АНТИПЛАГИАТ
ОБНАРУЖЕНИЕ ЗАИМСТВОВАНИЙ

Отчет предоставлен сервисом «Антиплагиат» - http://users.antiplagiat.ru

Отчет о проверке

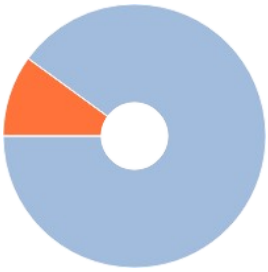
Автор: Charnavok Anastasia

Проверяющий: Charnavok Anastasia

Название документа: Введение_0A_0A B co.txt

РЕЗУЛЬТАТЫ ПРОВЕРКИ

Тариф: FULL



Совпадения: 10,37%

Оригинальность: 89,63%

Цитирования: 0%

Самоцитирования: 0%

«Совпадения», «Цитирования», «Самоцитирования», «Оригинальность» являются отдельными показателями, отображаются в процентах и в сумме дают 100%, что соответствует проверенному тексту документа.

- **Совпадения** — фрагменты проверяемого текста, полностью или частично сходные с найденными источниками, за исключением фрагментов, которые система отнесла к цитированию или самоцитированию. Показатель «Совпадения» – это доля фрагментов проверяемого текста, отнесенных к совпадениям, в общем объеме текста.
- **Самоцитирования** — фрагменты проверяемого текста, совпадающие или почти совпадающие с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа. Показатель «Самоцитирования» – это доля фрагментов текста, отнесенных к самоцитированию, в общем объеме текста.
- **Цитирования** — фрагменты проверяемого текста, которые не являются авторскими, но которые система отнесла к корректно оформленным. К цитированиям относятся также шаблонные фразы; библиография; фрагменты текста, найденные модулем поиска «СПС Гарант: нормативно-правовая документация». Показатель «Цитирования» – это доля фрагментов проверяемого текста, отнесенных к цитированию, в общем объеме текста.
- **Текстовое пересечение** — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
- **Источник** — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
- **Оригинальный текст** — фрагменты проверяемого текста, не обнаруженные ни в одном источнике и не отмеченные ни одним из модулей поиска. Показатель «Оригинальность» – это доля фрагментов проверяемого текста, отнесенных к оригинальному тексту, в общем объеме текста.

Обращаем Ваше внимание, что система находит текстовые совпадения проверяемого документа с проиндексированными в системе источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности совпадений или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

Номер документа: 1

Тип документа: Не указано

Дата проверки: 21.05.2025 00:49:03

Дата корректировки: Нет

Количество страниц: 34

Символов в тексте: 168225

Слов в тексте: 20224

Число предложений: 1788

Комментарий: не указано

ПАРАМЕТРЫ ПРОВЕРКИ

Выполнена проверка с учетом редактирования: Да

Исключение элементов документа из проверки: Нет

Выполнено распознавание текста (OCR): Нет

Выполнена проверка с учетом структуры: Да

Модули поиска: Цитирование, Медицина, ИПС Адилет, СМИ России и СНГ, Коллекция НБУ, Диссертации НББ, Перефразирования по коллекции IEEE, Перефразированные заимствования по коллекции Интернет в английском сегменте, Публикации РГБ, Рувики, Сводная коллекция ЭБС, СПС ГАРАНТ: нормативно-правовая документация, IEEE, Шаблонные фразы, Кольцо вузов, Публикации eLIBRARY (переводы и перефразирования), Публикации eLIBRARY, СПС ГАРАНТ: аналитика, Публикации РГБ (переводы и перефразирования), Патенты СССР, РФ, СНГ, Переводные заимствования по коллекции Интернет в английском сегменте, Переводные заимствования IEEE, Кольцо вузов (переводы и перефразирования), Переводные заимствования по коллекции Гарант: аналитика, Перефразирования по СПС ГАРАНТ: аналитика, Перефразированные заимствования по коллекции Интернет в русском сегменте, Переводные заимствования по коллекции Интернет в русском сегменте, Интернет Плюс, Переводные заимствования

