

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Санкт-Петербургский политехнический университет Петра Великого»
Институт компьютерных наук и технологий
Кафедра «Системы и технологии управления»

Работа допущена к защите

Заведующий кафедрой СТУ

_____ В.П. Шкодырев

«__» _____ 2016 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Система управления квадрокоптером на базе платформы Raspberry Pi
направление 27.04.04 – Управление в технических системах
магистерская программа 27.04.04_02 «Автоматизация технологических процессов
и производств»

Выполнил

студент гр. 63503/2

<

>

И.А. Смирнов

Научный руководитель

к.т.н., доц.

<

>

С.А. Ундаров

Санкт-Петербург

2016

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего
образования

«Санкт-Петербургский политехнический университет Петра Великого»
Институт компьютерных наук и технологий
Кафедра Системы и технологии управления

УТВЕРЖДАЮ
Зав. кафедрой СТУ д.т.н., профессор
_____ В.П. Шкодырев

«__» _____ 2016 г.

ЗАДАНИЕ

на выполнение магистерской диссертации

Институт компьютерных наук и технологий
по направлению 27.04.04 – Управление в технических системах
магистерская программа 27.04.04_02 «Автоматизация технологических процессов
и производств»

студента _____ Смирнова Ивана Александровича _____

1. Тема магистерской диссертации: _____ Система управления квадрокоптером на
базе платформы Raspberry Pi _____

2. Срок сдачи студентом законченной работы:

« 01 » июня 2016 г

3. Исходные данные к магистерской диссертации:

Литература по прилагаемому списку

Научные и практические материалы по системам управления квадрокоптерами

Содержание расчётно-пояснительной записки (перечень подлежащих разработке вопросов):

- 1) Обзор существующих типов беспилотных летательных аппаратов и их систем управления
- 2) Структурно-параметрический синтез системы управления на базе Raspberry Pi
- 3) Разработка алгоритма обработки данных, поступающих по радиоканалу, и реализация его в виде модуля ядра Linux
- 4) Исследование свойств разрабатываемой системы управления полетом и оценка её характеристик

5. Перечень графического материала (с точным указанием обязательных иллюстраций):

- 1) Структурная схема системы управления
- 2) Временные диаграммы канала ручного управления
- 3) Алгоритм обработки данных

6. Консультанты по магистерской диссертации:

7. Дата выдачи задания «__» _____ 201__ г.

Руководитель,
к.т.н., доц. Ундаров С.А.

подпись

(_____)

расшифровка

Задание принял к исполнению «__» _____ 20__ г.

Студент Смирнов И.А.

подпись

(_____)

расшифровка

Реферат

Диссертация, темой которой является "Система управления квадрокоптером на базе платформы Raspberry Pi" содержит 62 страницы, 3 таблиц, 16 рисунков, использовано 13 источников литературы. Приложение состоит из 4 листов.

Настоящая диссертация посвящена разработке метода обработки данных, поступающих с пульта по радиоканалу, с использованием контроллера прямого доступа к памяти. В частности рассматривается применение метода в полетном контроллере Navio+, функционирующего на базе одноплатного компьютера Raspberry Pi.

Приводится обзор существующих полетных контроллеров. Сформулированы требования к системе управления квадрокоптером.

Разработан алгоритм обработки данных при ручном управлении квадрокоптером. Программная реализация представляет собой модуль ядра Linux, который выполняет настройку контроллера прямого доступа к памяти, его запуск и обработку полученных массивов данных.

Abstract

This thesis, whose theme is "Quadcopter control system based on Raspberry Pi" contains 62 pages, 3 tables, 16 drawings, used 13 literature sources. The application consists of 4 pages.

This thesis is devoted to the development of the data processing method from the remote over the air, using the direct memory access controller. In particular, application of the method is seen in the flight controller Navio +, functioning on the basis of Raspberry Pi single-board computer.

A review of the existing flight controllers. Requirements to quadcopters management system.

A data processing algorithm for manual operation quadcopters. Software implementation is a Linux kernel module that adjusts to the memory controller directly access its launch and processing of the data sets.

Оглавление

Введение.....	8
1 Применение беспилотных летательных аппаратов.....	12
1.1 Виды БЛА.....	12
1.2 Первые квадрокоптеры.....	14
1.3 Сферы применения квадрокоптеров.....	17
1.4 Возможности квадрокоптеров.....	21
1.4.1 Базовые возможности.....	21
1.4.2 Возможности при расширенной аппаратной базе.....	22
1.4.3 Возможности при введении дополнительных программных компонентов	23
1.5 Выводы.....	24
2 Структурно-параметрически синтез системы управления на базе raspberry Pi...25	
2.1 Наиболее распространенные полетные контроллеры.....	25
2.2 Сравнение спецификаций контроллеров.....	26
2.3 Одноплатный компьютер Raspberry Pi.....	28
2.4 NAVIO+.....	30
2.4.1 ШИМ генератор.....	31
2.4.2 Инерциальное измерительное устройство.....	32
2.4.3 Приёмник GPS.....	32
2.4.4 Аналого-цифровой преобразователь.....	33
2.4.5 Модуль FRAM памяти.....	33
2.5 Программное обеспечение Raspberry Pi.....	33
2.6 Функциональное назначение полётных контроллеров.....	34
2.7 Полётные параметры ручного режима системы управления.....	36
2.8 Выводы.....	38
3 Разработка алгоритма работы системы ручного управления и программная реализация.....	40

3.1 Описание процесса преобразования команд пилота в задание на ШИМ генератор.....	40
3.2 Программная реализация.....	43
3.2.1 Сведения об операционной системе.....	43
3.2.2 Программирование модуля ядра.....	44
3.2.3 Размещение и применение драйвера-модуля ядра.....	49
3.3 Выводы.....	50
4. Тестирование разработанного модуля.....	51
4.1 Испытательный стенд.....	51
4.2 Программа и методика испытаний.....	53
4.2.2 Использование механизма прерываний.....	53
4.2.3 Использование модуля ядра.....	55
4.3 Выводы.....	56
Заключение.....	58
Список литературы.....	59
Приложение А.....	60
Приложение Б.....	62

Современные технологии развиваются с очень высокой скоростью. Одна их часть адаптируется для выполнения определенных работ и задач, другая – адаптирует методы достижения поставленных целей. Таким образом, с каждым годом совершенствуются подходы к реализации алгоритмов, обеспечивающих стабильное функционирование промышленности, высокотехнологичный уровень жизни человека и регулярную исследовательскую деятельность.

Имеются технологические процессы, в которых элементом системы автоматизации являются БПЛА. Помимо эффективности использования таких аппаратов для переноса фото и видео оборудования, это связано также с успешностью их внедрения для выполнения сложных технологических процессов и операций, таких как фотограмметрия фасадов зданий, инспектирование мостов, линий электропередач и других инфраструктур, орошение полей, съемка местности для составления 3D карт. Для выполнения вышеуказанных работ наибольшее распространение в последние годы получили квадрокоптеры — летательные аппараты с четырьмя несущими винтами.

В настоящее время управление полетом осуществляется в автоматическом режиме с использованием навигации по опорным точкам или в дистанционном режиме с помощью пульта управления. Наряду с этим существенно возрастает роль программного управления БПЛА на базе интеллектуальных автопилотов. Это связано с мировой тенденцией увеличения уровня автономности БПЛА при решении поставленных целевых задач, таких как планирование и автоматическое управление полетом по заданной траектории.

Реализация вышеупомянутых режимов осуществляется с помощью полетных контроллеров, которые управляют системой генерирования подъемной силы роторов (двигателя/винта/пропеллера), регулируют положение квадрокоптера в соответствии с выбранным режимом полета и обеспечивают обмен навигационными данными с наземной станцией управления.

Управление квадрокоптером полностью в ручном режиме нетривиальная задача. Для управления необходимо анализировать положение аппарата в пространстве и вырабатывать четыре управляющих воздействия на моторы в реальном времени. В связи с этим разработка систем автоматического управления (САУ) квадрокоптером, обеспечивающих стабилизацию положения во время полета, является актуальной задачей, так как это существенно облегчит оператору управление квадрокоптером. Оператор должен будет назначить цель и конечную ориентацию, а БПЛА самостоятельно ее достигнет.

Полетный контроллер представляют собой устройство, обладающее определенным набором программируемых элементов и датчиков, совокупность которых обеспечивает функциональность летательного аппарата. Современный полетный контроллер должен иметь вычислительное устройство (микроконтроллер), инерциальный датчик положения, в состав которого входят гироскоп, акселерометр и магнетометр, GPS приемник и система интерфейсов ввода-вывода для работы с периферией.

На рынке электроники для беспилотных летательных аппаратов существует ряд готовых решений, которые могут быть интегрированы в объект управления. Но эти решения ограничены набором программ, предоставляемым производителем, что сужает круг решаемых задач тем или иным полетным контроллером, так как невозможно внести изменения в алгоритм функционирования.

Однако существует более гибкая система, которая позволяет пользователю применять полетный контроллер не только для полетов, но и в качестве платформы для реализации собственных идей в области управления. Речь идет об автопилоте Navio+, функционирующем на основе одноплатного компьютера Raspberry Pi. Navio+ представляет собой плату с комплектом датчиков и интерфейсов, необходимых для управления беспилотным аппаратом, а всю вычислительную часть на себя берет Raspberry Pi, функционирующий под управлением операционной системы на основе ядра Linux. Таким образом с данного рода полетным контроллером можно взаимодействовать как с системой с открытым исходным

кодом, что дает возможность изменять программы управления беспилотником на свое усмотрение.

Тем не менее, эта система не лишена недостатков. Режим ручного управления с пульта по радиоканалу требует наличия аппаратных средств для обработки информации, поступающей от оператора. В случае с Navio+ и Raspberry Pi эту задачу приходится решать программно, так как эта система не имеет необходимого аппаратного обеспечения. В условиях ограниченных аппаратных ресурсов всю работу на себя берет центральный процессор Raspberry Pi, которому необходимо решать задачу обработки задания оператора и информации с датчиков, формирования целеуказаний на двигатели, запись информации в журнал полета (лог файл), и ряд других задач в зависимости от конкретной цели полета. Также центральному процессору придется обрабатывать прерывания, вызванные поступающим от радиоприемника сигналом, чтобы производить его обработку. Но обработкой прерываний невозможно получить точные данные о поступающем сигнале, так как возникают задержки в операционной системе, вносящие случайную ошибку в измерения.

Цель настоящей работы заключается в разработке метода точной обработки данных целеуказания, поступающих по радиоканалу.

Задачи, которые решаются в процессе выполнения диссертации, можно сформулировать так:

- обосновать необходимость замены существующего метода обработки данных, поступающих от радиоприемника к автопилоту Navio+;
- разработать подход, обеспечивающий более эффективное использование аппаратных ресурсов Raspberry Pi;
- разработать модуль ядра Linux, реализующий задачи по обработке данных;
- проанализировать работу системы управления после внесения изменений.

Разрабатываемый метод может быть применим далее в системах, функционирующих в условиях ограниченных аппаратных ресурсов при решении подобных задач.

1 Применение беспилотных летательных аппаратов

1.1 Виды БЛА

Беспилотным летательным аппаратом (БПЛА или БЛА) Летательный аппарат пилотируемые дистанционно, или выполняющий полёт автономно, без помощи пилота. Также их называют беспилотниками или дронами. БЛА до недавнего времени имели лишь военное предназначение, и возможность их использования была только у армии. Беспилотники выполняли задачи по аэросъемке (фото, видео), радиоразведке, обнаружению объектов и пр. Однако за последнее десятилетие сфера разработки и создания беспилотных систем вышла за эти рамки, и в настоящее время БЛА применяются по различным гражданским направлениям.

По виду и области выполняемых задач дроны делятся на 3 основных типа:

- беспилотные самолеты;
- беспилотные вертолеты;
- беспилотники мультироторного типа.

Беспилотные самолеты способны преодолевать большие расстояния, выполняя сложнейшую аэросъемку практически при любых метеоусловиях. Максимальное качество работы и эффективность выполняемых задач возможны на удалении не более 70 км от наземной станции управления. Во время полета требуется поддержание высокой скорости (до 400 км/час). Время нахождения в полете: от 30 минут до 8 часов. К недостаткам можно отнести сложность запуска и посадки. Запуск необходимо осуществлять, используя специальное устройство (катапульту), чтобы придать аппарату начальную скорость. Посадка требует посадочной полосы или выполняется с использованием парашюта, что в значительной мере затрудняет обеспечение точной посадки. Плюс ко всему, посадка на парашютной системе вызывает возникновение перегрузок, негативно сказывающихся на незащищенных элементах полезной нагрузки и фотооборудовании.

