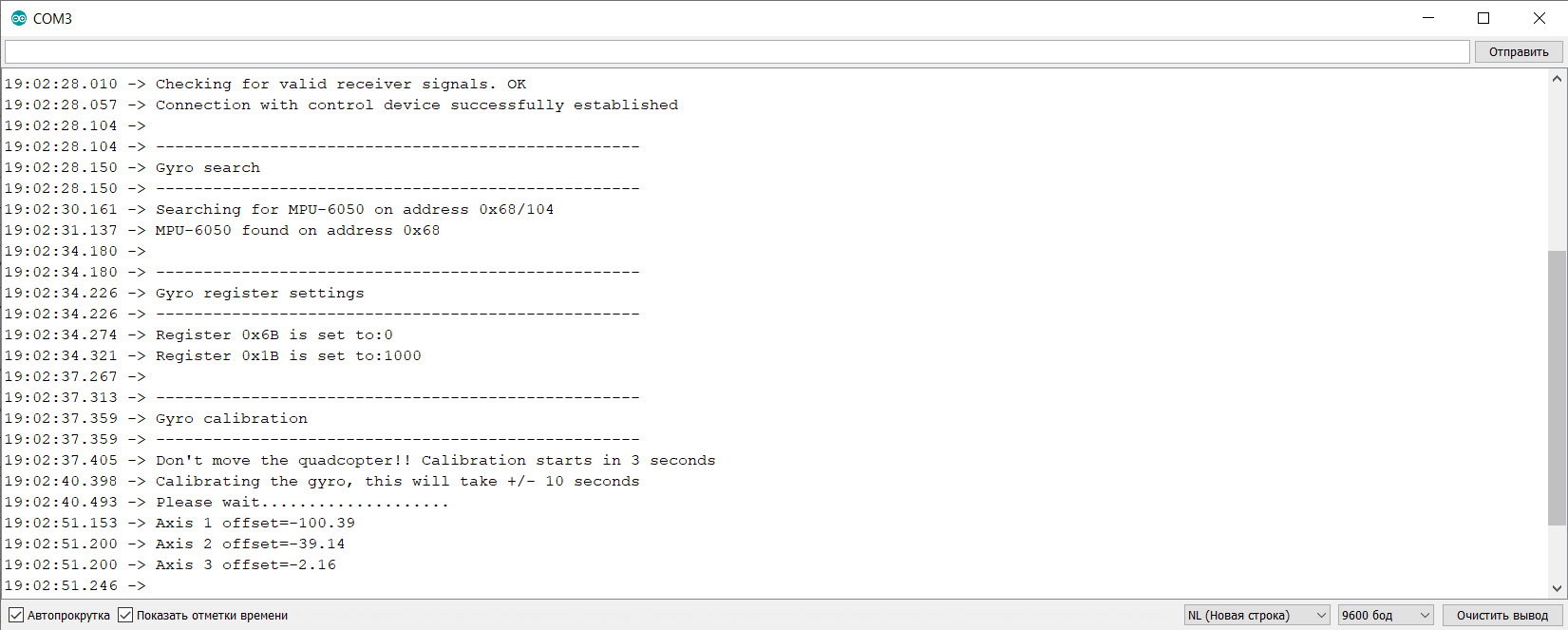


Рисунок 6.9 – Результаты калибровки модуля GY-521 (MPU6050)

7. Конфигурация осей гироскопа. Здесь происходит определение направления осей модуля GY-521 (MPU6050). Для этого необходимо выполнить 3 действия: поднять левую сторону БПЛА на 45°, поднять переднюю часть БПЛА на 45°, повернуть БПЛА на 45° в правую сторону. Информация о каждом действии выводится на экран, для продолжения необходимо ввести в Монитор порта символ ‘c’. По завершению, на экране выводится информация обо всех осях MPU6050 (см. рисунок 6.10). Если какое-то из действий не выполняется в течении 10 секунд, то на экран выводится сообщение «No angular motion is detected in the last 10 seconds!!! (ERROR)» и настройка завершается с ошибкой.

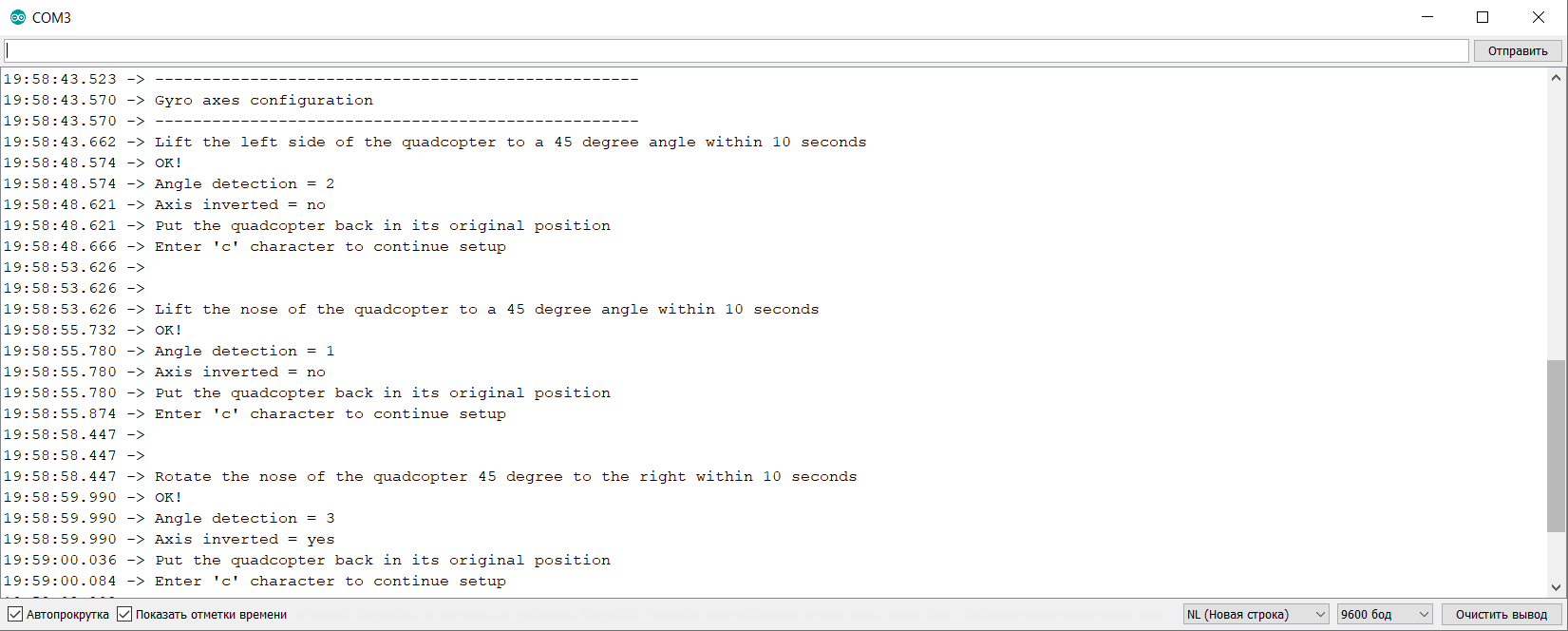


Рисунок 6.10 – Определение осей модуля GY-521 (MPU6050)

8. Проверка правильности определения осей модуля MPU6050. На данном этапе проверяется, что все оси модуля (ось x, y и z) были определены. Если проверка прошла успешно, то на экран выводится сообщение «Gyro axes ОК» (см. рисунок 6.11). В противном случае отображается сообщение «Gyro exes verification failed!!! (ERROR)»

9.  Проверка работоспособности светодиодов. Данный этап проверяет, что два подключённых светодиода на контроллере полёта корректно функционируют (см. рисунок 6.11). Для этого в начале зажигается красный светодиод, после ввода в Монитор порта символа ‘c’ красный светодиод гаснет и зажигается зелёный светодиод. После повторного ввода в Монитор порта символа ‘c’ зелёный светодиод гаснет. Если какой-то из светодиодов не загорелся, то необходимо его заменить.

10. Сохранение конфигурации в память устройства. На данном этапе происходит сохранение выполненных настроек в память устройства.

11. Проверка правильности сохранения информации в память устройства. После записи конфигурации в память устройства происходит чтение информации по записанным адресам и проверка, что считанные значения соответствуют записанным. Если проверка прошла успешно, то на экран выводится сообщение «Verification done», в противном случае – сообщение «EEPROM verification failed!!! (ERROR)».

После того, как все этапы настройки были успешно пройдены, на экран выводится сообщение «Setup is finished.» и можно загружать в контроллер полёта программное обеспечение (см. рисунок 6.11).

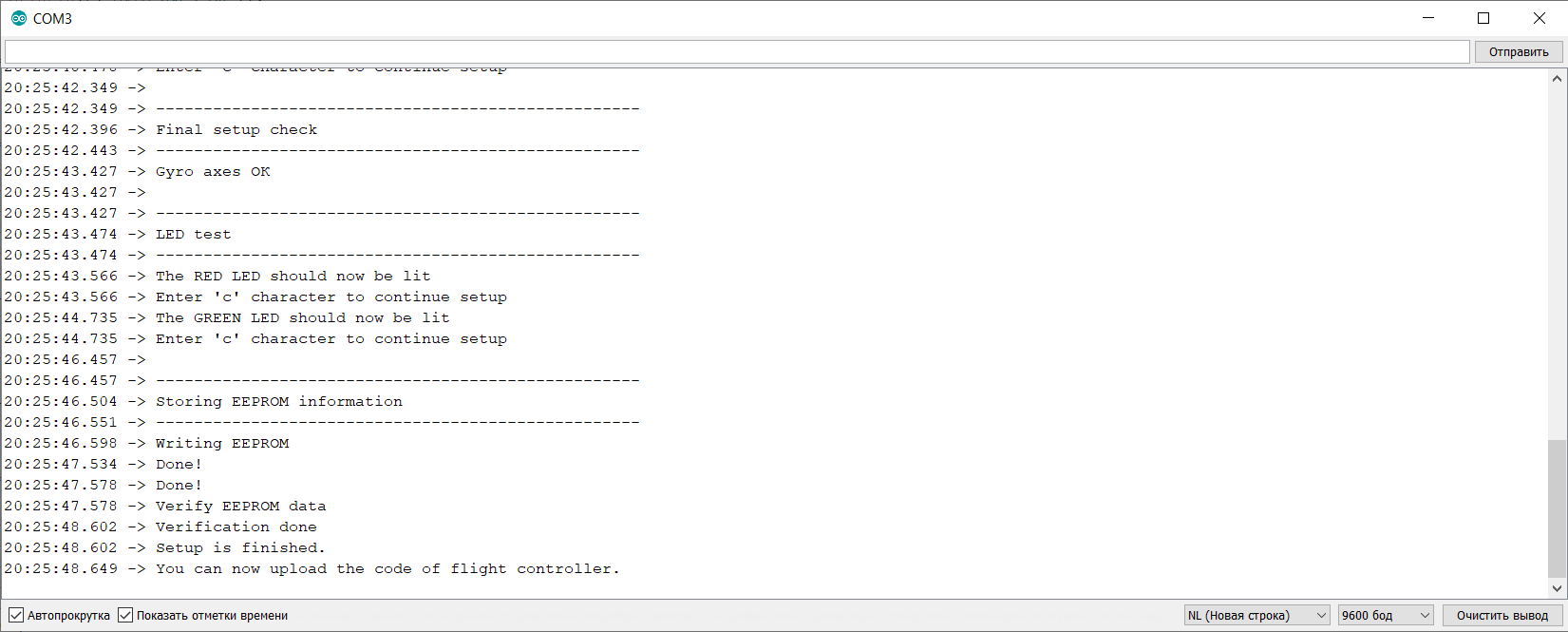


Рисунок 6.11 – Завершение настройки контроллера полёта

## **6.2 Руководство по эксплуатации устройства**

В данном подразделе описывается взаимодействие пользователя с аппаратно-программным комплексом «Беспилотный летательный аппарат с дистанционным управлением».