Модули, недоступные в рамках тарифа: Интернет Free

ИСТОЧНИКИ

№	Доля в тексте	Доля в отчете	Источник	Актуален на	Модуль поиска	Комментарий
[01]	5,16%	4,47%	12_100229_1_161696.pdf https://bsuir.by	09 Окт 2024	Перефразированные заимствования по коллекции Интернет в русском сегменте	
[02]	4,41%	0%	12_100229_1_161144.pdf https://bsuir.by	25 Сен 2024	Перефразированные заимствования по коллекции Интернет в русском сегменте	
[03]	4,25%	0%	12_100229_1_161144.pdf https://bsuir.by	29 Авг 2024	Перефразированные заимствования по коллекции Интернет в русском сегменте	
[04]	4,09%	0,12%	12_100229_1_161144.pdf https://bsuir.by	04 Окт 2024	Интернет Плюс	
[05]	4,09%	0%	https://www.bsuir.by/m/12_10022... https://bsuir.by	22 Июн 2023	Интернет Плюс	
[06]	4,09%	0%	12_100229_1_161144.pdf https://bsuir.by	29 Авг 2024	Интернет Плюс	
[07]	4,09%	0%	12_100229_1_161144.pdf https://bsuir.by	25 Сен 2024	Интернет Плюс	
[08]	3,8%	0%	12_100229_1_161144.pdf https://bsuir.by	04 Окт 2024	Перефразированные заимствования по коллекции Интернет в русском сегменте	
[09]	3,37%	0,74%	Учебно-методическое пособие п...	25 Фев 2020	Кольцо вузов (переводы и перефразирования)	
[10]	3,02%	0,25%	Учебно-методическое пособие п...	25 Фев 2020	Кольцо вузов	
[11]	1,26%	0,33%	ДР_Сычик ЭМ 41	08 Сен 2022	Кольцо вузов (переводы и перефразирования)	
[12]	0,93%	0%	МетУказ ТЭО ДП 2020.pdf	22 Ноя 2019	Кольцо вузов	
[13]	0,7%	0,55%	https://kola.opus.hbz-nrw.de/fron... https://kola.opus.hbz-nrw.de	07 Авг 2024	Переводные заимствования	
[14]	0,6%	0,6%	Как работает плата BMS? https://virtustec.ru	29 Ноя 2020	Интернет Плюс	
[15]	0,57%	0,46%	https://www.tyuiu.ru/wp-content/... https://tyuiu.ru	14 Дек 2023	Переводные заимствования по коллекции Интернет в русском сегменте	
[16]	0,5%	0%	Волк Александра Олеговна diplo...	29 Мая 2017	Кольцо вузов	
[17]	0,49%	0,49%	Mixed Media in Evolutionary Art https://ieeexplore.ieee.org	12 Июл 2023	Перефразирования по коллекции IEEE	
[18]	0,45%	0,45%	https://ebuah.uah.es/dspace/bitst... https://ebuah.uah.es	22 Авг 2024	Переводные заимствования	
[19]	0,45%	0%	ДР_Сычик ЭМ 41	08 Сен 2022	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.