Беспилотные вертолеты не требуют специальных устройств для взлета или взлетно-посадочных полос. В отличие от самолетов они чуть более прихотливы к

погодным условиям. Время полета: от 30 минут до 3 часов. Конструкция таких БЛА сложна по сравнению с самолетами или мультироторными аппаратами, так как требует наличия сложного автомата перекоса лопастей основного и хвостового винтов. Важным преимуществом вертолетов перед другими типами летательных аппаратов является наличие режима самовращения несущего винта (авторотации), что может значительно снизить ущерб от падения при отказе двигателя.

Мультироторные беспилотные аппараты (мультикоптеры) являются многомоторными и имеют несколько несущих винтов (роторов). Как правило, конструкция таких дронов обладает 3, 4, 6, 8 или 12 винтами. Как и вертолеты, мультикоптеры обладают способностью вертикального старта (не требуют дополнительных устройств для запуска) и способны зависать в воздухе с нулевой скоростью, но более просты в управлении по сравнению с ними. Мультироторные БЛА не содержат такие конструктивно сложные элементы, как, например, автомат перекоса, и в связи с этим имеют более низкую стоимость ремонта, являются более надежными, а так же относительно недороги. Серьезным недостатком такого типа беспилотников является практически полная потеря управляемости при выходе из строя одного из винтов. Однако в настоящее время ведутся исследования и проводятся эксперименты по обеспечению безопасной посадки или даже продолжения движения шести- и восьмироторных аппаратов в такой ситуации.

Наибольшее распространение получила мультироторная конструкция с четырьмя вентиляно-моторными группами, каждая из которых состоит из двигателя и винта с постоянным шагом (углом наклона лопастей). Такие дроны компактны, просты в сборке и настройке, имеют относительно невысокую стоимость и расход энергии по сравнению с мультикоптерами с количеством винтов больше 4.

1.2 Первые квадрокоптеры

Мультикоптеры, или же многовинтовые вертолеты (как их было принято тогда называть), разрабатывались еще в те времена, когда проектировались первые вертолеты, но реализация таких конструкций была намного сложнее, чем обычных вертолетов, поскольку сложной задачей являлась реализация трансмиссии, которая могла бы от одного мотора передавать крутящий момент сразу нескольким роторам. При этом надо либо дифференцированно подавать крутящий момент на роторы, либо делать управляющие рули под роторами. Поэтому изобретение хвостового ротора вертолета и автомата перекоса вынудило забросить это направление как неперспективное на тот момент.

Первым действующим прототипом квадрокоптера (и заодно всех ныне существующих вертолетов) был вертолет Ботезата, поднявшийся в воздух в 1922 году [1].

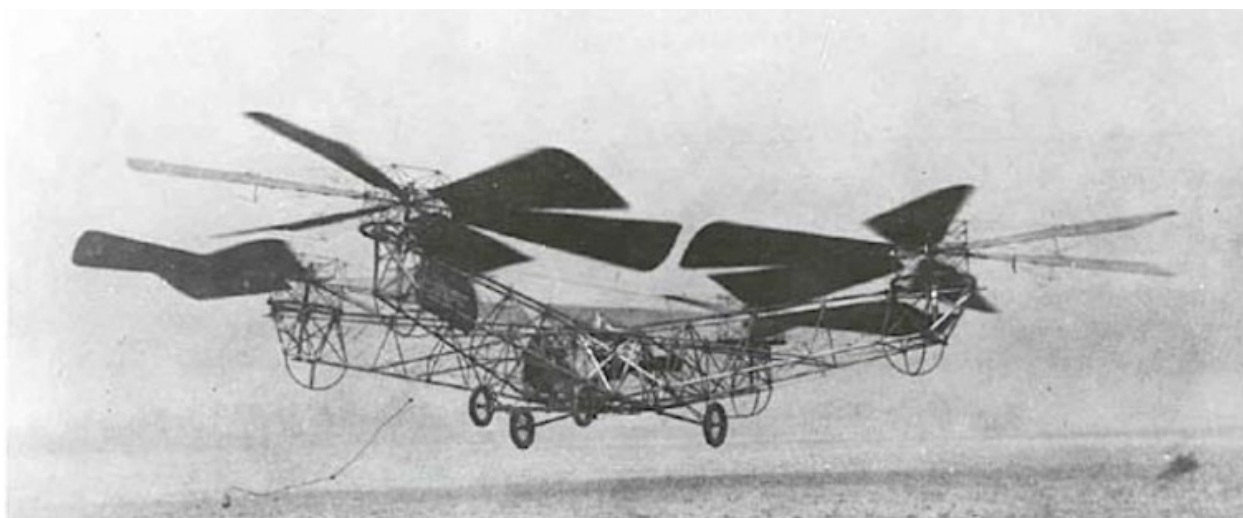


Рис. 1.2.1. Квадрокоптер Ботезата 1923г

В декабре 1922 года в Дайтоне начались полеты геликоптера Г. Ботезата, бывшего профессора Петроградского технологического института. Ботезат обратился к многовинтовой схеме: дифференциальное, отдельное управление тягой винтов сулило надежное управление поворотами в вертикальной плоскости, т.е. изменением положения носовой части вертолета относительно линии горизонта.

Четыре шестилопастных винта располагались на концах крестообразной

фермы из стальных труб с растяжками из рояльных струн. Два небольших винта с горизонтальной тягой служили для путевого управления и поворотов в режиме висения.

До весны 1923 года Г. Ботезат, пилоты Т. Бейн и А. Смит совершили несколько удачных полетов. В одном из них вертолет поднял на 4 м полезный груз весом 450 кг.

Разработка квадрокоптера велась на деньги армии США. Цель — создать летательный аппарат с вертикальными взлетом и посадкой. Однако проект был закрыт из-за ряда проблем. Например, чтобы просто лететь вперед, был нужен попутный ветер.

Были и позже определенные попытки создания таких аппаратов, но вторую жизнь эта идея получила у авиомodelистов в 2000-х годах, когда стала широко доступна модельная электроника и аппаратура радиоуправления. Первыми получили распространение самодельные трикоптеры благодаря тому, что для реализации такой схемы не нужно было никакого специфического контролера, только модельные бесколлекторные двигатели, регуляторы оборотов, вертолетные гироскопы и сервомашинка для реализации поворотного управляющего узла.



Рис. 1.2.2. Трикоптер. Один из первых мультикоптеров.

Именно с этого момента мультикоптеры начали становиться популярными. Они были просты и не очень затратны в изготовлении. Естественно, стали предприниматься попытки получить кадры с воздуха, для чего стали фиксировать на БЛА легкие и миниатюрные фотоаппараты, видеокамеры и даже телефоны.

В 2006 году немецкой компанией Mikrokopter был создан полетный контроллер для мультикоптеров, что позволило достичь совершенно новых возможностей. С этим устройством мультикоптеры были способны удерживать позицию по GPS, делать автономную посадку и автовозврат в случае потери сигнала, да и вообще, обладали большинством всех тех возможностей, которыми обладают современные мультикоптеры.

Именно с появлением этого контроллера можно связывать появление аэрофотосъемки с беспилотных мультикоптеров как профессионального вида деятельности. Стали появляться профессиональные команды по аэрофотосъемке, которые предлагали качественные услуги по съемке фото с высоты, созданию видеороликов с воздуха и съемке панорам.

Контроллеры от Mikrokopter имели и недостатки. Во-первых, стоимость их была велика, только один полетный контроллер стоил не меньше 1000\$. Во-вторых, он состоял из множества плат, которые собирались пользователем вручную, что требовало навыков пайки и начальных знаний радиоэлектроники. В-третьих, настройка этого контроллера была весьма сложной.

Следующий шаг в развитии этой области был совершен китайской компанией DJI-Innovations. Их полетный контроллер DJI Wookong-m был ориентирован на более широкий круг пользователей, и его использование не требовало специальных навыков и знаний. Элементы контроллера нужно было лишь соединить коннекторами между собой без пайки, а программа настройки была легка и понятна практически каждому, при этом возможности этого контроллера превышали по многим параметрам разработки немецкого Mikrokopter. Стоимость этого контроллера была тоже высокой, но простота настройки и управления, а также его надеж-

ность и возможности позволили DJI занять уверенную позицию на рынке полетных контроллеров.

Еще один значимый шаг в истории развития мультикоптеров совершили опять же DJI-Innovations, которые на этот раз выпустили дешевый, надежный и доступный многим контроллер Naza для мультикоптеров, который обладал большинством возможностей предыдущей профессиональной модели.

Благодаря этому использование мультикоптеров стало доступно каждому, и стала активно развиваться аэросъемка.

1.3 Сферы применения квадрокоптеров

Благодаря простоте в эксплуатации и маневренности квадрокоптеры становятся инструментом для решения все большего количества. Проводится огромное количество экспериментов по интеграции такого рода летательных аппаратов в различные сферы промышленности и деятельности человека.

Дроны в последнее время получают много негативной критики. Нередко публикуются новости об использовании квадрокоптеров с целью шпионажа и вторжения в частную жизнь людей или о случаях нанесения травм дронами. Тем не менее, есть много полезных способов применить беспилотные летательные аппараты

Авиакомпании обеспокоены возможностью дронов находиться вблизи их воздушных судов во время рейсов, так как это может нести угрозу. Тем не менее, EasyJet и ряд других авиакомпаний оценивают использование беспилотных летательных аппаратов, а именно квадрокоптеров, в качестве инструментов для ускорения проверки воздушных судов при профилактических осмотрах или перед введением судна в эксплуатацию [2].

Сегодня инспектирование самолета выполняется квалифицированными специалистами. Осматривать внутренние части легко, но осмотр внешней обшив-

ки может оказаться трудным, и требует подвижных платформ, которые помогут получить доступ ко всем частям самолета. В результате для полной инспекции будет потрачено много времени, которое является ценным ресурсом для авиакомпаний. Поэтому авиакомпании используют квадрокоптеры, оснащенные камерами высокой четкости, чтобы ускорить этот процесс. Квадрокоптеры могут летать вокруг самолетов, снимая фотографии высокой четкости и видео, которые инженер может затем просмотреть вместо того, чтобы применять сложные конструкции для получения доступа ко всем элементам воздушного судна для осмотра.

Также одним из показательных примеров является их применение в поисково-спасательных работах. Применение специализированных дронов для помощи спасательным командам во время чрезвычайных ситуаций ведется на протяжении нескольких лет. Этому способствуют следующие возможности дронов:

- летать невысоко над землей, не представляя опасности для людей;
- маневрировать и летать в труднодоступных зонах;
- производить фото- и видеосъемку с воздуха по значительно меньшей стоимости по сравнению с классическими вертолетами.

Как полноразмерные самолеты, они могут нести камеры высокой четкости и даже тепловизионные системы, чтобы помочь найти людей и животных. В связи с этим дроны используются все чаще и чаще в рамках поисково-спасательных работ.

Квадрокоптеры нашли применение и в медицине. Microsoft экспериментирует с использованием беспилотных летательных аппаратов предупреждения болезней. Дроны оснащаются устройством для сбора насекомых, а затем пролетают в различных областях, представляющих интерес для исследователей. Дрон возвращается с собранными насекомыми, которые затем используются учеными для анализа. Результаты анализа используются для прогнозирования вспышек в локальных областях [3].

Разумеется, сам анализ является очень трудоемкой задачей. Во-первых, дроны и связанные с ними устройства сбора насекомых не могут отличить одного насекомого от другого, и не могут собрать только комаров, в которых ученые заинтересованы больше всего. Эти насекомые должны быть отсортированы вручную, и москиты отобраны для исследований. Затем, извлечение образцов крови и их анализ является очень сложным процессом.

Но, несмотря на сложность этой методики, в скором будущем она получит более широкое распространение, и ученые смогут публиковать сводки, предупреждающие людей, живущих в условиях повышенного риска и/или отдаленных районах, о возможности заразиться малярией и другими угрожающими жизни заболеваниями. Это позволит людям принять защитные меры, прежде чем ситуация усугубится, и произойдет вспышка заболевания.

Также БЛА применяются для изучения вулканов, обработка сведений о которых дает возможность более точного прогнозирования извержений, помогает при исследовании минералов внутри и вокруг лавы, исследовании газов, испускаемых вулканом, а также позволяет узнать больше о ядре Земли.

Конечно, основной проблемой при изучении вулканов является наличие высоких температур и токсичных газов. Термокостюмы решили большую часть этой проблемы, но в них трудно передвигаться. Получение снимков с воздуха вместе с измерениями было практически невозможно, используя классическую авиационную технику, так как эта техника не могла летать достаточно близко. Поэтому беспилотные летательные аппараты начали использовать для аэрофотосъемки вулканов и сбора образцов воздуха вблизи них по всему миру. Это позволило ученым получить дополнительные данные, а также создать более точные виртуальные модели кратеров вулканов. Виртуальные модели позволяют ученым изучать изменения в кратерах с течением времени для прогнозирования извержений.