### **6.2.1 Включение и выключение устройства**

Для включения летательного аппарата аккумуляторную батарею необходимо подключить красным проводом к отметке «+», черным проводом к отметке «−» контроллера полёта. После этого батарея помещается в корпус БПЛА в специально отведённое под неё место. Для выключения аккумулятор необходимо отключить от контроллера полёта.

Устройство управления включается и выключается переключателем на левой стороне корпуса. Если устройство включено, то переключатель подсвечивается.

### **6.2.2 Зарядка устройства**

Заряжается БПЛА и устройство управления по-отдельности.

Для зарядки летательного аппарата необходимо отключить аккумуляторную батарею от контроллера полёта и подключить к блоку питания +5 В, используя зарядное устройство, представленное на рисунке 6.12.



Рисунок 6.12 – Зарядное устройство для БПЛА

На плату устройства управления установлен контроллер заряда, поэтому для его зарядки необходимо выключить устройство и подключить к блоку питания +5 В через microUSB порт, установленный на передней части корпуса.

### **6.2.3 Индикаторы состояний летательного аппарата**

Для индикации состояния летательного аппарата предусмотрены два светодиода: зелёный и красный. Графическое изображение включённых светодиодов представлено на рисунке 6.13.

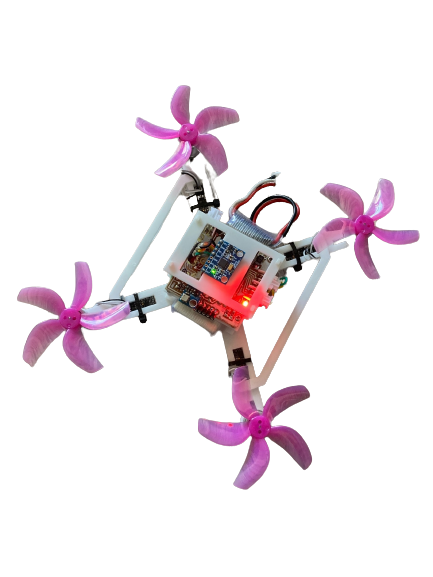
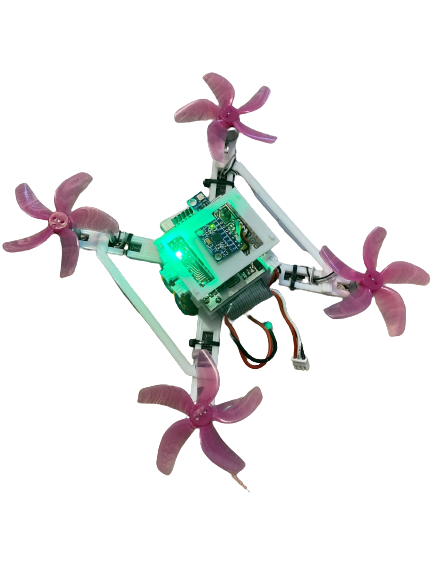
*а)**б)*

Рисунок 6.13 – Включённые светодиоды БПЛА: *а* – красный; *б* – зелёный

При включении летательного аппарата, начинается его инициализация. Если начинает мигать красный светодиод – это значит, что присутствуют ошибки в конфигурации устройства и необходимо его настроить, как описано в пункте 6.1.4. Если при запуске устройства загорается зелёный светодиод, то сохранённая конфигурация в памяти корректна.

Далее происходит калибровка двигателей и модуля GY-521 (MPU6050). В это время мигает зелёный светодиод.

Окончанием инициализации является подключение к устройству управления. В это время скорость на устройстве управления должна быть установлена на 0, чтобы избежать резкого запуска моторов БПЛА. Если на этом этапе мигает красный светодиод, то необходимо проверить подключение устройства управления и убедиться, что выставлена нулевая скорость.

После того, как все светодиоды погасли, можно начинать эксплуатацию устройства.

Во время работы устройства светодиоды могут находится в следующих состояниях:

− зелёный светодиод не горит − двигатели заблокированы;

− зелёный светодиод горит − двигатели разблокированы;

− красный светодиод не горит − ошибок в работе нет;

− красный светодиод горит − присутствуют ошибки в работе (например: время одних расчётов котроллера полёта превышает заданные 4000 мс);

− красный светодиод мигает − отсутствует подключение у устройству управления.

### **6.2.4 Управление летательным аппаратом**

Летательный аппарат управляется двумя джойстиками, установленными на устройство управления.

Левый джойстик отвечает за регулирование скорости БПЛА и вращение вокруг вертикальной оси (рысканье). Нулевая скорость летательного аппарата соответствует установки левого джойстика в верхнюю позицию. Для увеличения скорости джойстик опускается вниз. При повороте левого джойстика в левую или правую сторону, нос (передняя часть) летательного аппарата поворачивается в заданное направление.

Правый джойстик отвечает за наклон БПЛА вокруг его продольной оси (крен) и наклон относительно горизонтальной плоскости (тангаж). При перемещении правого джойстика в правую или левую сторону, летательный аппарат наклоняется в соответствующее направление. При перемещении правого джойстика вверх, аппарат наклоняется вперёд, при перемещении вниз − наклоняется назад.

Графическое изображение функций джойстиков представлено на рисунке 6.14.

*а)* *б)*

Рисунок 6.14 – Функции джойстиков: *а* – задачи; *б* – поведение БПЛА

### **6.2.5 Блокировка и разблокировка двигателей**

Блокировка и разблокировка двигателей реализована для предотвращения непредвиденного запуска моторов и обеспечения безопасности как самого летательного аппарата, так и оператора.

Чтобы запустить двигатели, нужно предварительно их разблокировать. Для этого необходимо выставить нулевую скорость и переместить правый джойстик в правый нижний угол. Если двигатели разблокированы – горит зелёный светодиод.

Для блокировки двигателей необходимо выставить нулевую скорость и переместить правый джойстик в левый нижний угол. Зелёный светодиод должен погаснуть.

Графическое изображение направления джойстиков для блокировки и разблокировки двигателей представлено на рисунке 6.15.



Рисунок 6.15 – Блокировка и разблокировка двигателей

### **6.2.6 Отображаемая информация на устройстве управления**

Отображаемая информация на ЖК дисплее устройства управления представлена на рисунке 6.16.

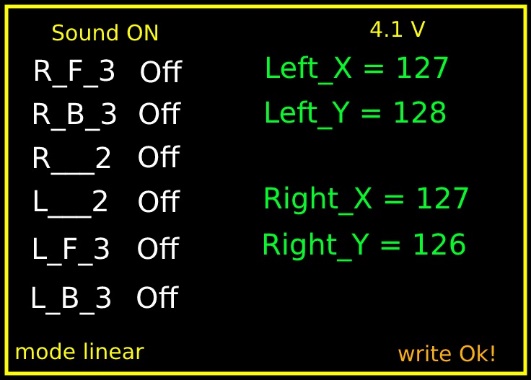


Рисунок 6.16 – Информация на экране устройства управления

В верхнем левом углу экрана отображается состояние звука. Если указано ON, то звук включен. Если отображается OFF – выключен.

В верхнем правом углу отображается заряд аккумулятора.

В нижнем левом углу отображается выбранный режим, в данном случае «линейный», также может быть «экспоненциальный». В зависимости от выбранного режима меняется получение данных для управления летательным аппаратом. Если выбран нормальный режим, то данные отправляются такими, какие они получены с джойстиков. Если выбран экспоненциальный режим, то значения меняются по экспоненциальному закону.

В нижнем правом углу отображается состояние передачи данных по радиосвязи. Если данные были успешно отправлены, то отображается сообщение «write Ok!», в противном случае – сообщение «Error NRF».

В правой части экрана выводится значения джойстиков по двум осям:

− Left\_X − смещение по оси X левого джойстика;

− Left\_Y − смещение по оси Y левого джойстика;

− Right\_X − смещение по оси X правого джойстика;

− Right\_Y − смещение по оси Y правого джойстика.

В левой части экрана отображается состояние тумблеров:

− R\_F\_3 − правый тумблер на 3 позиции, положение «вперёд»;

− R\_B\_3 − правый тумблер на 3 позиции, положение «назад»;

− R\_\_\_2 − правый тумблер на 2 позиции;

− L\_\_\_2 − левый тумблер на 2 позиции;

− L\_F\_3 − левый тумблер на 3 позиции, положение «вперёд»;

− L\_B\_3 − левый тумблер на 3 позиции, положение «назад».

Off означает, что позиция неактивна, On − тумблер установлен на позиции.

### **6.2.7 Изменение настроек на устройстве управления**

Управление настройками устройства управления осуществляется с помощью установленных тактовых кнопок. Расположение и обозначение кнопок представлено на рисунке 6.17.



Рисунок 6.17 – Расположение и обозначение тактовых кнопок

Нажатие одной из кнопок выполняет следующее действие:

− SW2 − увеличение начального рысканья на 1 (ось X левого джойстика);

− SW3 − уменьшение начального рысканья на 1 (ось X левого джойстика);

− SW4 − увеличение начальной скорости на 1 (ось Y левого джойстика);

− SW5 − уменьшение начальной скорости на 1 (ось Y левого джойстика);

− SW6 − увеличение начального тангажа на 1 (ось Y правого джойстика);

− SW7 − уменьшение начального тангажа на 1 (ось Y правого джойстика);

− SW8 − увеличение начального крена на 1 (ось X правого джойстика);

− SW9 − уменьшение начального крена на 1 (ось X правого джойстика);

Все изменения сразу сохраняются в память устройства и отображаются на экране.