[20]	0,42%	0%	https://libeldoc.bsuir.by/bitstrea... https://libeldoc.bsuir.by	19 Ноя 2024	Интернет Плюс	
[21]	0,32%	0,15%	Скоробогатов А.А._ЭЛб-1902а	23 Июн 2023	Кольцо вузов (переводы и перефразирования)	
[22]	0,31%	0,15%	Антиплагиат ВКР Лылов А.	14 Июн 2022	Кольцо вузов (переводы и перефразирования)	
[23]	0,28%	0%	ОПРЕДЕЛЕНИЕ СЕБЕСТОИМОСТ...	21 Дек 2022	Публикации eLIBRARY	
[24]	0,27%	0%	Surface reconstruction - Open3D ... https://open3d.org	03 Фев 2025	Интернет Плюс	Источник исключен. Причина: Маленький процент пересечения.
[25]	0,25%	0%	Нияковский Александр Александ...	25 Мая 2018	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.
[26]	0,24%	0,14%	Челочев Александр Сергеевич З...	02 Июн 2017	Кольцо вузов	
[27]	0,2%	0,2%	00_Studienarbeit_DLR-FK-TBS_Joh... https://elib.dlr.de	08 Ноя 2024	Переводные заимствования	
[28]	0,19%	0,19%	ОПРЕДЕЛЕНИЕ СЕБЕСТОИМОСТ...	21 Дек 2022	Публикации eLIBRARY (переводы и перефразирования)	
[29]	0,19%	0,19%	Моделирование новой архитект...	15 Авг 2022	Публикации eLIBRARY (переводы и перефразирования)	
[30]	0,17%	0,13%	Управление государственным и ... http://ivo.garant.ru	18 Фев 2023	Перефразирования по СПС ГАРАНТ: аналитика	
[31]	0,17%	0,17%	ВОПРОСЫ О СОЕДИНЕНИИ ШАГ...	24 Дек 2022	Публикации eLIBRARY (переводы и перефразирования)	
[32]	0,16%	0%	О НЕКОТОРЫХ ВОПРОСАХ ИССЛЕ...	23 Ноя 2023	Публикации eLIBRARY (переводы и перефразирования)	
[33]	0,16%	0%	О НЕКОТОРЫХ ВОПРОСАХ ИССЛЕ...	17 Июл 2022	Публикации eLIBRARY (переводы и перефразирования)	
[34]	0,16%	0,16%	ТРЕХМЕРНЫЕ ТЕХНОЛОГИИ В ПР...	14 Окт 2024	Публикации eLIBRARY (переводы и перефразирования)	
[35]	0,16%	0%	https://repozitorij.etfos.hr/islando... https://repozitorij.etfos.hr	30 Янв 2025	Интернет Плюс	Источник исключен. Причина: Маленький процент пересечения.
[36]	0,14%	0,14%	200610_b-elne41_2024_1	13 Фев 2025	Кольцо вузов (переводы и перефразирования)	
[37]	0,14%	0,14%	dc dc преобразователь понижаю... https://avito.ru	05 Ноя 2024	Переводные заимствования	
[38]	0,14%	0,14%	ESP32 WROOM DevKit v1: распин... http://wiki.amperka.ru	27 Апр 2024	Интернет Плюс	
[39]	0,14%	0%	Вахрушина М.А. Управленчески... http://ivo.garant.ru	09 Июн 2018	Перефразирования по СПС ГАРАНТ: аналитика	
[40]	0,13%	0%	2021_ИЭИТУС_ТК_270304_БР_Лок...	16 Июн 2021	Кольцо вузов (переводы и перефразирования)	Источник исключен. Причина: Маленький процент пересечения.
[41]	0,12%	0%	РЕАЛИЗАЦИЯ РЕВЕРС-ИНЖИНИР...	22 Ноя 2023	Публикации eLIBRARY (переводы и перефразирования)	Источник исключен. Причина: Маленький процент пересечения.
[42]	0,11%	0%	sbornik_ide-12_tom_1.pdf https://empirya.ru	11 Янв 2025	Переводные заимствования по коллекции Интернет в русском сегменте	Источник исключен. Причина: Маленький процент пересечения.
[43]	0,11%	0%	sbornik_ide-12_tom_1.pdf https://empirya.ru	20 Сен 2024	Переводные заимствования по коллекции Интернет в русском сегменте	Источник исключен. Причина: Маленький процент пересечения.
[44]	0,11%	0%	Ионов В.М. Технологии обработк...	27 Мар 2025	Перефразирования по СПС ГАРАНТ: аналитика	Источник исключен. Причина: Маленький процент пересечения.
[45]	0,11%	0%	Ионов В.М. Технологии обработк...	27 Мар 2025	Переводные заимствования по коллекции Гарант: аналитика	Источник исключен. Причина: Маленький процент пересечения.
[46]	0,1%	0%	Электронный демонстрационны... https://dspace.tltsu.ru	15 Фев 2024	Переводные заимствования по коллекции Интернет в русском сегменте	Источник исключен. Причина: Маленький процент пересечения.
[47]	0,09%	0%	https://www.lirmm.fr/~chaumont/... https://lirmm.fr	06 Окт 2024	Переводные заимствования	Источник исключен. Причина: Маленький процент пересечения.
[48]	0,07%	0%	не указано	13 Янв 2022	Шаблонные фразы	Источник исключен. Причина: Маленький процент пересечения.