Квадрокоптеры все большее применение находят БЛА в области безопасности и наблюдения. Они могут быстро добраться до охраняемой зоны и осмотреть

ее, полностью исключая все риски для внешнего пилота. Квадрокоптеры используются для:

- борьбы с браконьерами;
- охраны строений и зданий;
- патрулирования границ;
- наблюдения за тюремными территориями;
- наблюдения за скоплениями большого количества людей на протестах и демонстрациях;
- наблюдения за ситуацией на дорогах в зонах сильно оживленного движения и получении сведений о ДТП.

Использование беспилотников для безопасности и наблюдения становится действительно популярным, в связи с чем количество компаний, специализирующихся на этом, растет с каждым днем.

Квадрокоптеры начинают все больше и больше использовать при добыче полезных ископаемых. Дроны запускаются для выполнения заданий, которые являются трудными или опасными для работников. Как и в других отраслях промышленности, беспилотные летательные аппараты оказываются очень ценными при инспектировании инфраструктурной части горнодобывающей системы. Дроны вытесняют применение "ручной" проверки и дорогих в эксплуатации вертолетов при регулярных осмотрах таких объектов, как линии электропередач, дороги и оборудование.

Чтобы обеспечить безопасность сотрудников карьеров, беспилотные летательные аппараты также используются для периодической проверки потенциально опасных зон, таких как стены ям и входы в шахты.

Квадрокоптеры также могут быть использованы для составления карт и моделирования, что имеет большое значение для горнодобывающих компаний, причем применение беспилотных летательных аппаратов в этой области обеспечивает более точные карты и модели, и гораздо дешевле по сравнению с традиционными методами.

Возможно, именно фермерская индустрия больше всех созрела для применения беспилотных летательных аппаратов. Например, благодаря GPS, квадрокоптер может облететь ферму и отследить участок, на котором не хватает орошения или питательных удобрений. Затем, он отправит точные координаты хозяину, который с помощью трактора проведет его удобрение. Или если рассматривать эту задачу более глобально, квадрокоптер может сам произвести обработку поля, облетев всю его площадь. Также БЛА применяются, например, для мониторинга апельсиновых деревьев на предмет наличия болезней, опасных для цитрусовых.

Это далеко не полный список возможных способов применения квадрокоптеров, однако он наглядно демонстрирует их универсальность для решения задач самого широкого спектра.

1.4 Возможности квадрокоптеров

Использование квадрокоптеров в таких сферах, как мониторинг окружающей среды, сельское и коммунальное хозяйство, бытовая сфера и досуг, ликвидация последствий техногенных аварий, аэрофото- и видеосъемка и др. возможны благодаря ряду возможностей квадрокоптеров, дающим им преимущество по отношению к другим видам беспилотных летательных аппаратов.

1.4.1 Базовые возможности

В базовой комплектации квадрокоптеры обладают следующими возможностями:

- поднимать на высоту до 5 километров полезную нагрузку весом до 7 кг. В качестве полезной нагрузки может быть фото-, видеооборудование, датчики, тепловизоры, листовки для оповещения населения о чрезвычайных происшествиях и др.;

- зависать на заданной оператором высоте с возможностью ее плавного увеличения и уменьшения;

- перемещаться во всех направлениях со скоростью до 110 км/ч на расстояние до 12000 метров;
- находиться в воздухе в пределах от 7 до 50 минут - зависит от конфигурации аппарата и его полезной нагрузки;
- эксплуатироваться в широком диапазоне температур наружного воздуха от -30С до 55С, а также при ветре до 20 м/с.

1.4.2 Возможности при расширенной аппаратной базе

При оснащении аппарата дополнительным оборудованием, кроме вышеуказанных возможностей, аппарат способен:

- осуществлять автономный полет по маршруту, обозначенному точками на карте с остановками в данных точках на заданное оператором время, с заданной точностью удержания позиции;
- удерживать заданную оператором высоту и положение;
- возвращаться к месту взлета от любой точки маршрута и от любого удерживаемого положения;
- осуществлять полет с зафиксированной в определенном направлении осью аппарата;
- трансляция видеопотока с борта аппарата, для визуального контроля местности с борта аппарата.
- наличие возможности телеметрического контроля оператором на земле всех параметров летящего аппарата (его положения по координатам GPS, заряд батареи, ток потребления, израсходованная емкость батарей, полетное время, ориентацию по сторонам света, количество спутников в системе GPS и др.). Данные выводятся на экран РС с возможностью вмешательства в параметры, а также в виде наложенной текстовой информации на видео поток с борта аппарата.
- запуск исполнительного механизма на борту аппарата по команде с земли.

1.4.3 Возможности при введении дополнительных программных компонент

При оснащении аппарата дополнительным программным обеспечением, кроме вышеуказанных возможностей, аппарат способен:

- осуществлять автономный полет по маршруту, обозначенному точками на карте с остановками в данных точках на заданное оператором время, с заданной точностью удержания позиции и на заданной высоте, а также фиксация продольной оси аппарата на заданный объект, что позволяет снимать объект при облете его по точкам в полностью автоматическом режиме;

- реагировать на события, среди которых, разряд батареи, превышение дистанции или высоты, потеря связи с оператором. Реакцией может служить возврат на место взлета или любое другое действие;

- осуществлять запись траектории полета с сопутствующими параметрами на карту памяти;

- выводить голосовые сообщения на компьютер, являющийся наземной станцией, о наиболее критичных параметрах среди которых, разряд батареи, превышение дистанции или высоты, потеря связи с оператором, полетное время, дистанция до дома и до цели;

- управлять наземным поворотным механизмом с базовыми направленными антеннами для осуществления лучшей радиосвязи с подвижным объектом.

- компенсировать гироскопически стабилизированным подвесом фото- и видеооборудования наклона летающей платформы в 4-х направлениях, для обеспечения горизонтальности снимков и видеосъемки.

Возможности конкретного квадрокоптера во многом зависят от характеристик системы управления. На сегодняшний день существует большой выбор полетных контроллеров с разными характеристиками и в широком ценовом диапазоне. Выбор электроники зависит от того, решение каких задач возлагается на квадрокоптер.

1.5 Выводы

Многие области науки и промышленности требуют наличие гибкого инструмента, обладающего большим количеством функций, для выполнения самых разнообразных заданий.

В связи с этим особенное внимание уделяется разработке полетных контроллеров, представляющих собой систему управления квадрокоптером. Они должны быть надежными, иметь малые массогабаритные показатели, высокую производительность, возможность адаптации алгоритмов полета для решения поставленных целей.

Сочетание этих требований приводит к тому, что полетные контроллеры имеют высокую ценовую категорию (сотни долларов), но производители стремятся обеспечить контроллер максимальным количеством функций при меньшей стоимости. При решении разного рода задач и организации технологических процессов необходимо в условиях ограниченных аппаратных ресурсов обеспечить максимальную производительность, что ведет к поиску способов наиболее эффективно использовать аппаратуру.

В целом требования к системе управления квадрокоптером можно сформулировать так:

- а) гибкость - система должна иметь возможность легкого перепрограммирования под другой тип выполняемых квадрокоптером работ;
- б) универсальность - система может подходить к различным типам мультироторных БПЛА, с учетом наименьшей времени настройки;
- в) невысокая стоимость - система должна быть легко доступна на рынке, с целью массового использования.

2 Структурно-параметрически синтез системы управления на базе raspberry Pi

2.1 Наиболее распространенные полетные контроллеры

Существует большое множество полетных контроллеров, но когда мы сталкиваемся с большим разнообразием контроллеров, необходимо определить, какой из них лучше подходит для решения поставленных задач. Перед тем как сделать сравнение полетных контроллеров, стоит назвать наиболее часто упоминаемые в литературе [4, 5].

Таблица 2.1 — Модели полетных контроллеров

Полетный контроллер	Страна	Производитель	Конструкция
APM 2.8	США	3D Robotics	Моноблочный; в корпусе.
CC3D	США	OpenPilot	Моноблочный; без корпуса.
NAZE32	Япония	MultiWii	Моноблочный; без корпуса.
KK2	Норвегия	Rolf R Bakke	Моноблочный; без корпуса; встроенный ЖК экран
MultiWii Lite V1.0;	Япония	MultiWii	Моноблочный; без корпуса.
DJI Naza-M Lite	Китай	DJI	Моноблочный; в корпусе.
3DR Pixhawk.	США	3D Robotics	Моноблочный; в корпусе.

Есть три наиболее распространенных типа применения БЛА: спортивный полет, автономный полет и полет с видеозаписью.

На основе анализа типов применения, цены и спецификации рассматриваемых полетных контроллеров, выявляются некоторые важные особенности, которые являются принципиальными при выборе полетного контроллера.

2.2 Сравнение спецификаций контроллеров

Основные технические характеристики полетных контроллеров приведены в Таблице 2.2.

Таблица 2.2 — Технические характеристики полетных контроллеров

Контроллер	Особенности	Применение	Цена
APM 2.8	<ul style="list-style-type: none">- arduino совместимый;- бортовая флэш память для автоматической регистрации данных;- возможность подключения GPS, uBlox LEA-6H с компасом;- инерциальное измерительное устройство с 6 степенями свободы MPU-6000.	Видеосъемка	\$96.00
CC3D	<ul style="list-style-type: none">- 32-битный микроконтроллер STM32 (128 кБ Flash памяти и 20 кБ оперативной памяти);- 3-осевой высокопроизводительный гироскоп и 3-осевой акселерометр MPU-6000;- 4 МБ энергонезависимой памяти EEPROM для хранения настроек.	Спортивные полеты	\$20
NAZE32	<ul style="list-style-type: none">- 8 входных радиоканалов- встроенный преобразователь данных телеметрии FrSky- 32-битный процессор STM32 (3.3V/72MHz).	Спортивные полеты	\$13.9

KK2	<ul style="list-style-type: none"> - 8-битный микроконтроллер Atmel Mega644PA с 64кБ памяти - встроенный жидкокристаллический дисплей 	Спортивные полеты	\$20
MultiWii Lite V1.0	<ul style="list-style-type: none"> - FTDI/UART TTL разъем для отладки, загрузки программного обеспечения или подключения ЖК-дисплея\$ - I2C интерфейс для внешний датчиков; - Отдельные регуляторы напряжения для 3.3 и 5 В\$ - 6-ти осевой инерциальный датчик MPU6050\$ - ATmega 328P Microcontroller. 	Автономный полет	\$27
DJI Naza-M Lite	<ul style="list-style-type: none"> - интеллектуальный контроль ориентации; - встроенные функции стабилизации механизированных подвесов; - поддержка Sbus и PPM; - возможно изменение параметров системы удаленно во время полета. 	Видеосъемка	\$145.99
3DR Pixhawk	<ul style="list-style-type: none"> - 32-битный процессор ARM Cortex M4 - работает под управлением операционной системы NuttX - 14 выходных каналов ШИМ; - microSD карта памяти для записи данных полета; 	Автономный полет	\$204

Видеосъемочные полеты требуют контроллер, который обеспечивает наиболее плавное движение. Автономный полет требует наличие контроллера с открытым исходным кодом, чтобы пользователь мог изменить те или иные особенности полета квадрокоптера. Спортивные полеты могут быть реализованы при наличии полетного контроллера, который удовлетворяет требованиям по скорости реагирования. Контроллеры также не должны стоить дорого, так как такой тип полетов чаще всего приводит к авариям.

2.3 Одноплатный компьютер Raspberry Pi.

Raspberry Pi - одноплатный компьютер размером примерно с банковскую карту изначально разрабатывался как бюджетная система для обучения информатике, впоследствии получивший намного более широкое применение и распространение. Выпускается Raspberry Pi в нескольких версиях [6], данные о которых приведены в таблице 2.3.

Таблица 2.3 — Модели Raspberry Pi

Мо- дель	Процессор	Частота	Ядер	ОЗУ	GPIO	USB	Стоимость
A	ARM1176JZ-F	700MHz		256 Мб	26 пинов	1 порт	\$20
A+	ARM1176JZ-F	700MHz		256 Мб	40 пинов	1 порт	\$25
B	ARM1176JZ-F	700MHz		512 Мб	26 пинов	2 порт	\$35
B+	ARM1176JZ-F	700MHz		512 Мб	40 пинов	4 порта	\$30
2B	ARM Cortex-A7	900MHz	4	1 Гб	40 пинов	4 порта	\$30
Zero	ARM1176JZ-F	1GHz		512 Мб	40 пинов	1 порт miniUSB OTG	\$5
3	ARM Cortex-A53 x64	1,2GHz	4	1 Гб	40 пинов	4 порта	\$35

Для предлагаемого полетного контролера выбран компьютер Raspberry Pi B+. В отличие от других имеет характеристики:

- размеры 85 × 54×17 мм;

- масса 45 гр;
- напряжение питания 5В;
- потребляемый ток 700-1500 мА в зависимости от подключенной периферии.

Вывод видеосигнала возможен через композитный разъем RCA или через цифровой HDMI-интерфейс. В версии «В+» и «2В» вывод возможен через аудиоразъем 3,5.