# **7 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ И РЕАЛИЗАЦИИ НА РЫНКЕ АППАРАТНО-ПРОГРАММНОГО КОМПЛЕКСА «БЕСПИЛОТНЫЙ ЛЕТАТЕЛЬНЫЙ АППАРАТ С ДИСТАНЦИОННЫМ УПРАВЛЕНИЕМ»**

## **7.1 Характеристика аппаратно-программного комплекса**

Разрабатываемый в данном дипломном проекте аппаратно-программный комплекс «Беспилотный летательный аппарат с дистанционным управлением» является техническим устройством, состоящим из аппаратной части и программного обеспечения, предназначенным для выполнения полётов без участия человека на борту, управляемым с земли с помощью дистанционного управления. БПЛА широко используются в различных сферах, включая оборонную промышленность, сельское хозяйство, воздушные доставки, спасательные операции, контроль за транспортом и другие области, где требуется выполнение задач в труднодоступных и опасных условиях. Его пользователями могут быть как частные лица, так и владельцы предприятий.

Преимущество разрабатываемого аппаратно-программного комплекса –возможность модернизации в зависимости от предполагаемой области использования, поэтому пользователь может менять параметры управления, полётные функции, добавлять или же изменять программное обеспечение. Помимо возможности расширения функционала устройства, ещё одним достоинством является относительно небольшая итоговая стоимость. Летательные аппараты, представленные на рынке, поставляются с закрытым исходным кодом, и их цена в разы превышает суммарную стоимость всех элементов разрабатываемого устройства.

**7.2 Расчёт экономического эффекта от производства аппаратно-программного комплекса**

Для определения результата от вложения инвестиций в проектирование и производство аппаратно-программного комплекса необходимо определить отпускную цену аппаратно-программного комплекса на основе расчёта затрат на производство аппаратной части и разработку программной части.

Расчёт прямых затрат на материалы и комплектующие изделия для производства аппаратной части комплекса осуществляется в соответствии с представленной в конструкторской документации дипломного проекта номенклатурой, нормой расхода материалов, количеством комплектующих на изделие и рыночным ценам. Для производства данного комплекса были использованы материалы, представленные в таблице 7.1, и комплектующие, представленные в таблице 7.2.

Расчёт затрат по статье «Основные и вспомогательные материалы», в которую включается стоимость необходимых для изготовления изделия основных и вспомогательных материалов, осуществляется по формуле (7.1).

где Kтр – коэффициент транспортных расходов;

n – номенклатура применяемых материалов;

Hp*i* – норма расхода материала *i*-го вида на единицу изделия, нат. ед./шт.;

Цотп*i* – цена за единицу материала *i*-го вида, р.

Таблица 7.1 — Расчёт затрат на основные и вспомогательные материалы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование материала | Единица измерения | Норма расхода материала | Цена за единицу материала, р. | Сумма, р. |
| 1. Припой ПОС 61, 100 г | г | 0,28 | 26,21 | 7,60 |
| 2. Флюс ЛТИ-120, 20 мл | мл | 0,4 | 5,79 | 2,32 |
| 3. МГТФ 0.12 кв.мм, Провод монтажный, 1 м | м | 2,8 | 1,55 | 4,34 |
| 4. Текстолит двусторонний 1.5 мм, 100х100 мм | мм | 1,8 | 9,80 | 17,64 |
| 5. Текстолит односторонний 1.5 мм, 100х100 мм | мм | 0,6 | 8,00 | 4,80 |
| 6. Кембрик, 1 м | м | 1,3 | 0,72 | 0,94 |
| 7. Хомут-стяжка 2.5х80 мм, 100 шт | шт | 0,1 | 1,52 | 0,15 |
| 8. Пластиковая нить ABS, 1000 г | г | 1,1 | 51,00 | 56,10 |
| 9. Разъем штыревой 2.54 мм, 40 шт | шт | 3,8 | 0,69 | 2,62 |
| 10. Пропеллер, 1 шт | шт | 4 | 0,72 | 2,88 |
| 11. Посадочное место для аккумулятора 18650, 1 шт | шт | 1 | 0,50 | 0,50 |
| 12. Разъем L-KLS1-151-W, 1 шт | шт | 1 | 0,29 | 0,29 |
| Итого |  | | | 100,18 |
| Всего с учетом транспортных расходов (1,1)(Pм) | | | | 110,20 |

Расчёт затрат по статье «Покупные комплектующие изделия, полуфабрикаты», осуществляется по формуле (7.2).

где Kтр – коэффициент транспортных расходов;

m – номенклатура применяемых комплектующих;

*Ni* – количество комплектующих *i*-го вида на единицу изделия, нат. ед./шт.;

Цотп*i* – цена за единицу комплектующего *i*-го вида, р.

Таблица 7.2 — Расчёт затрат на комплектующие изделия

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование комплектующего | Количество на изделие, шт. | Цена за единицу, р. | Сумма, р. |
| 1 | 2 | 3 | 4 |
| 1. Микроконтроллер ATmega328P-AU | 2 | 16,86 | 33,72 |
| 2. Модуль беспроводной связи NRF24L01 | 2 | 9,00 | 18,00 |
| 3. Модуль трёхосевого гироскопа и акселерометра GY-521 (MPU6050) | 1 | 8,78 | 8,78 |
| 4. Датчик атмосферного давления GY-68 (BMP180) | 1 | 9,90 | 9,90 |
| 5. Микросхема CH340G | 1 | 2,10 | 2,10 |
| 6. Микросхема MCP23017 | 1 | 5,83 | 5,83 |
| 7. Мискросхема 03962A | 1 | 2,38 | 2,38 |
| 5. Бесколлекторный двигатель LD-Power 1106 4200KV | 4 | 6,58 | 26,32 |
| 6. Регулятор оборотов двигателя Emax BlHeli ESC | 4 | 12,30 | 49,20 |
| 7. ЖК дисплей LCD 160x128 1.77 ST7735S | 1 | 18,96 | 18,96 |
| 8. Микротумблер MTS-101-A2 | 2 | 1,34 | 2,68 |
| 9. Микротумблер MTS-103 | 2 | 1,96 | 3,92 |
| 10. Кнопка тактовая IT-1102WB | 9 | 0,10 | 0,90 |
| 11. Регулируемый джойстик SHUAICHI | 2 | 24,65 | 49,3 |
| 12. Модуль EEPROM ATMEL 2ECL | 1 | 2,20 | 2,20 |
| 13. Аккумулятор LI-ION 18650 3.7 В | 1 | 26,00 | 26,00 |
| 14. Аккумулятор 803063 7.4 В | 1 | 32,00 | 32,00 |
| 15. Повышающий DC-DC преобразователь BS01 5V | 1 | 3,20 | 3,20 |
| 16. Пьезодинамик | 1 | 0,69 | 0,69 |
| 17. Резистор SMD 0805 | 32 | 0,04 | 1,28 |
| 18. Резонатор кварцевый SMD NX2520SA-16MHZ-STD-CSW-5 | 1 | 2,14 | 2,14 |
| 19. Резонатор кварцевый SMD 12.0 МГц KX-K | 1 | 1,46 | 1,46 |
| 20. Резонатор кварцевый SMD 16.0 МГц HC49SM | 1 | 1,72 | 1,72 |
| 21. Конденсатор SMD 0805 | 13 | 0,05 | 0,65 |
| 22. Конденсатор SMD 6861 | 3 | 0,32 | 0,96 |
| 23. Конденсатор выводной 22 мкФ | 3 | 0,05 | 0,15 |
| 24. Конденсатор выводной 1000 мФ | 1 | 0,26 | 0,26 |

Продолжение таблицы 7.2

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 25. Светодиод SMD FYLS-1206SRC | 2 | 0,14 | 0,28 |
| 26. Светодиод SMD GNL-1206UBC | 2 | 0,17 | 0,34 |
| 27. Линейный стабилизатор напряжения AMS1117-3.3В | 2 | 0,24 | 0,48 |
| 28. Диод Шоттки SS14 | 3 | 0,10 | 0,30 |
| 29. Предохранитель SMD 1206 | 1 | 0,25 | 0,25 |
| Итого |  | | 306,35 |
| Всего с учётом транспортных расходов (1,1)(Pм) | | | 336,98 |

Расчёт общей суммы прямых затрат на производство аппаратной части представлен в таблице 7.3.

Таблица 7.3 — Расчёт общей суммы прямых затрат на производство аппаратной части

|  |  |
| --- | --- |
| Показатель | Сумма, р. |
| 1. Сырье и материалы | 110,20 |
| 2. Покупные комплектующие изделия | 336,98 |
| Всего прямые затраты на производство аппаратной части | 447,18 |

Расчёт затрат на основную заработную плату разработчиков программной части комплекса представлен в таблице 7.4.

При расчёте заработной платы используется среднемесячная заработная плата в Республике Беларусь для сотрудников ИТ-отрасли. Премия не начисляется.

Основная зарплата определяется по формуле (7.3).

где – коэффициент премий и иных стимулирующих выплат;

*n* – категории исполнителей, занятых разработкой;

– часовой оклад плата исполнителя *i*-й категории, р.;

– трудоёмкость работ, выполняемых исполнителем *i*-й категории, ч.

Часовая заработная плата каждого исполнителя определяется путём деления его месячной заработной платы (оклад плюс надбавки) на количество рабочих часов в месяце (расчётная норма рабочего времени на 2023 г. Для 5-дневной недели составляет 168 ч по данным Министерства труда и социальной защиты населения на момент проведения расчётов).

Таблица 7.4 — Расчёт затрат на основную заработную плату команды разработчиков

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Категория разработчика | Месячный оклад, р. | Часовой оклад, р. | Трудоёмкость работ, ч | Итого, р. |
| 1. Инженер-программист | 2150,00 | 12,80 | 6 | 76,80 |
| 2. Инженер-системотехник | 1620,00 | 9,64 | 12 | 115,68 |
| 3. Инженер-отладчик | 1370,00 | 8,15 | 4 | 32,60 |
| Итого | | | | 225,08 |
| Премия и стимулирующие выплаты | | | | 0,00 |
| Всего затраты на основную заработную плату разработчиков | | | | 225,08 |

Дополнительная заработная плата определяется по формуле (7.4).

где Нд – норматив дополнительной зарплаты, 15%

Отчисления в фонд социальной защиты населения и обязательное страхование БелГосстрах (Зсз) определяется в соответствии с действующим законодательством по формуле (7.5)

где Нсоц – норматив отчислений в ФСЗН и Белгосстрах (в соответствии с действующим законодательством по состоянию на апрель 2023 г. – 35%)

Расчёт общей суммы затрат на разработку программной части программно-управляемого комплекса представлен в таблице 7.5.