Введение

В современном мире технологии для работы с трехмерной графикой постепенно становятся неотъемлемой частью множества областей человеческой деятельности. 3D моделирование и печать, виртуальная и дополненная реальность, а также компьютерная графика, – все это инструменты, позволяющие анализировать и визуализировать объекты в цифровой среде, тем самым оптимизируя производственные процессы и сокращая затраты времени и ресурсов. Одним из самых необычных и в то же время немаловажных инструментов в этой области является сканирование, позволяющее «оцифровывать» физические данные объектов.

Трехмерное сканирование находит широкое применение в самых различных сферах – от инженерии и производства до медицины и искусства. Создание цифровых копий объектов позволяет не только анализировать их форму и текстуру, но использовать полученные данные для дальнейшей обработки, проектирования, а также 3D-печати. Два основных метода сканирования: лазерный и оптический – способствуют быстрому и высокоточному фиксированию геометрических параметров объектов.

Однако процесс 3D сканирования является сложным и ресурсозатратным. Существующие на рынке решения зачастую требуют специализированного дорогостоящего оборудования и программного обеспечения. Кроме того, не все сканеры способны обеспечить удобное управление и автоматизацию процесса, что ограничивает их использование в повседневных и малобюджетных проектах.

Целью данного дипломного проекта является разработка и реализация программно-аппаратного комплекса, позволяющего получать и обрабатывать трехмерные данные объекта.

В соответствии с поставленной целью были определены следующие задачи 26 комплекса:

– фиксация параметров объекта на вращающейся платформе посредством специальных датчиков;

– обработка данных и создание 3D модели в реализованном прикладном ПО;

– предоставление возможности управлять физической установкой и настройки точности получаемых результатов через программу.

Работа над устройством включает следующие этапы:

– разработка схем аппаратной части и выбор необходимых компонентов;

– создание печатных плат;

– разработка программного обеспечения;

– изготовление корпуса и сборка устройства;

– отладка устройства и устранение недочётов.

Данный дипломный проект выполнен мной лично, проверен на заимствования, процент оригинальности составляет XX % (отчет о проверке на заимствования прилагается).

1 Обзор литературы

1.1 Обзор аналогов

Существует множество реализаций 3D-сканеров, которые можно условно разделить на промышленные, бюджетные модели, а также на самодельные устройства, созданные на основе таких микроконтроллеров, как Arduino, и мини-компьютеров, например, Raspberry Pi. Промышленные устройства обеспечивают высокую точность, но требуют больших затрат, в том числе финансовых. Бюджетные модели предлагают компромисс между ценой и качеством, а самодельные решения позволяют снизить затраты, но требуют значительных усилий по разработке и калибровке.

По конструкции 3D-сканеры можно разделить на ручные и стационарные. Ручные модели обеспечивают мобильность и удобство, но могут требовать большей точности в движении оператора. Стационарные модели зачастую используют вращающуюся платформу для ориентации сканируемого объекта, что обеспечивает более высокую точность и автоматизированный процесс сканирования. По принципу работы сканеры бывают лазерными и оптическими. Оптические сканеры обеспечивают более высокое разрешение и лучшую цветопередачу, в то время как лазерные сканеры обеспечивают более быстрое сканирование и лучшую производительность на отражающих или темных поверхностях.