Одной из самых интересных особенностей Raspberry Pi является наличие портов GPIO (general purpose input/output). Благодаря этому компьютер можно использовать для управления различными устройствами. В модели «В» платы присутствуют 26 портов, а в модели «В+» и «2 В» - 40 портов GPIO

При работе на частоте 700 МГц (по умолчанию), первое поколение Raspberry Pi обеспечил реальную производительность примерно эквивалентно 0,041 GFLOPS. На уровне процессора производительность практически как на 300 MHz Pentium II 1999 года. Графический процессор обеспечивает 24 GFLOPS вычислительной производительности. Графические возможности в Raspberry Pi примерно эквивалентны уровню исполнения Xbox 2001.

Raspberry Pi 2 основан на Broadcom BCM2836, который включает в себя четырехъядерный процессор Cortex-A7 с тактовой частотой 900 МГц и 1 Гб оперативной памяти. Он в 4-6 раз более мощный, чем его предшественник. Графический процессор заменен не был.

Устройство питается через порт микро-USB от источника 5V. Сколько именно тока потребуется Raspberry Pi зависит от того, что будет к ней подключаться. Чаще всего применяются источники питания на 1200 мА.

Как правило, модель В потребляет 700-1000 мА в зависимости от того, какие периферийные устройства подключены; Максимальная потребляемая мощность Raspberry Pi может достигать 1 ампера.

Потребление мощности увеличивается по мере использования различных интерфейсов на Raspberry Pi. Выводы GPIO могут безопасно потреблять 50мА, распределенных по всем контактам; одиночный GPIO контакт может безопасно использовать 16 мА. Порт HDMI использует 50мА, модуль камеры требует 250мА, а подключаемые клавиатуры и мыши могут потреблять от 100мА до 1000мА.

Raspberry Pi позволяет программировать GPIO, чтобы взаимодействовать с реальным миром. Входные сигналы могут поступать с датчика или от другого компьютера или устройства. Выходной сигнал может использоваться для различных целей от включения светодиода до передачи данных на другое устройство. Если Raspberry Pi находится в сети, вы можете управлять устройствами, которые подключены к нему, из любой точки мира, и эти устройства могут передавать данные обратно.

2.4 NAVIO+

Navio+ является платой расширения для Raspberry Pi A + / B +, благодаря которой Raspberry Pi обретает полную функциональность полетного контроллера. Navio+ был разработан для пользовательских робототехнических проектов и в качестве платформы для Linux версии APM (ArduPilot). APM это платформа для беспилотных летательных аппаратов с открытым исходным кодом.

Navio+ может обеспечивать WiFi или 3G соединение, необходимо лишь подключить соответствующий адаптер в порт USB, сжимать видео с камеры или даже запускать программу распознавания изображений. Navio+ позволяет управлять всеми видами движущихся роботов: автомобилями, лодками, самолетами, мультикоптерами. Для получения точных знаний о положении и ориентации Navio+ оснащена инерциальным измерительным устройством (ИИУ) и GPS / Глонасс / Beidou приемником.

На борту Navio находятся:

- инерциальное измерительное устройство с девятью степенями свободы MPU9250;

- барометр MS5611;
- Глонасс/GPS/Beidou приемник U-blox M8N;
- АЦП ADS1115;
- модуль FRAM-памяти MB85RC;
- генератор ШИМ PCA9685;
- RGB светодиод;
- 13 выходных каналов ШИМ.

Navio+ использует хорошо зарекомендовавшую себя платформу APM и может работать в различных режимах полета, включая ручное управление, режим стабилизации, режим преследования и автоматический режим. Код выполняется непосредственно на Raspberry Pi в режиме реального времени Linux ядра, и есть возможность запускать несколько приложений вместе.

2.4.1 ШИМ генератор

PCA9685 - генератор ШИМ на Navio+, который может быть использован для управления сервоприводами и светодиодами. Он имеет:

- 16 каналов с отдельным управлением
- 12-битное разрешение
- настраиваемую частоту
- возможность использовать I2C на частоте до 1 МГц

PCA9685 тактируется генератором 24.576MHz TCXO и позволяет регулировать частоту с помощью регистра PRE_SCALE.

2.4.2 Инерциальное измерительное устройство

MPU9250 является одним из лучших в инерциальных датчиков в своем классе, который сочетает в себе гироскоп, акселерометр и магнитометр в одном устройстве. Семейство датчиков MPU является не только популярным в составе полетных контроллеров для БЛА, но также широко используется в таких устройствах, как мобильные телефоны, планшеты и т. д.

2.4.3 Приёмник GPS

На борту Navio+ располагается GPS приемник NEO-M8N. Этот приемник обладает наибольшей функциональностью в семействе NEO-M8 и имеет лучшие показатели по таким параметрам как чувствительность (-167 дБм), энергопотребление (22 мА в режиме непрерывного поиска спутников) и скорость обновления навигационной информации (10 Гц). NEO-M8N подключается по SPI, отправляет сообщения, содержащие информацию о местоположении.

2.4.4 Аналого-цифровой преобразователь

ADS1115 представляет собой прецизионный аналого-цифровой преобразователь (АЦП) с разрешением 16 бит. Данные передаются через последовательный интерфейс I2C. ADS1115 может выполнять преобразования со скоростью до 860 выборок в секунду. И большие и малые сигналы могут быть измерены с высокой разрешающей способностью в диапазоне от 156mV до 3,3В. ADS1115 также имеет входной мультиплексор, что обеспечивает два дифференциальных или четыре несимметричных входа. ADS1115 работает либо в режиме непрерывного преобразования, либо в режиме одиночной выборки, который автоматически выключается

после преобразования и значительно снижает потребление тока во время периодов простоя.

2.4.5 Модуль FRAM памяти

Сегнетоэлектрическая оперативная память имеет такую же функциональность как флэш-память, но с более высокой скоростью записи, более низким энергопотреблением и значительно большим максимальным количеством циклов перезаписи. В Navio+ используется модуль MB85RC04 Fujitsu с 4096 бит памяти, который подключен через I2C.

2.5 Программное обеспечение Raspberry Pi

Raspberry Pi работает в основном на операционных системах, основанных на Linux ядре. Запуск Windows возможен благодаря средствам виртуализации/ARM11 основан на 6 версии ARM, на котором несколько популярных версий Linux больше не запускаются. Raspberry предоставляет Debian и Arch Linux ARM дистрибутивы для загрузки, и продвигает Python в качестве основного языка программирования, с поддержкой BBC BASIC, C, C ++, Java, Perl, Ruby и многого другого. Для установки операционных систем существует инструмент NOOBS.

2.6 Функциональное назначение полётных контроллеров

Главным отличием полетных контроллеров, основным применением которых являются спортивные полеты, заключается в способности обеспечивать высокую маневренность летательного аппарата. Но эти контроллеры не применимы

для автономных перемещений аппарата или при выполнении видеосъемки с воздуха. Они не имеют компаса и барометра, что делает невозможным реализацию функции удержания высоты, и не имеют GPS модуля, из-за чего БПЛА не способен выполнять полет по заданным точкам на карте или возвращаться в точку взлета при снижении уровня заряда аккумулятора. Таким образом, контроллеры для спортивных полетов непригодны для решения более серьезных задач.

Функциональность контроллеров, целевым назначением которых являются полеты с видеосъемкой, имеют больший функционал по сравнению с контроллерами предыдущего класса. Они используют более сложные алгоритмы стабилизации полета, что обеспечивает плавное движение БЛА. Также программные компоненты дополнены модулями, позволяющими "накладывать" данные телеметрии (о состоянии заряда аккумулятора, о высоте полета и т.д.) на видеоряд, что дает полное представление о характеристиках летательного аппарата и собственно полета, и модулями, реализующими функции следования за съемки или облета его по окружности заданного радиуса при сохранении ориентации объектива на объект.

Они способны обеспечивать как видеосъемку, так и выполнение определенных технологических процессов. Основные способности таких контроллеров:

- автоматический взлет и посадка;
- самостоятельный полет по заданным точкам на карте;
- функция failsafe — набор действий при разряде аккумулятора / возникновении препятствия / потере связи с наземной станцией;
- программируемые границы, за которые квадрокоптер не может вылететь;
- режимы удержания высоты и позиции;
- облет заданной площади;
- возможность подключения дополнительных периферийных устройств.

Контроллеры для автономных полетов являются самыми многофункциональными и дорогими (см. таблицу), хотя есть и исключения. Например, контроллер MultiWii Lite V1.0 имеет стоимость порядка 30\$ и способен совершать полет

по заданным точкам на карте. Однако набор его функций сильно ограничен (нет функции failsafe, полета над заданной площадью, автоматического взлета и посадки), и используемые электронные компоненты очень чувствительны к вибрациям, что отражается на показаниях датчиков.

Одним из лидеров среди полетных контроллеров для автономных полетов является 3DR Pixhawk. Он функционирует под управлением операционной системы реального времени NuttX, позволяющей изменять программные компоненты системы и вносить правки в гибкую систему логгирования данных полета. Pixhawk обладает всеми вышеперечисленными функциями контроллеров своего класса.

Благодаря тому, что Navio+ функционирует под управлением операционной системы на основе ядра Linux и есть возможность вносить правки в программное обеспечение, функционал контроллера может расширяться пользователем. Navio+ обладает всеми достоинствами полетных контроллеров для автономных полетов и способен выполнять все необходимые для этого функции. При этом серьезным преимуществом перед другими контроллерами является наличие портов универсального интерфейса USB, которые позволяют осуществлять связь с большим количеством устройств. Например, к Navio+ можно подключить систему компьютерного зрения, что позволяет решать широкий спектр задач, связанных с распознаванием образов. При этом стоимость данного полетного контроллера (Raspberry Pi — Navio+) составляет порядка 198\$.

Для летательных аппаратов, управляемых дистанционно, в первую очередь применяется управление по радиоканалу с помощью пульта ручного управления. При этом происходит задание нескольких параметров движения (например, углов крена, тангажа и рысканья, высоты полета) и воздействия на дополнительные исполнительные механизмы (например, на подвес с камерой). Поэтому все современные системы радиуправления являются многоканальными.

Пакеты сигнала радиоканала имеют фиксированную длину, и содержат информацию сразу о нескольких каналах в виде последовательности импульсов.

Обработка поступающих данных полностью ложится на полетный контроллер и в большинстве случаев выполняется бортовым процессором. Вычислительной мощности Raspberry Pi недостаточно, чтобы качественно обрабатывать этот сигнал по прерыванию и при этом заниматься управлением полета.

2.7 Полётные параметры ручного режима системы управления.

При управлении квадрокоптером необходимо влиять на 4 параметра полета:

- угол крена;
- угол тангажа;
- угол рысканья;
- уровень высоты.

Изменение значения, поступающего по каналу управления высотой, влечет изменение скорости вращения всех двигателей квадрокоптера. При изменении значений на других каналах меняются скорости вращения пар двигателей, что обеспечивает перемещение квадрокоптера в горизонтальной плоскости или его вращение вокруг вертикальной оси. Величина отклонения каждой ручки пульта определяет скорость изменения соответствующего параметра движения БЛА.

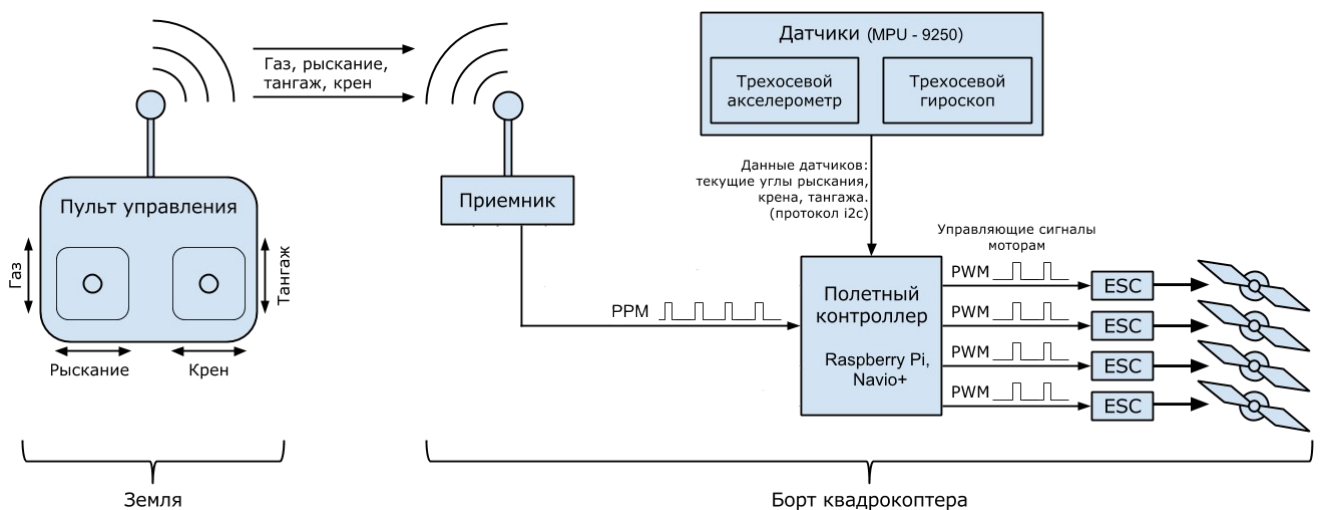


Рисунок 2.1. Основные компоненты системы управления квадрокоптером

На основании получаемого сигнала полетный контроллер генерирует ШИМ сигнал для соответствующих двигателей. Контроллер управляет двигателями не напрямую, а через регуляторы скорости, которые питаются от бортового аккумулятора и по заданию полетного контроллера коммутируют ток бесколлекторных двигателей летательного аппарата.