Таблица 7.5 — Расчёт затрат на разработку программного средства

|  |  |  |
| --- | --- | --- |
| Наименование статье затрат | Формула/таблица для расчёта | Сумма, р. |
| 1. Основная заработная плата разработчиков | Табл. 7.4 | 225,08 |
| 2. Дополнительная заработная плата разработчиков | Формула (7.4) | 33,76 |
| 3. Отчисления на социальные нужды | Формула (7.5) | 90,59 |
| Затраты на разработку программной части | | 349,43 |

Формирование отпускной цены аппаратно-программного комплекса осуществляется в соответствии с методикой, представленной в таблице 7.6.

Таблица 7.6 – Методика формирования отпускной цены аппаратно-программного комплекса

|  |  |  |
| --- | --- | --- |
| Показатель | Формула/таблица для расчёта | Сумма, р. |
| 1. Затраты на производство аппаратной части | Табл. 7.3 | 447,18 |
| 2. Затраты на разработку программной части | Табл. 7.5 | 349,43 |
| 3. Сумма затрат на производство аппаратно-программного комплекса | 447,18 + 349,43 | 796,61 |
| 4. Накладные расходы | 796,61 ∙ 0,56 | 446,10 |
| 5. Расходы на реализацию | 796,61 ∙ 0,02 | 15,93 |
| 6. Полная себестоимость | 796,61 + 446,10 + 15,93 | 1258,64 |
| 7. Плановая прибыль, включаемая в цену | 1258,64 ∙ 0,25 | 314,66 |
| 8. Отпускная цена | 1258,64 + 314,66 | 1573,30 |

Результатом производства аппаратно-программного комплекса является прирост чистой прибыли, полученный от их реализации и рассчитываемый по формуле (7.6).

где – прогнозируемый годовой объем производства и реализации, шт.;

– прибыль, включаемая в цену, р.;

– ставка налога на прибыль согласно действующему законодательству,

% (по состоянию на апрель 2023 г. – 20 %).

= 314,66 ∙ 300 ∙ (1 0,20) = 75518,40 р.

**7.3 Расчёт инвестиций в проектирование и производство аппаратно-программного комплекса**

Инвестиции в производство аппаратно-программного комплекса включают в общем случае:

‒ инвестиции на его разработку;

‒ инвестиции в прирост основного капитала (затраты на приобретение необходимого для производства нового изделия оборудования, станков и т.п.);

‒ инвестиции в прирост собственного оборотного капитала (затраты на приобретение необходимых для производства нового изделия материалов, комплектующих, начатой, но незавершённой продукции и т.п.).

### **7.3.1 Расчёт инвестиций на разработку аппаратно-программного комплекса**

Инвестиции рассчитываем (Ир) по затратам на разработку нового изделия. Для начала необходимо рассчитать заработную плату технологам и инженерам предприятия-производителя, ввиду того что именно они являются первой статьёй первичных затрат.

Расчёт заработной платы разработчиков нового изделия в таблице 7.7.

Таблица 7.7 – Расчёт заработной платы разработчиков нового изделия

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Категория исполнителя | Численность исполнителей, чел. | Месячный оклад, р. | Дневной оклад, р. | Время, д. | Сумма, р. |
| 1. Руководитель проекта | 1 | 2350,00 | 111,90 | 52 | 5818,80 |
| 2. Инженер-программист | 1 | 2150,00 | 102,38 | 40 | 4095,20 |
| 3. Инженер-системотехник | 2 | 1620,00 | 77,14 | 52 | 8022,56 |
| 4. Инженер-отладчик | 1 | 1370,00 | 65,24 | 20 | 1304,80 |
| Итого |  | | | | 19241,36 |
| Премия и иные стимулирующие выплаты, 30% | | | | | 5772,41 |
| Всего основная заработная плата | | | | | 25013,77 |

Расчёт затрат на разработку нового изделия представлен в таблице 7.8.

Таблица 7.8 – Расчёт инвестиций на разработку нового изделия

|  |  |  |
| --- | --- | --- |
| Наименование статьи затрат | Формула/таблица для расчёта | Сумма, р. |
| 1. Основная заработная плата разработчиков Зо | Табл. 6.7 | 25013,77 |
| 2. Дополнительная заработная плата разработчиков | Формула (7.4) | 3752,07 |
| 3. Отчисления на социальные нужды | Формула (7.5) | 10068,04 |
| Инвестиции на разработку нового изделия | | 38833,88 |

### **7.3.2 Расчёт инвестиций в прирост основного капитала**

Инвестиции в прирост основного капитала не требуются, так как производство нового изделия планируется осуществлять на действующем оборудовании в связи с наличием на предприятии-производителе свободных производственных мощностей.

### **7.3.3 Расчёт инвестиций в прирост собственного оборотного капитала**

Расчёт инвестиций в прирост собственного оборотного капитала осуществляется следующим образом:

1. Определяется годовая потребность в материалах по формуле (7.7).

где – прогнозируемый годовой объем производства и реализации, шт.;

– затраты на материалы на единицу изделия, р.

2. Определяется годовая потребность в комплектующих изделиях по формуле (7.8).

где – прогнозируемый годовой объем производства и реализации, шт.;

– затраты на комплектующие изделия на единицу продукции, р.

3. Вычисляются инвестиции в прирост собственного оборотного капитала в процентах от годовой потребности в материалах и комплектующих изделиях – *β* (исходя из среднего уровня по экономике: 20‒30 %) по формуле (7.9).

Расчёт инвестиций в прирост собственного оборотного капитала представлен в таблице 7.9.

Таблица 7.9 – Расчёт инвестиций в прирост собственного оборотного капитала

|  |  |  |
| --- | --- | --- |
| Наименование статьи затрат | Формула/таблица для расчёта | Сумма, р. |
| 1. Годовая потребность в материалах | Формула (7.7) | 33060,00 |
| 2. Годовая потребность в комплектующих изделиях | Формула (7.8) | 101094,00 |
| Инвестиции в прирост собственного капитала | Формула (7.9) | 26830,80 |

**7.4 Расчёт показателей экономической эффективности инвестиций в проектирование и производство аппаратно-программного комплекса**

Оценка экономической эффективности разработки и производства нового изделия зависит от результата сравнения инвестиций в производство нового изделия (инвестиции в разработку и прирост собственных оборотных средств) и полученного годового прироста чистой прибыли.

Сумма инвестиций меньше суммы годового экономического эффекта, то есть инвестиции окупятся менее чем за год, оценка экономической эффективности инвестиций в производство нового изделия осуществляется на основе расчёта рентабельность инвестиций (затрат) по формуле (7.10).

где  – прирост чистой прибыли от производства и реализации новых изделий, р.;

, – инвестиции в разработку нового изделия и прирост собственного оборотного капитала, р.

Сравнивая инвестиции в разработку изделия и прирост собственных оборотных средств, с приростом годовой чистой прибыли можно сделать вывод, что инвестиции окупаются в течении года.

Таким образом, рентабельность инвестиций в разработку и производство данного комплекса превышает ставки по долгосрочным депозитам в белорусских рублях на момент расчётов, что свидетельствует об экономической эффективности инвестиций. Следовательно, производство аппаратно-программного комплекса «Беспилотный летательный аппарат с дистанционным управлением» является экономически эффективным, инвестиции в его разработку целесообразны.

# **ЗАКЛЮЧЕНИЕ**

В ходе работы над дипломным проектом была произведена работа по проектированию и разработке аппаратно-программного комплекса «Беспилотный летательный аппарат с дистанционным управлением».

В результате проделанной работы были рассмотрены существующие аналоги, что позволило разработать прототип контроллера полёта на основе микроконтроллера ATmega328P-AU. Дополнительно было спроектировано и реализовано устройство управления летательным аппаратом с использованием беспроводного интерфейса на 2.4 ГГц. Также была выполнена попытка разработки ПИД-регулятора и написано программное обеспечение для тестирования и эксплуатации реализованного комплекса.

Были проведены эксперименты с реализованной моделью ПИД-регулятора, выявлены недостатки и перспективы их устранения. Основная проблема заключается в наличии шумов и вибраций, создаваемых летательным аппаратом, которые влияют на работу математической модели ПИД-регулятора.

Для разработки данного комплекса использовался переносимый язык программирования AVR C и такие инструменты, как интегрированная среда разработки для микроконтроллеров Arduino IDE, среда автоматизации проектирования электроники Sprint Layout, система автоматизированного проектирования для промышленного дизайна Autodesk Fusion 360.

Проведя расчёт экономической эффективности, можно сделать вывод, что разработка и производство данного комплекса являются целесообразными, принесут выгоду как компании-разработчику, так и пользователю данной системы. Все инвестиции окупаются в течении первого года.

Основные достоинства комплекса:

− открытое программное обеспечение;

− возможность модернизации;

− небольшая итоговая стоимость относительно аналогов.

Возможные пути улучшения комплекса:

− добавление GPS-навигации;

− добавление трансляции видео в реальном времени;

− добавление управления через мобильные устройства;

− добавление поддержки различных модулей в программное обеспечение;

− улучшение программного обеспечения для более эффективного управления.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1] Вычислительные машины, системы и сети: дипломное проектирование (методическое пособие) [Электронный ресурс] : Минск БГУИР 2019. – Электронные данные. – Режим доступа: https://www.bsuir.by/m/12\_100229\_1\_136308.pdf – Дата доступа: 28.09.2022.

[2] Орехов С. Ю., Гайгеров М. А., Бородулин А. В. ИССЛЕДОВАНИЕ ПРИМЕНЕНИЯ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ И ИХ ОСНОВНЫХ ЭЛЕМЕНТОВ / Научно-образовательный журнал для студентов и преподавателей №6/2022 – С. 7062–7073.

[3] Мельниченко А. С. Анализ возможностей моделирования и применение регуляторов для управления квадрокоптером / А. С. Мельниченко, В. А. Шель, С. В. Кирильчик // Известия Южного федерального университета. Технические науки. – 2015. – №2 (163). – С. 40–49.

[4] Белоконь А.И., Золотухин Ю.Н., Котов К.Ю., Мальцев А.С., Нестеров А.А. Управление параметрами полёта квадрокоптера при движении по заданной траектории // Автометрия. – 2013. – № 4. – С. 32–42.