В случае, если связь с внешним пилотом потеряна или происходит пропуск пакета радиосигнала, то полетный контроллер переключается в режим защиты от падения. В зависимости от настроек в этом режиме квадрокоптер может совершить посадку или вернуться к месту запуска.

Сигнал, поступающий с пульта, несет в себе информацию о наборе каналов управления (от 2 до 14 каналов). В простейшем случае радиоприемник имеет соответствующее количество выходов, по каждому из которых передается информация об отдельном канале. Такая конфигурация аппаратуры подразумевает, что полетный контроллер должен иметь большое количество входов для приема сигналов от радиоприемника. Это влечет значительное увеличение габаритов полетного контроллера и использование пучка проводов для соединения контроллера и приемника.

Почти все современные модели аппаратуры для радиоуправления имеют приемники, на выходе которых можно получить комбинированный PPM сигнал о всех каналах управления. Этот сигнал передается по одному проводу к полетному контроллеру, который в свою очередь занимается обработкой получаемой информации.

Контроллер Navio+ имеет контакт для подключения радиоприемника с целью передачи комбинированного сигнала. Этот контакт напрямую связан с одним из контактов GPIO порта Raspberry Pi, поэтому о его состоянии можно узнать, считывая регистр GPIO, который содержит информацию о всех контактах.

Каждый пакет радиосигнала имеет фиксированную длину 20 мс, и чтобы получить достоверные данные о каждом канале, необходимо считывать состояние ножки GPIO каждую микросекунду. Программная реализация цикла считывания с

такой частотой будет лишать центральный процессор возможности выполнять другие ресурсоемкие задачи.

Чтобы избавить центральный процессор Raspberry Pi от необходимости обработки регистра GPIO с высокой частотой, принято решение воспользоваться контроллером прямого доступа к памяти (DMAC), который является частью процессора ARM1176JZ-S Raspberri Pi [7]. Его задачей в данном случае является пересылка данных из регистра GPIO в выделенный буфер памяти каждую микросекунду. Таким образом, буфер будет заполнен информацией о текущем состоянии пакета сигнала, поступающего с приемника, и CPU может обрабатывать уже накопленные данные с меньшей частотой (1 кГц).

2.8 Выводы

Проведен анализ свойств серийно выпускаемых полетных контролеров, и предложены решения, расширяющие функциональность за счет применения контроллера прямого доступа к памяти, являющегося частью процессора Raspberry Pi.

Показаны преимущества полетного контроллера Navio+ перед другими контроллерами для автономных полетов. Модульная конструкция данного контроллера позволяет наращивать его аппаратную базу, расширяя производительность.

Navio+ способен выполнять полетные миссии по заданному маршруту, но при этом имеет недостаточный набор аппаратных средств, чтобы реализовать традиционный метод (метод с использованием прерываний) обработки данных, поступающих по радиоканалу.

3 Разработка алгоритма работы системы ручного управления и программная реализация

3.1 Описание процесса преобразования команд пилота в задание на ШИМ генератор

Целеуказание поступает с пульта оператора в Navio в виде комбинированного фазо-импульсного сигнала (рисунок 3.1). Каждый пакет сигнала имеет фиксированную длину $T = 20\text{мс}$ и состоит из 5 импульсов и синхروпаузы, позволяющей отделять один пакет радиосигнала от другого. Импульсы имеют фиксированную длину 0.2мс , а их положение внутри пакета (или, если говорить конкретнее, расстояние между ними) несет информацию о соответствующем канале. Расстояние между импульсами может варьироваться от 0.7мс до 2.2мс [8].

Вычисленное значение фазы используется для задания скважности широтно-импульсного преобразователя:

$$\tau_{\text{ШИМ}} = K_i(P, H) \varphi_i,$$

где K_i — функционал, зависящий от давления P и высоты полета H , φ_i — фаза i -го импульса.

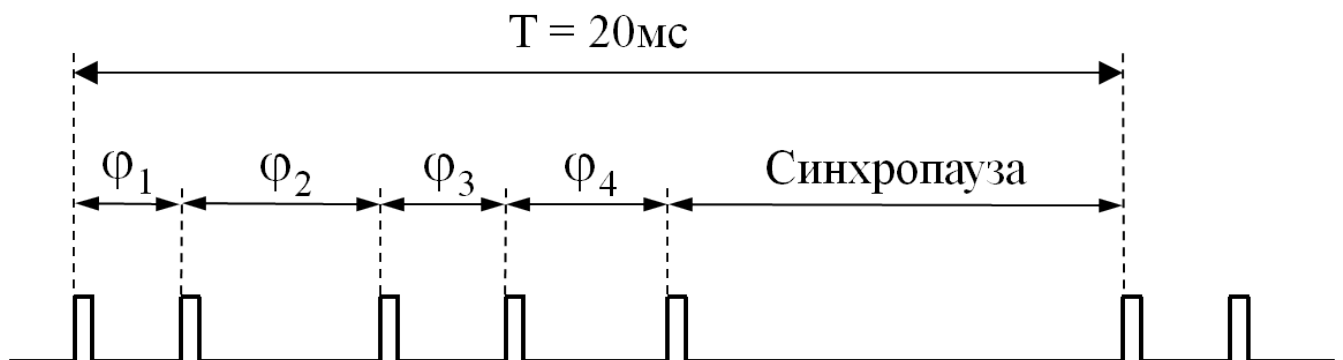


Рисунок 3.1 ФИМ-сигнал с четырехканального пульта управления.

Информация о фазе получается, в результате считывания значения регистра GPIO с частотой 1МГц . Все полученные данные буферизируются для последу-

ющей обработки и представляют собой битовый массив. На рисунке 3.2 показано соответствие между поступающим сигналом и состоянием регистра GPIO.

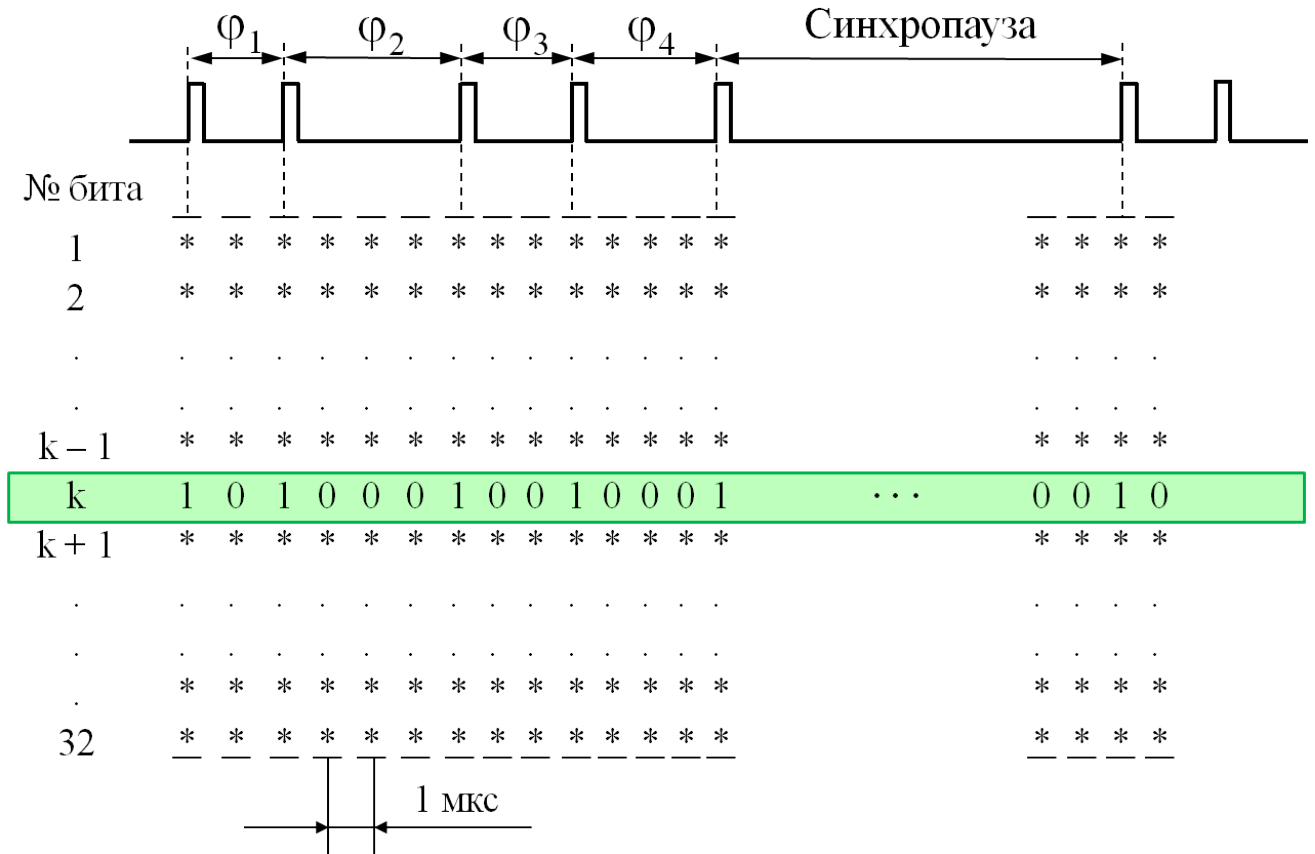


Рисунок 3.2. GPIO – битовый массив.

Для заполнения битового массива используется контроллер прямого доступа к памяти, входящего в состав процессора Raspberry Pi. От центрального процессора требуется лишь послать запрос на пересылку данных. DMA контроллер выполняет считывание состояние ножки GPIO и записывает эту величину в буфер (рисунок 3.3). Центральный процессор обращается к буферу с меньшей частотой и обрабатывает битовый массив сразу по всему пакету радиосигнала, производя расчеты необходимые для широтно-импульсного модулирования.

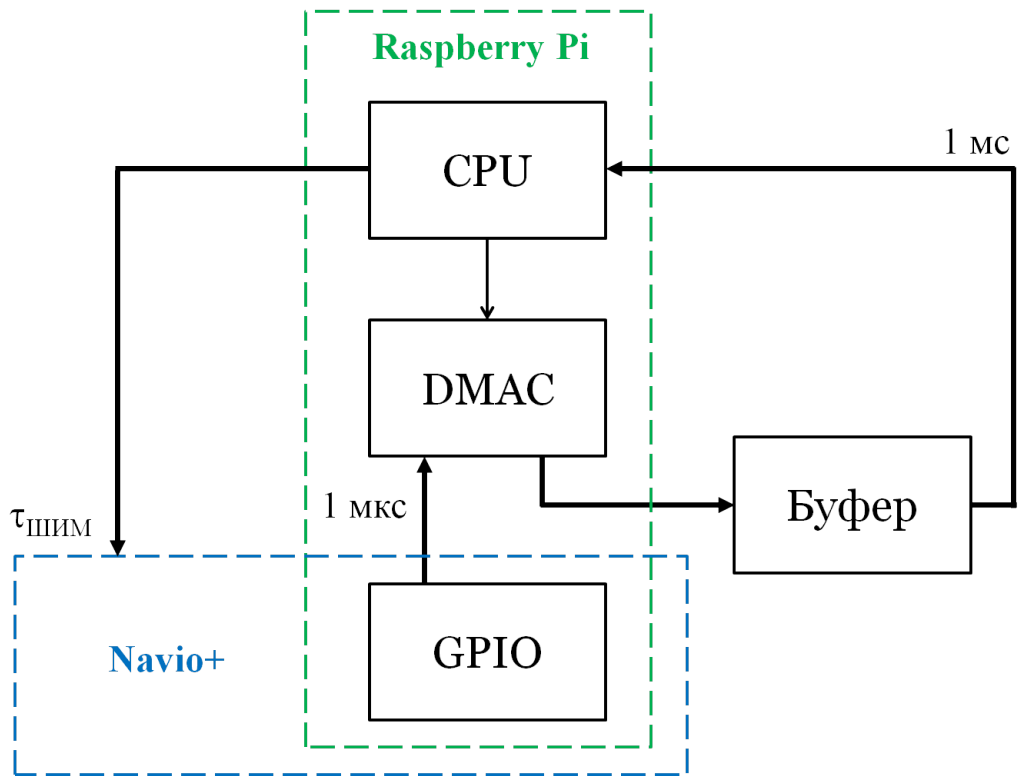


Рисунок 3.3 Взаимодействие компонентов системы обработки данных.

Алгоритм работы системы ручного управления квадрокоптером с использованием прямого доступа к памяти представлен на рисунке 3.4.

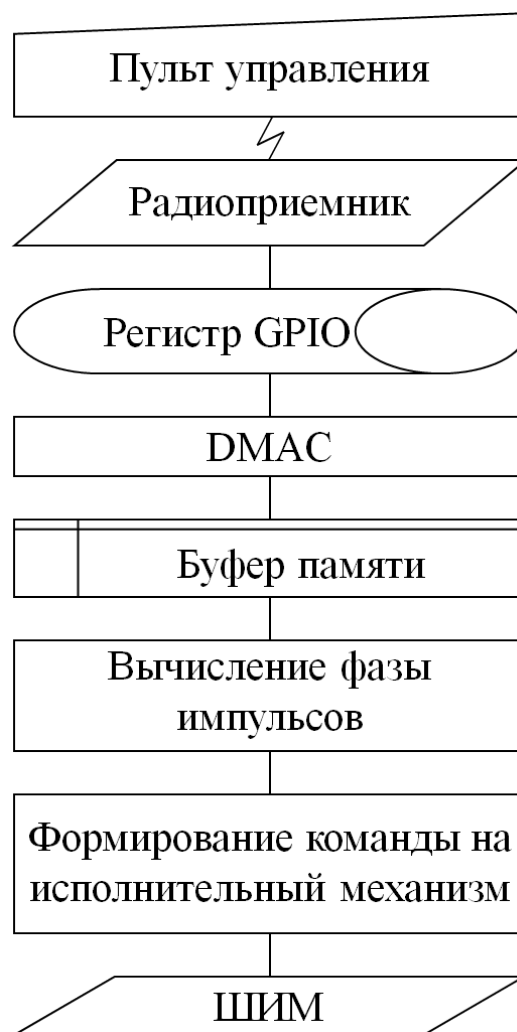


Рисунок 3.4. Алгоритм работы системы ручного управления.

3. Broadcom Corporation. BCM2835 ARM Peripherals [Электронный ресурс] URL: <https://www.raspberrypi.org/documentation/hardware/ra...>

Программная реализация обработки буфера памяти и запуск контроллера прямого доступа к памяти описаны далее.

3.2 Программная реализация

3.2.1 Сведения об операционной системе

Для использования всего функционала платы Navio+ необходимо использовать модифицированную версию операционной системы Raspbian, созданную разработчиками данного полетного контроллера. Raspbian базируется на Unix-подоб-

ной операционной системе Debian, которая использует ядро Linux. Отличие модифицированной версии Raspbian от оригинальной заключается в наличии встроенных интерфейсов для взаимодействия Raspberry Pi со всеми компонентами платы Navio+.

Ядро является центральной частью операционной системы, обеспечивающая приложениям координированный доступ к ресурсам компьютера, таким как процессорное время, память, внешнее аппаратное обеспечение, внешнее устройство ввода и вывода информации. Также обычно ядро предоставляет сервисы файловой системы и сетевых протоколов.

Linux является ядром монолитной архитектуры. Это означает, что в ядро встроены основные компоненты, обеспечивающие нормальное функционирование операционной системы. Но в то же время ядро Linux является модульным, т.е. есть возможность расширять функционал ядра, выполняя динамическую вставку и удаление кода ядра. Соответствующие подпрограммы и данные группируются в общий бинарный образ, загружаемый объект ядра, который называется модулем [9].

Скомпилированные модули можно загружать или выгружать во время работы системы, без необходимости в ее остановке. При этом сбой в одном из компонентов не нарушит работоспособность всей системы.

Ядро и его модули занимают привилегированное положение по отношению к пользовательским программам. Это включает доступ ко всем областям защищенной памяти и полный доступ к аппаратному обеспечению, что дает возможность тщательнее контролировать работу системы.

3.2.2 Программирование модуля ядра

Реализация алгоритма обработки данных, поступающих по радиоканалу возможна и в пространстве пользовательских приложений. Однако в этом случае

нет возможности полностью контролировать работу контроллера прямого доступа к памяти. Его настройку и запуск возможно реализовать, устанавливая определенные биты в регистрах, отвечающих за работу DMA. Остановка выполнения операций DMA происходит таким же образом, однако это является небезопасным для системы в целом. При сбое в выполнении программы и ее аварийном завершении остановки контроллера прямого доступа к памяти не произойдет, так как не выполнится установка нового состояния регистра, отвечающего за функционирование DMA, и перезапись буфера физической памяти не прекратится. Процессы, которые в дальнейшем начнут использовать этот участок памяти, будут работать некорректно, так как информация в памяти будет постоянно перезаписываться контроллером DMA. Это может привести к возникновению непредсказуемых сбоев при дальнейшем функционировании операционной системы.

В пространстве ядра Linux существует надежный механизм управления прямым доступом к памяти – интерфейс DMA Engine. Он предоставляет удобные функции по настройке и запуску операций DMA контроллера. Использование этого интерфейса в модуле ядра исключает возможность потерять контроль над операциями с физической памятью, так как в модуль можно добавить определенный набор инструкций, обеспечивающих остановку работы DMA контроллера, которые будут выполнены при выгрузке модуля из ядра.

Программирование и использование модуля ядра несколько отличается от программирования пользовательского приложения. Приложение имеет одну точку входа, которая начинает исполняться сразу же после размещения запущенного приложения в оперативной памяти компьютера. Эта точка входа описывается на языке Си, например, как функция `main()`. После выполнения всей последовательности команд функция `main()` завершает свою работу, что означает завершение приложения. Модуль имеет несколько точек входа, исполняемых при установке и удалении модуля из ядра, а также при обработке поступающих, от пользователя, запросов. Работа модуля начинается с исполнения функции `module_init()`. Это функция запуска модуля, которая подготавливает его для последующих вызовов.

После завершения исполнения функции `module_init()` модуль находится в режиме ожидания запроса ядра на выполнение специфических действий.

Вторая точка входа в модуль — функция `module_exit()`. Она вызывается непосредственно перед выгрузкой модуля, производя отмену изменений, выполненных при его инициализации. После выполнения `module_exit()` ядро получает информацию о том, что ранее введенная модулем функциональность больше недоступна [10].

На рисунке 3.5 изображена реализация механизма вызова функций, являющихся точками входа в модуль. Загрузка модуля в ядро вручную осуществляется командами `insmod` и `rmmod`. Также, здесь отражен механизм взаимодействия устанавливаемого модуля с ядром. Функция `module_init()` вызывает некоторую функцию `register_capability()`, передав в качестве аргумента указатель на структуру, содержащую указатели на функции в модуле. Функция `register_capability()` помещает указатель во внутреннюю структуру данных "возможностей". Затем система определяет протокол для того, как приложения будут получать доступ к информации в структуре данных `capabilities` посредством системных вызовов. Таким образом при инициализации модуля происходит объявление новой функциональности в ядре. [11]

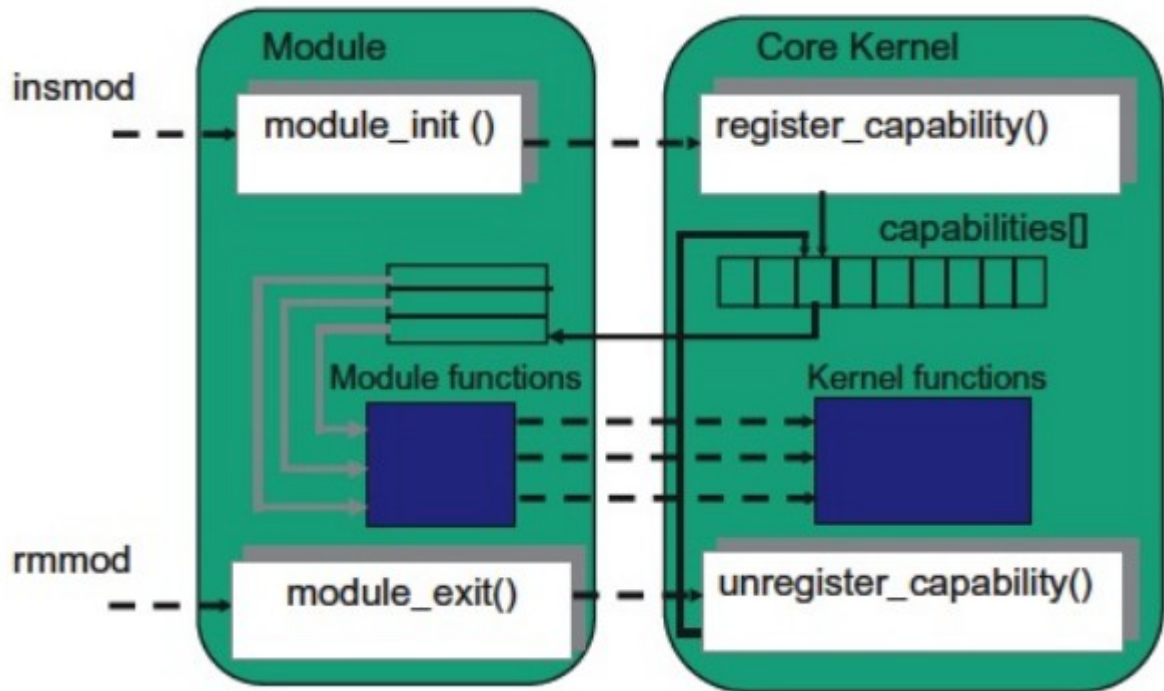


Рисунок 3.5 Связь модуля с ядром. Пунктирная линия — вызов функции, черная линия — указатель на данные, серая линия — указатель на функцию.

Другое отличие модуля от приложения заключается в том, что приложение может вызвать функцию, которая объявлена во внешней библиотеке. Модуль, с другой стороны, связан только с ядром и может вызывать только те функции, которые экспортируются ядром. Код исполняемый в ядре не может использовать внешние библиотеки.

Инициализация DMA состоит из 5 основных этапов [12]:

- выделение канала прямого доступа к памяти;
- настройка специфических параметров DMA контроллера и подчиненного устройства;
- получение дескриптора транзакции;
- утверждение транзакции;
- запуск операций передачи и ожидание выполнения функции обратного вызова.

Настройка механизма транспортировки данных посредством прямого доступа к памяти начинается с запроса свободного канала. В общем случае ведется поиск первого свободного канала, но в некоторых случаях необходимо использовать конкретный канал.

Для выполнения данной операции используется функция `dma_request_channel()`, которая объявлена следующим образом:

```
struct dma_chan *dma_request_channel(dma_cap_mask_t  
mask, dma_filter_fn filter_fn, void *filter_param)
```

Функция осуществляет поиск канала, удовлетворяющего характеристикам, которые задает параметр `mask`. Также при заданном `filter_fn` параметре происходит дополнительная фильтрация каналов по некоторым признакам (например, по номеру канала). Канал, выделенный с использованием `dma_request_channel()` может использоваться только процессом, вызвавшим эту функцию.

После получения канала необходимо передать определенную информацию в драйвер DMA, а именно направление передачи данных, адреса блоков памяти, используемых при транспортировке, пропускную способность шины и т. д. Эти данные хранятся в структуре `dma_slave_config`. После заполнения полей этой структуры выполняется функция `dmaengine_slave_config()`, которая производит настройку выделенного канала в соответствии с параметрами, хранящимися в структуре `dma_slave_config`.

Завершающим этапом настройки DMA контроллера является получение дескриптора транзакции (операции передачи), без которого невозможен запуск DMA. В зависимости от того, как распределены физические адреса памяти, работа с которыми производится контроллером прямого доступа, возможно использование одного из двух режимов работы DMA: циклический (`cyclic`) и режим рассеяния-сборки (`scatter-gather`). В первом случае физические адреса располагаются последовательно, во втором - в разных регионах памяти. Настройка циклического режима выполняется функцией `maengine_prep_dma_cyclic()`, режима рассеяния - функцией `dmaengine_prep_slave_sg()`.

При решении задачи обработки данных, поступающих по радиоканалу, используется циклический режим работы DMA. При этом буфер памяти, в котором накапливаются данные, циклически перезаписывается. Выделение участка памяти для буфера осуществляется с помощью функции `dma_alloc_coherent()` [13].

После конфигурации канала передачи и получения дескриптора, происходит перевод канала в состояние готовности с помощью функции `dmaengine_submit()`. Затем выполняется функция `dma_async_issue_pending()`, запускающая механизм транспортировки данных. По мере выполнения передачи данных, могут выполняться функции обратного вызова, если таковые были указаны в соответствующих полях дескриптора транзакции.