[5] Квадрокоптер MJX Bugs 7 с камерой 4K - MJX-B7 [Электронный ресурс]. – Электронные данные. – Режим доступа: https://mjx-russia.com/product/radioupravlyaemyy-skladnoy-kvadrokopter-mjx-bugs-7-s-kameroy-4k-mjx-b7 – Дата доступа: 05.10.2022

[6] Квадрокоптер DJI Mini 2 [Электронный ресурс]. – Электронные данные. – Режим доступа: https://dji-minsk.by/catalog/kvadrokoptery-dji/ kvadrokopter\_dji\_mini\_2/ – Дата доступа: 08.10.2022

[7] Квадрокоптер SYMA X56W-P [Электронный ресурс]. – Электронные данные. – Режим доступа: https://dronomania.ru/syma/x56w-p.html – Дата доступа: 11.10.2022

[8] ATMEGA328P-AU [Электронный ресурс]. – Электронные данные. – Режим доступа: https://ru.findic.com/compare/atmega328p-au-vs-atmega328p-mu-O8GoGX208.html – Дата доступа: 17.10.2022

[9] ATmega328/P datasheet [Электронный ресурс]. – Электронные данные. – Режим доступа: https://static.chipdip.ru/lib/549/DOC001549488.pdf – Дата доступа: 17.10.2022

[10] ATMEGA328-AU Intel / Altera [Электронный ресурс]. – Электронные данные. – Режим доступа: https://html.alldatasheet.com/html-pdf/392280/ATMEL/ATMEGA328-AU/155/1/ATMEGA328-AU.html – Дата доступа: 17.10.2022

[11] STM32F103 datasheet [Электронный ресурс]. – Электронные данные. – Режим доступа: https://static.chipdip.ru/lib/235/DOC000235733.pdf – Дата доступа: 19.10.2022

[12] ESP32-DOWDQ6-V3 [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.compel.ru/infosheet/ESPRES/ESP32-D0WDQ6-V3 – Дата доступа: 23.10.2022

[13] ESP32 Series Datasheet [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.espressif.com/sites/default/files/ documentation/esp32\_datasheet\_en.pdf – Дата доступа: 23.10.2022

[14] NRF24L01 Datasheet (PDF) – List of Unclassified Manufacturers [Электронный ресурс]. – Электронные данные. – Режим доступа: https://pdf1.alldatasheet.com/datasheet-pdf/view/1243924/ETC1/NRF24L01.html – Дата доступа: 28.10.2022

[15] Радиомодуль nRF24L01 [Электронный ресурс]. – Электронные данные. – Режим доступа: https://robototehnika.ru/content/article/radiomodul-nrf24l01-chast-1-obzor/ – Дата доступа: 28.10.2022

[16] HC-12 Datasheet [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.alldatasheet.com/datasheet-pdf/pdf/1242982/ETC1/ HC-12.html – Дата доступа: 30.10.2022

[17] Радиомодуль HC-12 [Электронный ресурс]. – Электронные данные. – Режим доступа: https://3d-diy.ru/wiki/arduino-moduli/radiomodul-hc-12/ – Дата доступа: 30.10.2022

[18] MPU-6050 GY-521 гироскоп и акселерометр [Электронный ресурс]. – Электронные данные. – Режим доступа: https://duino.ru/MPU-6050-giroskop-i-akselerometr.html – Дата доступа: 02.11.2022

[19] Десятиосный модуль положения GY-91 (MPU-9250 + BMP280) [Электронный ресурс]. – Электронные данные. – Режим доступа: https://ardmag.ru/arduino-sensors/2677.html – Дата доступа: 04.11.2022

[20] GY-91 10DOF гироскоп + акселерометр + магнитометр + барометр/термометр MPU-9250 [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.robostore.com.ua/gy-91-10dof-giroskop-akselerometr-magnitometr-barometr-termometr-mpu-9250/ – Дата доступа: 04.11.2022

[21] GY-9255 [Электронный ресурс]. – Электронные данные. – Режим доступа: https://radioled.com/image/catalog/pdf/GY-9255.pdf – Дата доступа: 07.11.2022

[22] PS-MPU-9255 datasheet [Электронный ресурс]. – Электронные данные. – Режим доступа: https://stanford.edu/class/ee267/misc/MPU-9255-Datasheet.pdf – Дата доступа: 07.11.2022

[23] Датчик давления и температуры BMP180 (Digital Pressure Sensor) [Электронный ресурс]. – Электронные данные. – Режим доступа: https://roboshop.spb.ru/index.php?route=product/product/download&product\_id=131&download\_id=225 – Дата доступа: 10.11.2022

[24] BMP280 datasheet [Электронный ресурс]. – Электронные данные. – Режим доступа: https://robot-kit.ru/wa-data/public/blog/download/BST-BMP280.pdf – Дата доступа: 13.11.2022

[25] Модуль GY-BMP280-3.3 датчик атмосферного давления и температуры [Электронный ресурс]. – Электронные данные. – Режим доступа: https://robot-kit.ru/3077/ – Дата доступа: 13.11.2022

[26] NEO-6 u-blox 6 GPS Modules datasheet [Электронный ресурс]. – Электронные данные. – Режим доступа: https://content.u-blox.com/ sites/default/ files/products/documents/NEO-6\_DataSheet\_%28GPS.G6-HW-09005%29.pdf – Дата доступа: 20.11.2022

[27] GPS-модуль NEO-6M [Электронный ресурс]. – Электронные данные. – Режим доступа: https://3d-diy.ru/wiki/arduino-moduli/gps-modul-neo-6m/ – Дата доступа: 23.11.2022

[28] ATGM336H-5N module Datasheet [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.icofchina.com/d/file/xiazai/2016-12-05/b5c57074f4b1fcc62ba8c7868 548d18a.pdf – Дата доступа: 27.11.2022

[29] ATGM336H, Модуль GPS UART 3.3-5V [Электронный ресурс]. – Электронные данные. – Режим доступа: https://umnaya-elektronika.ru/svyaz/gps/ atgm336h-modul-gps-uart-3.3-5v/ – Дата доступа: 29.11.2022

[30] Щеточные и бесщеточные двигатели квадракоптеров: свойства, различия, плюсы и минусы [Электронный ресурс]. – Электронные данные. – Режим доступа: https://dronnews.ru/o-dronakh/dvigateli-kvadrocopterov. html?utm\_referrer=https%3A%2F%2Fwww.google.com%2F – Дата доступа: 02.12.2022

[31] Бесколлекторный двигатель EMAX RS1408 2300KV-3600KV для гоночного квадроцикла Micro FPV [Электронный ресурс]. – Электронные данные. – Режим доступа: https://emaxmodel.com/products/emax-rs1408-2300kv-3600kv-brushless-motor-for-micro-fpv-racing-quad – Дата доступа: 08.12.2022

[32] Купить EMAX бесщеточный двигатель EMAX RS1408 [Электронный ресурс]. – Электронные данные. – Режим доступа: https://aliexpress.ru/item/32977111063.html?sku\_id=66771168187 – Дата доступа: 09.12.2022

[33] SKYSTARS KOKO 1404 [Электронный ресурс]. – Электронные данные. – Режим доступа: https://ru.banggood.com/SKYSTARS-KOKO-1404-3000KV-6S-3800KV-or-4600KV-4S-Brushless-Racing-Motor-For-Long-Rang-FPV-Racing-2\_5inch-4inch-Toothpick-RC-Drone-p-1897433.html?cur\_warehouse =CN& ID=517763 – Дата доступа: 11.12.2022

[34] Купить SKYSTARS KOKO 1404 3000KV [Электронный ресурс]. – Электронные данные. – Режим доступа: https://aliexpress.ru/item/ 1005002967385400.html?sku\_id=12000023003484511 – Дата доступа: 11.12.2022

[35] Бесколлекторные моторы HappyModel EX1102 9000KV [Электронный ресурс]. – Электронные данные. – Режим доступа: https://flymod.net/item/ex1102\_kv9000 – Дата доступа: 12.12.2022

[36] Купить HappyModel EX1102 [Электронный ресурс]. – Электронные данные. – Режим доступа: https://aliexpress.ru/item/33010772549.html ?sku\_id=67114365055 – Дата доступа: 13.12.2022

[37] Купить EMAX 1104 4300KV [Электронный ресурс]. – Электронные данные. – Режим доступа: https://aliexpress.ru/item/1005002190328136.html?sku\_id=12000019002391933 – Дата доступа: 14.12.2022

[38] Купить Бесщеточный двигатель LD-Power 1106 4200KV [Электронный ресурс]. – Электронные данные. – Режим доступа: https://aliexpress.ru/item/1005003385186703.html?sku\_id=12000025534470448 – Дата доступа: 15.12.2022

[39] Как выбрать регулятор скорости для квадрокоптера [Электронный ресурс]. – Электронные данные. – Режим доступа: https://blog.rcdetails.info/ kak-vybrat-regulyator-hoda-dlya-gonochnogo-kvadrokoptera/ – Дата доступа: 18.12.2022

[40] ESC регуляторы оборотов, что это, как они работают и как выбрать [Электронный ресурс]. – Электронные данные. – Режим доступа: https://profpv.ru/esc-regulyatory-oborotov-chto-ehto-kak-oni-r/ – Дата доступа: 19.12.2022

[41] Hobbywing XRotor Micro 45A 4-in-1 3-6S BLHeli\_S32 ESC [Электронный ресурс]. – Электронные данные. – Режим доступа: https://rcdrive.ru/unit.php?unit=29608 – Дата доступа: 24.12.2022

[42] Flycolor Raptor 30A BLS Pro [Электронный ресурс]. – Электронные данные. – Режим доступа: https://rcsearch.ru/-c1712?page=2 – Дата доступа: 26.12.2022

[43] Регулятор скорости 4 в 1 T-Motor Velox V45A V2 BLHeli\_32 Dshot 1200 [Электронный ресурс]. – Электронные данные. – Режим доступа: https://flymod.net/item/tmotor\_velox\_v45a – Дата доступа: 29.12.2022

[44] Архитектура микроконтроллера ATmega328P [Электронный ресурс]. – Электронные данные. – Режим доступа: https://ppt-online.org/149556 – Дата доступа: 04.01.2023

[45] Обзор модуля GY-521 (MPU-6050) [Электронный ресурс]. – Электронные данные. – Режим доступа: https://robotchip.ru/obzor-modulya-gy-521/ – Дата доступа: 12.01.2023

[46] Обзор датчика давления BMP180 (BMP080) [Электронный ресурс]. – Электронные данные. – Режим доступа: https://robotchip.ru/obzor-datchika-davleniya-bmp180/ – Дата доступа: 13.01.2023

[47] nRF24L01+ [Электронный ресурс]. – Электронные данные. – Режим доступа: https://micro-pi.ru/nrf24l01-spi-модуль-беспроводной-связи/ – Дата доступа: 18.01.2023

[48] Радио модуль NRF24L01 [Электронный ресурс]. – Электронные данные. – Режим доступа: https://3d-diy.ru/wiki/arduino-moduli/radio-modul-nrf24l01/ – Дата доступа: 20.01.2023

[49] Главное отличие цветных светодиодов [Электронный ресурс]. – Электронные данные. – Режим доступа: http://schip.com.ua/glavnoe-otlichie-czvetnyh-svetodiodov/ – Дата доступа: 25.01.2023