По окончании функционирования модуля, необходимо вызвать функции `dmaengine_terminate_all()` и `dma_release_channel()` в точке входа `module_exit()`, чтобы остановить все операции DMA и освободить занимаемый канал.

Настройка DMA в функции `module_init()` разрабатываемого модуля описана в Приложении А.

По мере выполнения процесса циклической транспортировки данных из регистра GPIO в буфера памяти происходит обработка получаемого массива информации.

3.2.3 Размещение и применение драйвера-модуля ядра

Скомпилированный модуль (объект ядра с расширением `.ko`) может быть загружен в ядро вручную командой `insmod`.

При необходимости, модуль может быть помещен в каталог системных модулей `/lib/modules/`uname -r`/`, где `"uname-r"` это текущая версия ядра операционной системы. Далее обновляется информация о дереве зависимостей модулей, так как был добавлен новый. Таким разработанный модуль будет добавляться во время каждой загрузки операционной системы.

3.3 Выводы

Разработан и описан алгоритм работы подсистемы ручного управления полетом квадрокоптера.

Полетный контроллер реализован на базе Raspberry Pi с платой расширения Navio+. В алгоритме учтены особенности использования контроллера прямого доступа к памяти центрального процессора Raspberry Pi для обеспечения считывания состояния контакта, к которому присоединяется выход радиоприемника.

Описана программная реализация алгоритма и интегрирование его в ядро операционной системы.

4. Тестирование разработанного модуля

4.1 Испытательный стенд

Испытательный стенд состоит из комплекта радиоаппаратуры Turnigy (передатчик RF 9x и приемник 9x8cV2), одноплатного компьютера Raspberry Pi B+ и платы Navio+. Navio+ соединяется с Raspberry Pi через порт GPIO.

На выходе приемника Turnigy 9x8cV2 имеем сигнал о 8 каналах управления, период сигнала которых составляет 19,6 Гц. Для получения ФИМ-сигнала используется аппаратный преобразователь, комбинирующий информацию о 8-ми каналах в один пакет, который, в итоге, и передается в Navio+.

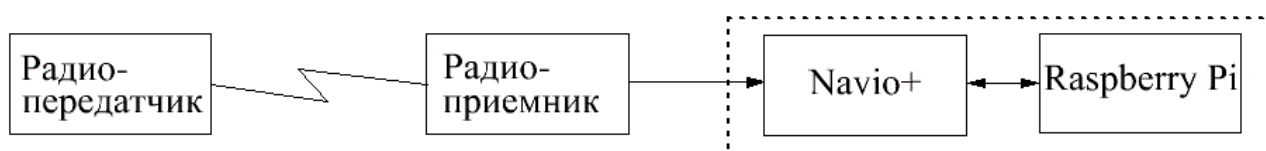


Рисунок 4.1 Схема испытательного стенда

Такая конфигурация оборудования имеет недостаток - период комбинированного сигнала получается больше периода сигнала каждого канала. Однако ошибка, вносимая преобразователем является статической и не влияет на качество сигнала, так как с точки зрения управления квадрокоптером неважно, какой период имеет комбинированный сигнал, потому что перед полетами проводится процедур калибровки радио, во время которой полетный контроллер определяет длину пакетов и минимумы и максимумы всех каналов. Затем проводится калибровка всех регуляторов скорости, после которой их диапазон регулирования соответствует диапазонам управляющих сигналов пульта радиоуправления.

На рисунке 4.2 показаны осциллограммы выходных сигналов радиоприемника и преобразователя, снятые логическим анализатором Saleae Logic 8. Метки A1 и A2 обрамляют один пакет комбинированного сигнала. В правой части окна

приложения видно, что этот пакет имеет длину 21,7 мс. В то же время, пакеты исходных сигналов каждого канала имели период 19,6 мс (метки B1-B2). Соответственно ширина импульса отдельного канала (проводилось изменение на канале номер 3 приемника Turnigy) отличается от значения этого же канала в пакете комбинированного сигнала. На рисунке 4.3 видно, что значение, определяемое метками A1-A2, отличается от значения определяемого метками B1-B2 на 2 микросекунды.

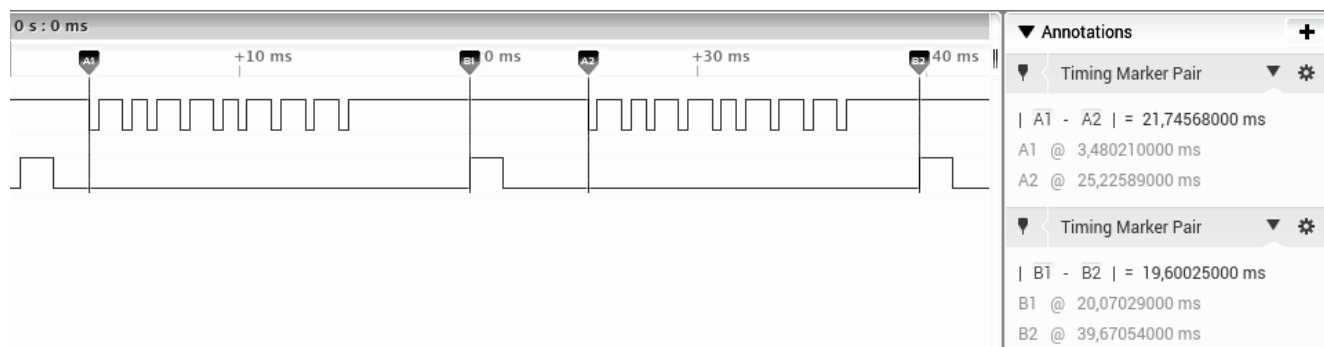


Рисунок 4.2 Период сигнала одного из каналов радиоприемника и комбинированного сигнала.

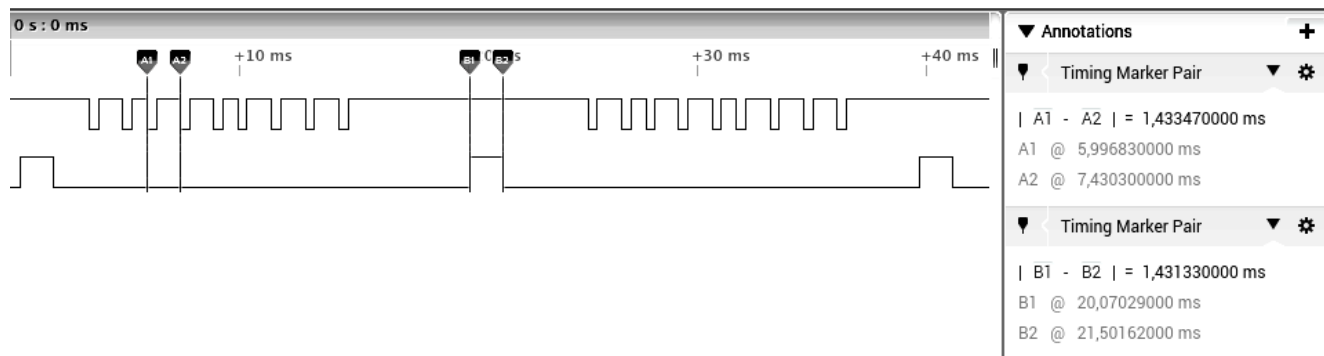


Рисунок 4.3. Ширина импульса третьего канала, измеряемая на выходе приемника и после преобразования.

Результаты работы созданного модуля ядра должны соответствовать измерениям логического анализатора с учетом заданной погрешности (1 мкс).

4.2 Программа и методика испытаний

Для анализа качества выполнения модулем ядра обработки данных, поступающих по радиоканалу, проведены два эксперимента. В первом эксперименте получение ширины импульса ШИМ-сигнала производится путем обработки прерываний, возникающих при изменении логического уровня на контакте GPIO, связанного с разъемом Navio+, к которому подключается радиоприемник. Во втором эксперименте величина ширины импульса получается в результате использования модуля ядра.

4.2.2 Использование механизма прерываний

Собирается стенд и запускается программа, которая выполняет измерение текущего времени при появлении нарастающего и спадающего фронтов сигнала, поступающего на Navio+. Затем, производится вычисление разности между моментами появления нарастающего и спадающего фронтов, которая и определяет ширину импульса ШИМ-сигнала, поступающего по одному из каналов приемника. В таблице 1 приложения Б представлена часть полученных результатов.

Если производить обработку данных, поступающих по радиоканалу, используя механизм прерываний, то невозможно добиться хороших результатов. Измерения ширины импульса имеют в этом случае случайные отклонения. На рисунке 4.4 представлена диаграмма, наглядно демонстрирующая разброс значения ширины импульса при использовании прерываний. По оси абсцисс располагается вычисленная ширина импульса. По оси ординат - количество появлений соответствующего значения в выборке из 1390 измерений.

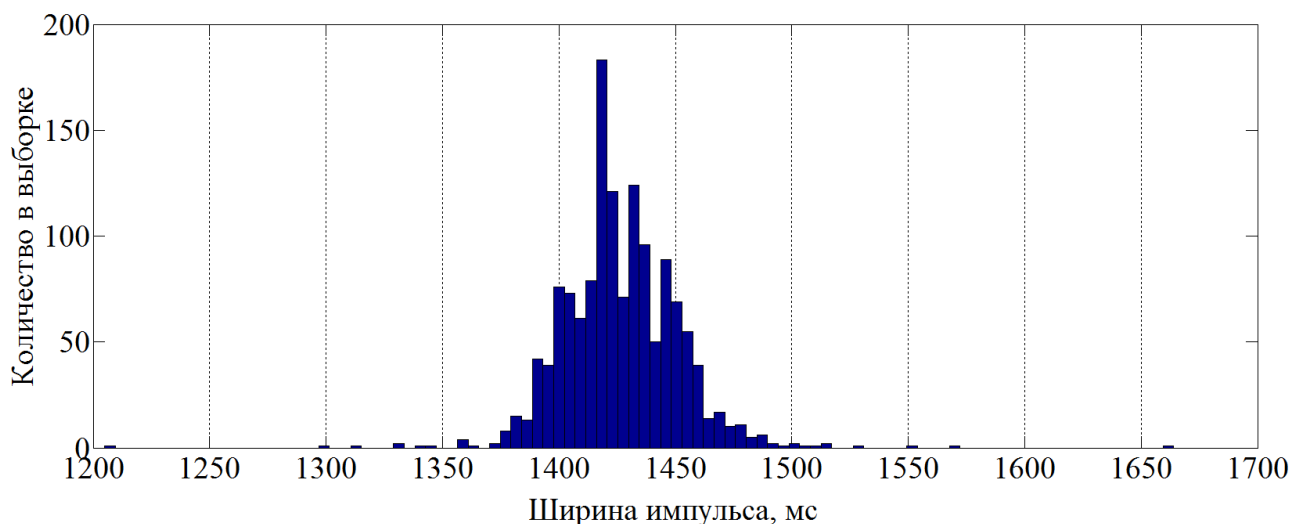


Рисунок 4.4 Разброс величины ширины импульса при обработке сигнала с помощью прерываний.

Значение, измеренное логическим анализатором, составляет 1,431 мс. Но значение, получаемое программными средствами, лежит в интервале от 1370 мс до 1490 мс. Такая нестабильность сигнала приводит к вибрациям рамы квадрокоптера, что влияет на, например, качество видеосъемки.

Также видны редкие появления значений порядка 1210 мс и 1660 мс. Так как Navio+ имеет механизм защиты от падения, недопустимо иметь подобные отклонения. Механизм защиты срабатывает при выходе ширины импульса канала управления газом (скоростью вращения всех четырех двигателей) за нижнюю границу диапазона, полученную при калибровке. В этом случае квадрокоптер перестает принимать команды внешнего пилота и начинает исполнять инструкции по обеспечению безопасного завершения полета (либо возвращение к месту взлета, либо незамедлительная посадка). Поэтому сильные скачки обработанной величины приведут к незапланированному прекращению полетной миссии квадрокоптера или выполняемого им технологического процесса.

4.2.3 Использование модуля ядра

Использование разработанного модуля дает гораздо более высокую точность по сравнению с предыдущим методом вычисления ширины импульса. После обработки поступающих данных модулем ядра имеем массив значений ширины каждого канала пакета радиосигнала. В таблицах 2 и 3 приложения Б приведены фрагменты массивов, полученных при обработке сигнала отдельного канала и комбинированного сигнала.