[50] Как выбрать регулятор скорости для гоночного квадрокоптера [Электронный ресурс]. – Электронные данные. – Режим доступа: https://rcdetails.info/kak-vybrat-regulyator-hoda-dlya-gonochnogo-kvadrokoptera/ – Дата доступа: 29.01.2023

[51] Добавление внешней памяти EEPROM в Ардуино [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.joyta.ru/13126-dobavlenie-vneshnej-pamyati-eeprom-v-arduino-24lc256/ – Дата доступа: 29.01.2023

[52] Расширитель GPIO MCP23017 [Электронный ресурс]. – Электронные данные. – Режим доступа: https://kotyara12.ru/iot/mcp23017/ – Дата доступа: 02.02.2023

[53] Микросхема CH340G – преобразователь интерфейса USB в UART (мост USB-UART) [Электронный ресурс]. – Электронные данные. – Режим доступа: http://mypractic.ru/mikrosxema-ch340g-preobrazovatel-interfejsa-usb-v-uart-most-usb-uart-xarakteristiki-usloviya-ekspluatacii-tipovye-sxemy-vklyucheniya – Дата доступа: 06.02.2023

[54] Подключение ST7735S к Arduino [Электронный ресурс]. – Электронные данные. – Режим доступа: https://arduino-ide.com/modules/32-podkljuchenie-st7735s-k-arduino.html – Дата доступа: 10.02.2023

[55] TP4056 - контроллере зарядки Li-ion аккумуляторов [Электронный ресурс]. – Электронные данные. – Режим доступа: http://beriled.biz/data/files/tp4056new.pdf – Дата доступа: 17.02.2023

[56] Интерфейсная шина (I2C) [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.rotr.info/electronics/interface/spi/spi.htm – Дата доступа: 23.02.2023

[57] SPI [Электронный ресурс]. – Электронные данные. – Режим доступа: https://easyelectronics.ru/interface-bus-iic-i2c.html – Дата доступа: 29.02.2023

[58] UART [Электронный ресурс]. – Электронные данные. – Режим доступа: https://habr.com/ru/articles/109395/ – Дата доступа: 04.03.2023

[59] Enhanced ShockBurst User Guide [Электронный ресурс]. – Электронные данные. – Режим доступа: https://infocenter.nordicsemi.com/index.

jsp?topic=%2Fcom.nordic.infocenter.sdk5.v12.0.0%2Fesb\_users\_guide.html – Дата доступа: 12.03.2023

[60] FSK, GFSK, MSK, GMSK и некоторые другие режимы [Электронный ресурс]. – Электронные данные. – Режим доступа: http://www.radioscanner.ru/info/article345/ – Дата доступа: 16.03.2023

[61] Широтно-импульсная модуляция (ШИМ) [Электронный ресурс]. – Электронные данные. – Режим доступа: https://ledjournal.info/spravochnik/shir

otno-impulsnaya-modulyaciya.html – Дата доступа: 21.03.2023

[62] USB Программатор UART CP2102 [Электронный ресурс]. – Электронные данные. – Режим доступа: https://iarduino.ru/shop/programmers/usb-programmator-uart-cp2102-podhodit-dlya-arduino-pro-mini.html – Дата доступа: 01.04.2023

[63] NRF24L01 Datasheet [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.mouser.com/datasheet/2/297/nRF24L01\_Product\_

Specification\_v2\_0-9199.pdf – Дата доступа: 04.04.2023

[64] CH340G USB to UART Interface Datasheet [Электронный ресурс]. – Электронные данные. – Режим доступа: http://images.100y.com.tw/pdf\_file/34-WCH-CH340G.pdf – Дата доступа: 15.04.2023

[65] Резистивные кнопки [Электронный ресурс]. – Электронные данные. – Режим доступа: http://arduino-kid.ru/blog/rezistivnye-knopki – Дата доступа: 21.04.2023

[66] Arduino библиотека Wire [Электронный ресурс]. – Электронные данные. – Режим доступа: https://all-arduino.ru/biblioteki-arduino/arduino-biblioteka-wire/ – Дата доступа: 28.04.2023

[67] SPI library [Электронный ресурс]. – Электронные данные. – Режим доступа: https://github.com/PaulStoffregen/SPI/blob/master/SPI.h – Дата доступа: 29.04.2023

[68] EEPROM Library [Электронный ресурс]. – Электронные данные. – Режим доступа: https://docs.arduino.cc/learn/built-in-libraries/eeprom – Дата доступа: 05.05.2023

[69] nRF24/RF24 [Электронный ресурс]. – Электронные данные. – Режим доступа: https://github.com/nRF24/RF24/blob/master/nRF24L01.h – Дата доступа: 06.05.2023

[70] i2cdevlib [Электронный ресурс]. – Электронные данные. – Режим доступа: https://github.com/jrowberg/i2cdevlib/blob/master/Arduino/I2Cdev/I2Cdev.h – Дата доступа: 06.05.2023

[71] mpu6050 library [Электронный ресурс]. – Электронные данные. – Режим доступа: https://github.com/ElectronicCats/mpu6050 – Дата доступа: 07.05.2023

[72] Adafruit-BMP085-Library [Электронный ресурс]. – Электронные данные. – Режим доступа: https://github.com/adafruit/Adafruit-BMP085-Library – Дата доступа: 07.05.2023

[73] LCDWIKI\_SPI [Электронный ресурс]. – Электронные данные. – Режим доступа: https://github.com/gitcnd/LCDWIKI\_SPI – Дата доступа: 08.05.2023

[74] Adafruit-MCP23017-Arduino-Library [Электронный ресурс]. – Электронные данные. – Режим доступа: https://github.com/adafruit/Adafruit-MCP23017-Arduino-Library – Дата доступа: 09.05.2023

[75] Arduino IDE [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.arduino.cc/en/software – Дата доступа: 12.05.2023

# **ПРИЛОЖЕНИЕ А**

*(обязательное)*

Код программы

Файл Flight\_controller.ino

001. #include <Wire.h>

002. #include <EEPROM.h>

003. #include <nRF24L01.h>

004. #include <RF24.h>

005.

006. //---------------------- LEDS -------------------------

007. #define RED\_LED\_PIN 3

008. #define GREEN\_LED\_PIN 4

009.

010. //---------------------- MPU6050 ----------------------

011. long acc\_x, acc\_y, acc\_z, acc\_total\_vector;

012. double gyro\_pitch, gyro\_roll, gyro\_yaw;

013.

014. double gyro\_axis\_cal[3];

015. int acc\_axis[3], gyro\_axis[3];

016.

017. int cal\_int, gyro\_address;

018. int temperature;

019.

020. float angle\_roll\_acc, angle\_pitch\_acc, angle\_pitch, angle\_roll;

021. float roll\_level\_adjust, pitch\_level\_adjust;

022. boolean gyro\_angles\_set;

023.

024. //---------------------- MOTORS -----------------------

025. int start = 0;

026.

027. unsigned long loop\_timer;

028. unsigned long timer\_fr, timer\_br, timer\_bl, timer\_fl, esc\_timer, esc\_loop\_timer;

029. int esc\_fr, esc\_br, esc\_bl, esc\_fl;

030.

031. //---------------------- EEPROM -----------------------

032. byte eeprom\_data[13];

033.

034. //---------------------- NRF24L01 ---------------------

035. static const uint8\_t PROGMEM CE\_NRF = 7; // pin CE модуля NRF24L01

036. static const uint8\_t PROGMEM CS\_NRF = 8; // pin CS модуля NRF24L01

037. static const uint64\_t pipe = 0xE8E8F0F0E1LL;

038.

039. RF24 radio(CE\_NRF, CS\_NRF);

040.

041. struct MyData {

042.   uint16\_t throttle; //газ

043.   uint16\_t yaw; //рысканье

044.   uint16\_t pitch; //тангаж

045.   uint16\_t roll; //крен

046.   uint16\_t AUX1;

047.   uint16\_t AUX2;

048. };

049. MyData data;

050.

051. int throttle, yaw, pitch, roll;

052.

053. //---------------------- PID --------------------------

054. float pid\_p\_gain\_roll = 0.2;

055. float pid\_i\_gain\_roll = 0.00;

056. float pid\_d\_gain\_roll = 0.12;

057. int pid\_max\_roll = 200;

058.

059. float pid\_p\_gain\_pitch = pid\_p\_gain\_roll;

060. float pid\_i\_gain\_pitch = pid\_i\_gain\_roll;

061. float pid\_d\_gain\_pitch = pid\_d\_gain\_roll;

062. int pid\_max\_pitch = pid\_max\_roll;

063.

064. float pid\_p\_gain\_yaw = pid\_p\_gain\_roll;

065. float pid\_i\_gain\_yaw = pid\_i\_gain\_roll;

066. float pid\_d\_gain\_yaw = pid\_d\_gain\_roll;

067. int pid\_max\_yaw = pid\_max\_roll;

068.

069. float pid\_error\_temp;

070. float pid\_i\_mem\_roll, pid\_roll\_setpoint, gyro\_roll\_input, pid\_output\_roll, pid\_last\_roll\_d\_error;

071. float pid\_i\_mem\_pitch, pid\_pitch\_setpoint, gyro\_pitch\_input, pid\_output\_pitch, pid\_last\_pitch\_d\_error;

072. float pid\_i\_mem\_yaw, pid\_yaw\_setpoint, gyro\_yaw\_input, pid\_output\_yaw, pid\_last\_yaw\_d\_error;

073.

074.

075. //---------------------- SETUP -------------------------

076. void setup() {

077.   //инициализация последовательной соединение на скорости 9600 бит/с

078.   Serial.begin(9600);

079.

080.   //инициализация светодиодов

081.   pinMode(RED\_LED\_PIN, OUTPUT);

082.   pinMode(GREEN\_LED\_PIN, OUTPUT);

083.

084.   //чтение данных из EEPROM

085.   for (start = 0; start <= 12; start++) eeprom\_data[start] = EEPROM.read(start);

086. start = 0; //устанавливаем значение старта в 0

087.

088.   //читаем адрес MPU6050

089. gyro\_address = eeprom\_data[5];

090.

091. //инициализация шины I2C

092. Wire.begin();

093. TWBR = 12; //скорость 400 кГц

094.

095. //устанавливаем пины с ESC как OUTPUT

096. DDRD |= B01000000; //PD6 - pin(6)

097. DDRD |= B00100000; //PD5 - pin(5)

098. DDRB |= B00000100; //PB2 - pin(10)

099. DDRB |= B00000010; //PB1 - pin(9)

100.

101.   //если данные в EEPROM некорректные - мигаем красным светодиодом

102.   while (eeprom\_data[10] != 'J' || eeprom\_data[11] != 'M' || eeprom\_data[12] != 'B') {

103.     digitalWrite(RED\_LED\_PIN, !digitalRead(RED\_LED\_PIN));

104. delay(200);

105. }

106.

107. //инициализация MPU6050

108. init\_mpu6050();

109.

110. //калибровка ESC

111. for (cal\_int = 0; cal\_int < 1250; cal\_int++) {

112. //подаем ШИМ сигнал длительностью 1000мс на ESC

113. PORTD |= B01000000; //PD6 - pin(6)

114. PORTD |= B00100000; //PD5 - pin(5)

115. PORTB |= B00000100; //PB2 - pin(10)

116. PORTB |= B00000010; //PB1 - pin(9)

117. delayMicroseconds(1000);

118.

119. //выключаем двигатели

120. PORTD &= B10111111; //PD6 - pin(6)

121. PORTD &= B11011111; //PD5 - pin(5)

122. PORTB &= B11111011; //PB2 - pin(10)

123. PORTB &= B11111101; //PB1 - pin(9)

124. delayMicroseconds(3000);

125. }

126.

127. //калибровка MPU6050 и ESC

128. for (cal\_int = 0; cal\_int < 2000 ; cal\_int ++) {

129. //мигаем зелёным светодиодом

130.     if (cal\_int % 15 == 0) digitalWrite(GREEN\_LED\_PIN, !digitalRead(GREEN\_LED\_PIN));

131.

132. //получаем данные с MPU6050

133. read\_mpu6050();

134. gyro\_axis\_cal[1] += gyro\_axis[1];

135. gyro\_axis\_cal[2] += gyro\_axis[2];

136. gyro\_axis\_cal[3] += gyro\_axis[3];

137.

138. //подаем ШИМ сигнал длительностью 1000мс на ESC

139. PORTD |= B01000000; //PD6 - pin(6)

140. PORTD |= B00100000; //PD5 - pin(5)

141. PORTB |= B00000100; //PB2 - pin(10)

142. PORTB |= B00000010; //PB1 - pin(9)

143. delayMicroseconds(1000);

144.

145. //выключаем двигатели

146. PORTD &= B10111111; //PD6 - pin(6)

147. PORTD &= B11011111; //PD5 - pin(5)

148. PORTB &= B11111011; //PB2 - pin(10)

149. PORTB &= B11111101; //PB1 - pin(9)

150. delay(3);

151. }

152. digitalWrite(GREEN\_LED\_PIN, LOW); delay(200);

153.

154. //высчитываем среднее смещение MPU6050

155. gyro\_axis\_cal[1] /= 2000;

156. gyro\_axis\_cal[2] /= 2000;

157. gyro\_axis\_cal[3] /= 2000;

158.

159. //ожидаем инициализацию NRF24L01

160. while (!radio.begin()) {

161. digitalWrite(RED\_LED\_PIN, !digitalRead(RED\_LED\_PIN));

162. delay(200);

163. }

164.

165. //настраиваем NRF24L01

166. init\_nrf24l01();

167.

168.   //ждем, пока скорость на передатчике не будет установлена на 0

169. while (1) {

170. //если есть данные для чтения - прочитать

171. if (radio.available()) {

172. radio.read(&data, sizeof(MyData));

173. throttle = map(data.throttle, 0, 258, 1000, 2000);

174. yaw = map(data.yaw, 0, 258, 1000, 2000);

175. pitch = map(data.pitch, 0, 258, 1000, 2000);

176. roll = map(data.pitch, 0, 258, 1000, 2000);

177.

178. //если скорость 0 - выходим из цикла

179. if (throttle == 1000) {

180. break;

181. }

182. }

183. start++;

184. delay(3);

185.

186. //мигаем красным светодиодом

187. if (start == 125) {

188. digitalWrite(RED\_LED\_PIN, !digitalRead(RED\_LED\_PIN));

189. start = 0;

190. }

191.

192. //подаем ШИМ сигнал длительностью 1000мс на ESC

193. PORTD |= B01000000; //PD6 - pin(6)

194. PORTD |= B00100000; //PD5 - pin(5)

195. PORTB |= B00000100; //PB2 - pin(10)

196. PORTB |= B00000010; //PB1 - pin(9)

197. delayMicroseconds(1000);

198.

199. //выключаем двигатели

200. PORTD &= B10111111; //PD6 - pin(6)

201. PORTD &= B11011111; //PD5 - pin(5)

202. PORTB &= B11111011; //PB2 - pin(10)

203. PORTB &= B11111101; //PB1 - pin(9)

204. delay(3);

205. }

206. start = 0;

207.

208. //выключаем все светодиоды

209. digitalWrite(GREEN\_LED\_PIN, LOW);

210. digitalWrite(RED\_LED\_PIN, LOW);

211.

212. //получаем текущее время (мс) для первого цикла

213. loop\_timer = micros();

214. }

215.

216.

217. //---------------------- LOOP -------------------------

218. void loop() {

219.   //вычисление входных параметров ПИД-регулятора на основе показаний MPU6050

220.   gyro\_roll\_input = (gyro\_roll\_input \* 0.7) + ((gyro\_roll / 65.5) \* 0.3);

221.   gyro\_pitch\_input = (gyro\_pitch\_input \* 0.7) + ((gyro\_pitch / 65.5) \* 0.3);

222.   gyro\_yaw\_input = (gyro\_yaw\_input \* 0.7) + ((gyro\_yaw / 65.5) \* 0.3);

223.

224. //расчет пройденного угла по гироскопу

225. angle\_pitch += gyro\_pitch \* 0.0000611;

226. angle\_roll += gyro\_roll \* 0.0000611;

227.

228.   //если устройство отклонилось по оси z - компенсируем углы противоположными

229. angle\_pitch -= angle\_roll \* sin(gyro\_yaw \* 0.000001066);

230. angle\_roll += angle\_pitch \* sin(gyro\_yaw \* 0.000001066);

231.

232. //вычисление полного вектора акселерометра

233.   acc\_total\_vector = sqrt((acc\_x \* acc\_x) + (acc\_y \* acc\_y) + (acc\_z \* acc\_z));

234.

235. //вычисление угла тангажа

236. if (abs(acc\_y) < acc\_total\_vector) {

237.     angle\_pitch\_acc = asin((float)acc\_y / acc\_total\_vector) \* 57.296;

238. }

239. //вычисление угла крена

240. if (abs(acc\_x) < acc\_total\_vector) {

241.     angle\_roll\_acc = asin((float)acc\_x / acc\_total\_vector) \* -57.296;

242. }

243.

244. //компенсация гироскопа акселерометром

245.   angle\_pitch = angle\_pitch \* 0.9996 + angle\_pitch\_acc \* 0.0004;

246.   angle\_roll = angle\_roll \* 0.9996 + angle\_roll\_acc \* 0.0004;

247.

248. //вычисление поправки углов x и y

249. pitch\_level\_adjust = angle\_pitch \* 15;

250. roll\_level\_adjust = angle\_roll \* 15;

251.

252. //если получены данные передатчика - прочитать

253. if (radio.available()) {

254. radio.read(&data, sizeof(MyData));

255. throttle = map(data.throttle, 0, 257, 1000, 2000);

256. yaw = map(data.yaw, 0, 257, 1000, 2000);

257. pitch = map(data.pitch, 0, 257, 1000, 2000);

258. roll = map(data.roll, 0, 257, 1000, 2000);

259.

260.     //для запуска моторов: скорость 0, правый джойстик нижний правый угол

261. if (throttle < 1020 && pitch > 1950 && roll > 1950) {

262. start = 1;

263. }

264.

265.     //разблокировка двигателей: возврат правого джойстика в центр и скорость 0

266.     if (start == 1 && throttle < 1020 && (pitch < 1550 && pitch > 1450) && (roll < 1550 && roll > 1450)) {

267.       digitalWrite(GREEN\_LED\_PIN, HIGH); //включаем зелёный светодиод

268. start = 2;

269.

270.       //при разблокировке двигателей сбрасываем предыдущие параметры ПИД

271. angle\_pitch = angle\_pitch\_acc;

272. angle\_roll = angle\_roll\_acc;

273.

274. pid\_i\_mem\_roll = 0;

275. pid\_last\_roll\_d\_error = 0;

276. pid\_i\_mem\_pitch = 0;

277. pid\_last\_pitch\_d\_error = 0;

278. pid\_i\_mem\_yaw = 0;

279. pid\_last\_yaw\_d\_error = 0;

280. }

281.

282.     //блокировка двигателей: скорость 0, правый джойстик нижний левый угол

283.     if (start == 2 && throttle < 1020 && pitch < 1050 && roll > 1950) {

284. start = 0;

285.       digitalWrite(GREEN\_LED\_PIN, LOW); //выключаем зелёный светодиод

286. }

287.

288.     //вычисляем желаемые углы на основе полученных от передатчика

289. pid\_roll\_setpoint = 0;

290. if (roll > 1508) pid\_roll\_setpoint = roll - 1508;

291. else if (roll < 1492) pid\_roll\_setpoint = roll - 1492;

292. pid\_roll\_setpoint -= roll\_level\_adjust;

293. pid\_roll\_setpoint /= 3.0;

294.

295. pid\_pitch\_setpoint = 0;

296. if (pitch > 1508)pid\_pitch\_setpoint = pitch - 1508;

297. else if (pitch < 1492)pid\_pitch\_setpoint = pitch - 1492;

298. pid\_pitch\_setpoint -= pitch\_level\_adjust;

299. pid\_pitch\_setpoint /= 3.0;

300.

301. pid\_yaw\_setpoint = 0;

302. if (yaw > 1050) {

303. if (yaw > 1508)pid\_yaw\_setpoint = (yaw - 1508) / 3.0;

304.       else if (yaw < 1492)pid\_yaw\_setpoint = (yaw - 1492) / 3.0;

305. }

306.

307. //вычисляем ПИД

308. calculate\_pid();

309. }

310.

311. //если двигатели разблокированы и скорость не нулевая

312. if (start == 2 && throttle > 1020) {

313. //ограничиваем скорость для корректировки

314.      if (throttle > 1950) throttle = 1800;

315.      esc\_fl = throttle - pid\_output\_pitch + pid\_output\_roll - pid\_output\_yaw; //Скорость с поправками PD6 - pin(6)

316.     esc\_fr = throttle - pid\_output\_pitch - pid\_output\_roll + pid\_output\_yaw; //Скорость с поправками PB2 - pin(10)

317.     esc\_bl = throttle + pid\_output\_pitch + pid\_output\_roll + pid\_output\_yaw; //Скорость с поправками PD5 - pin(5)

318.     esc\_br = throttle + pid\_output\_pitch - pid\_output\_roll - pid\_output\_yaw; //Скорость с поправками PB1 - pin(9)

319.

320. //ограничиваем длительность ШИМ сигнала до 2000мс

321. if (esc\_fr > 2000) esc\_fr = 2000;

322. if (esc\_fl > 2000) esc\_fl = 2000;

323. if (esc\_br > 2000) esc\_br = 2000;

324. if (esc\_bl > 2000) esc\_bl = 2000;

325.