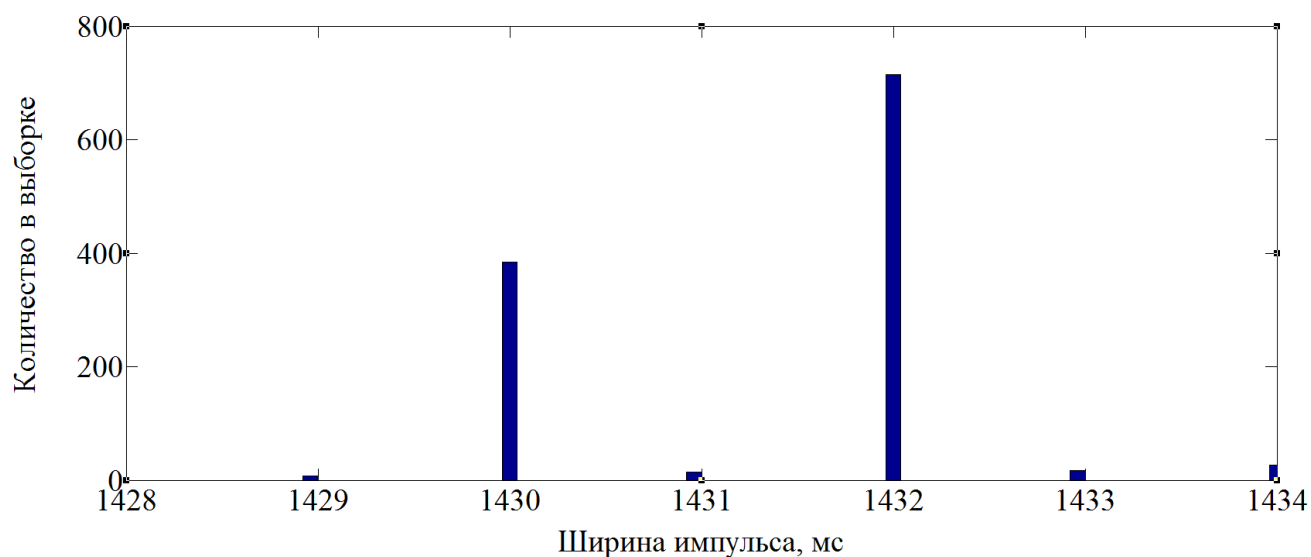


Рисунок 4.5 Разброс величины ширины импульса при обработке сигнала с использованием прямого доступа к памяти

На рисунке 4.5 видно, что значение отклонения измеренной величины от первоначальной составляет 1 микросекунду, и очень редки случаи большего отклонения. В выборке из 1160 измерений всего 40 (3.5 %) имеют величину больше 1432 и меньше 1430.

По массивам данных, полученных с помощью контроллера прямого доступа к памяти, построены графики комбинированного ФИМ-сигнала (рисунок 4.7) и график ШИМ-сигнала отдельного радиоканала (рисунок 4.8).

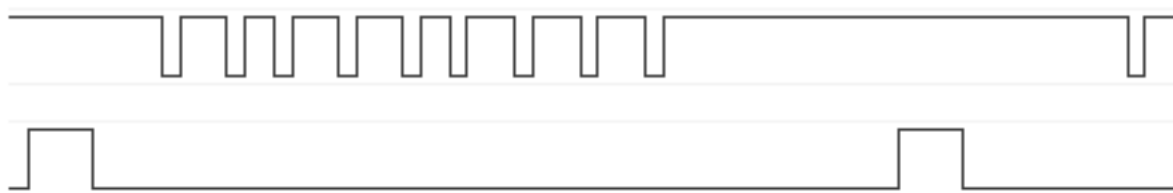


Рисунок 4.6. Осциллограмма комбинированного сигнала и сигнала отдельного канала.

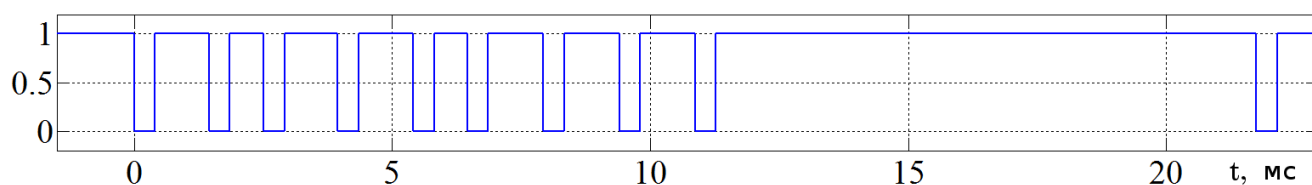


Рисунок 4.7. График комбинированного сигнала, построенный по массиву данных, обработанных модулем ядра.

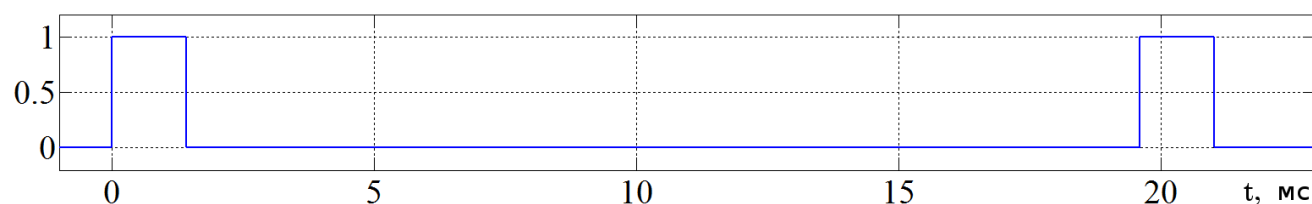


Рисунок 4.8. График комбинированного сигнала, построенный по массиву данных, обработанных модулем ядра.

Графики 4.7 и 4.8 полностью соответствуют осциллограммам логического анализатора (рисунок 4.6), что говорит о правильности выполнения всех этапов обработки.

4.3 Выводы

Разработан и смонтирован испытательный стенд для практической проверки работоспособности предложенной системы.

Проведено натурное моделирование работы подсистемы ручного управления квадрокоптером.

Проведенные эксперименты демонстрируют значительное преимущество метода обработки данных, поступающих по радиоканалу, посредством считывания состояния регистра GPIO с помощью контроллера DMA перед методом, в котором определение состояния GPIO и ширины канала ведется посредством использования прерываний.

Заключение

Проведенные теоретические и экспериментальные исследования позволили расширить функциональность полетного контроллера, обеспечив повышение точности обработки сигналов, поступающих по радиоканалу, в условиях ограниченных аппаратных ресурсов.

Предложен и реализован способ обработки данных целеуказания с использованием имеющегося на плате Raspberry Pi контроллера прямого доступа к памяти, который обеспечивает стабильную передачу данных между областями памяти с необходимой скоростью.

Настройка и запуск контроллера DMA и обработка буфера памяти программно реализованы в виде модуля ядра GNU/Linux, который выполняет функции драйвера для работы с радиоприемником.

Получены экспериментальные данные по обработке полетным контроллером сигнала с радиоприемника. Показано преимущество применения разработанного модуля ядра перед использованием прерываний для обработки поступающего сигнала. По результатам экспериментов отклонение величины ширины импульса ШИМ-сигнала, приходящего с приемника, составляет 1 микросекунду, что является удовлетворительным для процесса управления квадрокоптером.

Список литературы

1. Форум о науке и современных технологиях : электронный ресурс:
<https://geektimes.ru/hub/history/>.
2. EasyJet привлечет к ремонту самолетов беспилотники и Google Glass [Электронный ресурс] / Деловой авиационный портал.- Режим доступа:
<http://www.ato.ru/content/easyjet-privlechet-k-remontu-samoletov-bespilotniki-i-google-glass> (дата обращения 29.01.2016).
3. Microsoft wants to use-catching drones to fight disease [Электронный ресурс] / Popular Science. – Режим доступа: <http://www.popsci.com/microsoft-drones-will-catch-mosquitos-fight-disease> (дата обращения 29.01.2016).
4. Ardupilot Community: электронный ресурс: <http://ardupilot.org/>.
5. DIY Drones: электронный ресурс: <http://diydrones.com/> .
6. Официальный сайт Raspberry Pi: электронный ресурс:
<https://www.raspberrypi.org/products/> .
- 7 ARM1176JZ-S Technical Reference Manual [Электронный ресурс]: ARM Limited. –Режим доступа:
http://infocenter.arm.com/help/topic/com.arm.doc.ddi0333h/DDI0333H_arm1176jzs_r0p7_trm.pdf (дата обращения 25.03.2016).
8. Форум по радиотехнике: электронный ресурс:
http://www.rcdesign.ru/articles/radio/ppm_pcm .
9. Лав, Роберт. Разработка ядра Linux, 2-е издание.: Пер. с англ. — М.: ООО "И.Д. Вильямс" 2006. — 448 с.
10. Peter Jay Salzman. The Linux Kernel Module Programming Guide [Электронный ресурс] / Peter Jay Salzman, Michael Burian, Ori Pomerantz. - Режим доступа: <http://www.tldp.org/LDP/lkmpg/2.6/lkmpg.pdf> (дата обращения 08.03.2016).
- 11 Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman. Linux Device Drivers [Электронный ресурс] / Corbet, Alessandro Rubini, Greg Kroah-Hartman. –

Режим доступа: <http://free-electrons.com/doc/books/ldd3.pdf> (дата обращения 8.03.2016).

12 DMA Engine API Guide [Электронный ресурс] /Linux Kernel Documentation. – Режим доступа: <https://www.kernel.org/doc/Documentation/dmaengine/client.txt> (дата обращения 11.03.2016).

13 Dynamic DMA mapping Guide [Электронный ресурс] /Linux Kernel Documentation. – Режим доступа: <https://www.kernel.org/doc/Documentation/DMA-API-HOWTO.txt>(дата обращения 11.03.2016).

Текст программы настройки контроллера прямого доступа к памяти

```
dma_cap_mask_t mask;
struct dma_slave_config sconf;
struct dma_async_tx_descriptor *desc;

dma_cap_zero(mask);
dma_cap_set(DMA_CYCLIC, mask);
chan = dma_request_channel(mask, NULL, NULL);

if (!chan) {
    printk("DMA channel request FAIL\n");
    return -1;
} else {
    printk("Requested channel %d\n", chan->chan_id);
}

buf_for_alloc = dma_alloc_coherent(chan->device->dev,
BUF_LEN, &buf_addr, GFP_KERNEL);
if (!buf_for_alloc) {
    printk("dma_alloc_coherent error\n");
    dma_release_channel(chan);
    return -1;
}

sconf.direction = DMA_DEV_TO_MEM;
sconf.src_addr = GPIO_BUS_BASE + GPLEV0_OFFSET;
sconf.device_fc = false;
sconf.src_addr_width = DMA_SLAVE_BUSWIDTH_4_BYTES;
```

- 60 -

```
sconf.dst_addr_width = DMA_SLAVE_BUSWIDTH_4_BYTES;

ret = dmaengine_slave_config(chan, &sconf);
if (ret < 0) {
    return ret;
}

desc = dmaengine_prep_dma_cyclic(chan, buf_addr,
BUF_LEN, PERIOD_LEN, DMA_DEV_TO_MEM, DMA_CTRL_ACK |
DMA_PREP_INTERRUPT);

if (!desc) {
    printk("Cannot prepare cyclic dma\n");
    dma_release_channel(chan);
    return -1;
}

desc->callback = dma_callback;

dma_cookie = dmaengine_submit(desc);
dma_async_issue_pending(chan);
```

Результаты экспериментов по вычислению ширины импульсов, поступающих с радиоприемника

Таблица 1 - Фрагмент массива данных, полученных обработкой канала №3 приемника Turnigy с использованием прерываний. φ - ширина импульса, Δ - интервал времени между окончанием одного импульса и началом следующего.

φ , мкс	Δ , мкс
1408	18199
1397	18209
1420	18186
1451	18178
1433	18153
1423	18178
1404	18200
1419	18187
1438	18177
1459	18168
1430	18151
1418	18179
1400	18205
1420	18185
1460	18178
1435	18145
1452	18146
1392	18186
1419	18178
1435	18169
1515	18133

Таблица 2 - Фрагмент массива данных, полученных обработкой сигнала с канала №3 приемника Turnigy модулем ядра

φ, мкс	Δ, мкс
1430	18170
1430	18168
1432	18168
1432	18170
1430	18170
1432	18166
1430	18168
1432	18170
1432	18168
1432	18168
1430	18170
1430	18170
1430	18170
1432	18169
1429	18171
1430	18170
1430	18170
1432	18168
1430	18170
1430	18170
1432	18170

Таблица 3 - Фрагмент массива данных, полученных обработкой комбинированного сигнала модулем ядра

Пакет №	Состояние	φ1, мкс	φ2, мкс	φ3, мкс	φ4, мкс	φ5, мкс	φ6, мкс	φ7, мкс	φ8, мкс	Синхروпауза, мкс
N	0	398	400	398	398	398	398	398	398	400
	1	1052	666	1036	1068	652	1070	1070	1068	10478
N+1	0	398	398	398	398	398	398	398	398	398
	1	1052	666	1036	1068	652	1070	1070	1070	10478