326. //если двигатели не запущены - устанавливаем 0 скорость

327. } else {

328. esc\_fr = 1000;

329. esc\_fl = 1000;

330. esc\_br = 1000;

331. esc\_bl = 1000;

332. }

333.

334.   //ждем пока пройдет 4000 мс цикла. Если превышено - включаем красный светодиод (ошибка)

335.   if (micros() - loop\_timer > 4050) digitalWrite(RED\_LED\_PIN, HIGH);

336. while (micros() - loop\_timer < 4000);

337. loop\_timer = micros();

338.

339. //начинаем подавать ШИМ-сигнал на ESC

340. PORTD |= B01000000; //PD6 - pin(6)

341. PORTD |= B00100000; //PD5 - pin(5)

342. PORTB |= B00000100; //PB2 - pin(10)

343. PORTB |= B00000010; //PB1 - pin(9)

344.

345. //вычисляем длительность ШИМ сигнала каждого двигателя

346. timer\_fr = esc\_fr + loop\_timer;

347. timer\_br = esc\_br + loop\_timer;

348. timer\_bl = esc\_bl + loop\_timer;

349. timer\_fl = esc\_fl + loop\_timer;

350.

351. //читаем данные с MPU6050

352. read\_mpu6050();

353.

354.   //пересдаем подавать ШИМ сигнал когда выйдет время

355. esc\_loop\_timer = micros();

356.   while (timer\_fr > esc\_loop\_timer || timer\_br > esc\_loop\_timer || timer\_bl > esc\_loop\_timer || timer\_fl > esc\_loop\_timer) {

357. esc\_loop\_timer = micros();

358.     if (timer\_fr <= esc\_loop\_timer) PORTB &= B11111011; //PB2(10)

359.     if (timer\_br <= esc\_loop\_timer) PORTB &= B11111101; //PB1(9)

360.     if (timer\_bl <= esc\_loop\_timer) PORTD &= B11011111; //PD5(5)

361.     if (timer\_fl <= esc\_loop\_timer) PORTD &= B10111111; //PD6(6)

362. }

363. }

364.

365. //метод инициализирует MPU6050

366. //установка всех регистров

367. void init\_mpu6050() {

368.

369. //установка регистра PWR\_MGMT\_1 для активации модуля

370. Wire.beginTransmission(gyro\_address);

371. Wire.write(0x6B);

372. Wire.write(0x00);

373. Wire.endTransmission();

374.

375. //установка регистра GYRO\_CONFIG для диапазона 500dps

376. Wire.beginTransmission(gyro\_address);

377. Wire.write(0x1B);

378. Wire.write(0x08);

379. Wire.endTransmission();

380.

381. //установка регистра ACCEL\_CONFIG для точности +/- 8g

382. Wire.beginTransmission(gyro\_address);

383. Wire.write(0x1C);

384. Wire.write(0x10);

385. Wire.endTransmission();

386.

387. //проверка правильности установки регистра GYRO\_CONFIG

388. Wire.beginTransmission(gyro\_address);

389. Wire.write(0x1B);

390. Wire.endTransmission();

391. Wire.requestFrom(gyro\_address, 1);

392. while (Wire.available() < 1);

393.

394.   //регистр некорректный - цикл, мигаем красным светодиодом

395. if (Wire.read() != 0x08) {

396. while (1) {

397. digitalWrite(RED\_LED\_PIN, !digitalRead(RED\_LED\_PIN));

398. delay(200);

399. }

400. }

401.

402. //установка регистра CONFIG для н.ч. фильтра на 43 Гц

403. Wire.beginTransmission(gyro\_address);

404. Wire.write(0x1A);

405. Wire.write(0x03);

406. Wire.endTransmission();

407. }

408.

409.

410. //метод инициализирует NRF24L01

411. void init\_nrf24l01() {

412. //отключение автоматического подтверждения

413. radio.setAutoAck(0);

414.   //задаем одну попытку для передачи и ожидание в 3750 мс

415. radio.setRetries(0, 15);

416. //задаем размер данных

417. radio.setPayloadSize(sizeof(MyData));

418. //выбираем 60й канал

419. radio.setChannel(0x60);

420. //устанавливаем скорость передачи в 1 Мбит/сек

421. radio.setDataRate(RF24\_1MBPS);

422. //устанавливаем максимальную мощность передатчика

423. radio.setPALevel(RF24\_PA\_HIGH);

424. //задаем 1 канал для передачи по адресу PIPE

425. radio.openReadingPipe(1, pipe);

426. //включение радиомодуля и инициализация настроек

427. radio.powerUp();

428. //переключаем модуль в режим приёма данных

429. radio.startListening();

430. }

431.

432. //метод получает данные с MPU6050

433. void read\_mpu6050() {

434. //заправшиваем 14 байт данных

435. Wire.beginTransmission(gyro\_address);

436. Wire.write(0x3B);

437. Wire.endTransmission();

438. Wire.requestFrom(gyro\_address, 14);

439.

440. //ожидаем, когда будет получено 14 байт

441. while (Wire.available() < 14);

442.

443. //читаем 7 чисел по 2 байта

444. acc\_axis[1] = Wire.read() << 8 | Wire.read();

445. acc\_axis[2] = Wire.read() << 8 | Wire.read();

446. acc\_axis[3] = Wire.read() << 8 | Wire.read();

447. temperature = Wire.read() << 8 | Wire.read();

448. gyro\_axis[1] = Wire.read() << 8 | Wire.read();

449. gyro\_axis[2] = Wire.read() << 8 | Wire.read();

450. gyro\_axis[3] = Wire.read() << 8 | Wire.read();

451.

452. //если калибровка была выполнена - делаем поправку

453. if (cal\_int == 2000) {

454. gyro\_axis[1] += gyro\_axis\_cal[1];

455. gyro\_axis[2] += gyro\_axis\_cal[2];

456. gyro\_axis[3] += gyro\_axis\_cal[3];

457. }

458.

459. //задание правильных направлений модуля по данным EEPROM

460. gyro\_roll = gyro\_axis[eeprom\_data[1] & 0b00000011];

461. if (eeprom\_data[1] & 0b10000000)gyro\_roll \*= -1;

462. gyro\_pitch = gyro\_axis[eeprom\_data[2] & 0b00000011];

463. if (eeprom\_data[2] & 0b10000000)gyro\_pitch \*= -1;

464. gyro\_yaw = gyro\_axis[eeprom\_data[3] & 0b00000011];

465. if (eeprom\_data[3] & 0b10000000)gyro\_yaw \*= -1;

466.

467. acc\_x = acc\_axis[eeprom\_data[2] & 0b00000011];

468. if (eeprom\_data[2] & 0b10000000)acc\_x \*= -1;

469. acc\_y = acc\_axis[eeprom\_data[1] & 0b00000011];

470. if (eeprom\_data[1] & 0b10000000)acc\_y \*= -1;

471. acc\_z = acc\_axis[eeprom\_data[3] & 0b00000011];

472. if (eeprom\_data[3] & 0b10000000)acc\_z \*= -1;

473. }

474.

475. //метод производит вычисление всех ПИД-регуляторов

476. void calculate\_pid() {

477. //PID крена

478. pid\_error\_temp = gyro\_roll\_input - pid\_roll\_setpoint;

479. pid\_i\_mem\_roll += pid\_i\_gain\_roll \* pid\_error\_temp;

480.   if (pid\_i\_mem\_roll > pid\_max\_roll) pid\_i\_mem\_roll = pid\_max\_roll;

481.   else if (pid\_i\_mem\_roll < pid\_max\_roll \* -1) pid\_i\_mem\_roll = pid\_max\_roll \* -1;

482.

483.   pid\_output\_roll = pid\_p\_gain\_roll \* pid\_error\_temp + pid\_i\_mem\_roll + pid\_d\_gain\_roll \* (pid\_error\_temp - pid\_last\_roll\_d\_error);

484.   if (pid\_output\_roll > pid\_max\_roll)pid\_output\_roll = pid\_max\_roll;

485.   else if (pid\_output\_roll < pid\_max\_roll \* -1) pid\_output\_roll = pid\_max\_roll \* -1;

486.

487. pid\_last\_roll\_d\_error = pid\_error\_temp;

488.

489. //PID тангажа

490. pid\_error\_temp = gyro\_pitch\_input - pid\_pitch\_setpoint;

491. pid\_i\_mem\_pitch += pid\_i\_gain\_pitch \* pid\_error\_temp;

492.   if (pid\_i\_mem\_pitch > pid\_max\_pitch)pid\_i\_mem\_pitch = pid\_max\_pitch;

493.   else if (pid\_i\_mem\_pitch < pid\_max\_pitch \* -1)pid\_i\_mem\_pitch = pid\_max\_pitch \* -1;

494.

495.   pid\_output\_pitch = pid\_p\_gain\_pitch \* pid\_error\_temp + pid\_i\_mem\_pitch + pid\_d\_gain\_pitch \* (pid\_error\_temp - pid\_last\_pitch\_d\_error);

496.   if (pid\_output\_pitch > pid\_max\_pitch)pid\_output\_pitch = pid\_max\_pitch;

497.   else if (pid\_output\_pitch < pid\_max\_pitch \* -1)pid\_output\_pitch = pid\_max\_pitch \* -1;

498.

499. pid\_last\_pitch\_d\_error = pid\_error\_temp;

500.

501. //PID рысканья

502. pid\_error\_temp = gyro\_yaw\_input - pid\_yaw\_setpoint;

503. pid\_i\_mem\_yaw += pid\_i\_gain\_yaw \* pid\_error\_temp;

504.   if (pid\_i\_mem\_yaw > pid\_max\_yaw)pid\_i\_mem\_yaw = pid\_max\_yaw;

505.   else if (pid\_i\_mem\_yaw < pid\_max\_yaw \* -1)pid\_i\_mem\_yaw = pid\_max\_yaw \* -1;

506.

507.   pid\_output\_yaw = pid\_p\_gain\_yaw \* pid\_error\_temp + pid\_i\_mem\_yaw + pid\_d\_gain\_yaw \* (pid\_error\_temp - pid\_last\_yaw\_d\_error);

508.   if (pid\_output\_yaw > pid\_max\_yaw)pid\_output\_yaw = pid\_max\_yaw;

509.   else if (pid\_output\_yaw < pid\_max\_yaw \* -1)pid\_output\_yaw = pid\_max\_yaw \* -1;

510.

511.   pid\_last\_yaw\_d\_error = pid\_error\_temp;

512. }

# **ПРИЛОЖЕНИЕ Б**

*(обязательное)*

Спецификация

# **ПРИЛОЖЕНИЕ В**

*(обязательное)*

Перечень элементов

# **ПРИЛОЖЕНИЕ Г**

*(обязательное)*

Ведомость документов