Далее в данном подразделе будут рассмотрены несколько существующих похожих устройств, представленных на рынке.

1.1.1 Ручной 3D сканер EinScan HX – это портативный ручной сканер от китайской компании Shining 3D [1], изображение которого представлено на рисунке 1.1. Компактное ручное устройство позволяет создавать цифровые 3D-копии деталей и кузовных элементов автомобилей и конструктивных элементов самолетов, малых архитектурных сооружений, скульптур, а также предметов мебели.

HX использует новейшую технологию, где сканирование происходит с помощью проецируемого светового паттерна, условно светодиодов, и лазера, второго источника, использующегося для улучшения качества сканирования.

В режиме структурного подсвечивания светодиодами на объект проецируется сетка, по которой программное обеспечение строит цифровую модель. Заявленная точность для данного способа сканирования составляет до 0,05 мм и позволяет сканировать не только детали, но и текстуры на поверхности объекта.

В лазерном сканировании (триангуляции) вместо сетки лазерный луч создает семь перекрещивающихся лучей, и сканирование происходит только по маркерам, повышая точность до 0,04 мм.

Из главных достоинств выделяются:

– возможность сканирования тела и волос человека;

– качественное ПО;

– захват цветовых текстур;

– высокая точность;

– возможность корректировки небольших движений человека.

Ниже представлены основные характеристики:

– скорость сканирования 1 200 000 точек/с (LED) и 480 000 точек/с (лазер);

– скорость выравнивания 20 кадров в секунду (LED) и 55 кадров в секунду (лазер);

– глубина резкости 200–700 мм (LED) и 350–610 мм (лазер);

– расстояние между точками 0,25–3 мм (LED) и 0,5–3 мм (лазер);

– максимальный FOV (область сканирования) 420×440 мм (LED) и 380×400 мм (лазер).

Рисунок 1.1 – Ручной 3D сканер EinScan HX [1]

1.1.2 Настольный сканер EinScan-SE является настольным 3D-сканером нового поколения, разработанный с учетом современных требований к точности,

удобству и универсальности, что идеально подходит для технически не подкованных пользователей за счет предоставления самого простого опыта в 3D сканировании [2]. Сканер представлен на рисунке 1.2.

EinScan-SE предлагает сверхвысокое разрешение сканирования, обеспечивая создание высококачественных 3D-данных с отличной детализацией, что позволяет восстанавливать реальную геометрию объектов. Сканер работает с белым светом в качестве источника, что обеспечивает высокую точность и детализацию даже для сложных объектов. Сам белый свет гарантирует безопасность при его использовании как для детей, так и для взрослых, а также минимизирует влияние на сканируемые объекты.

Сканер полностью совместим с 3D-принтерами, что позволяет быстро и легко печатать отсканированные модели. EinScan SE также предоставляет API для интеграции с 3D-принтерами любого бренда, упрощая процесс передачи данных и настройку печати. Это делает сканер надежным инструментом для тех, кто работает с 3D-печатью и проектированием.

EinScan-SE не подходит для деятельностей, связанных с инженерией и здравоохранением, но тем не менее сканер находит широкое применение в областях от образования и 3D-дизайна до реверс-инжиниринга, анимации, CG, VR и AR.

Из главных особенностей можно выделить:

– автоматическое сцепление сетки с 3D сканирования;

– один скан занимает 8 с;

– сканирование 360° в автоматическом режиме занимает 2 мин;

– автоматический и фиксированный режимы сканирования;

– возможность сканирования объектов как малых, так и больших размеров;

– цветное сканирование с возможностью легкой реконструкции объекта до его оригинальной физической формы;

– облачное хранилище для работы с данными от 3D сканера.

Ниже представлены основные характеристики:

– максимальная область сканирования 200×200×200 мм (автоматический режим) и 700×700×700 мм (фиксированный режим);

– точность сканирования 0,1 мм;

– расстояние до объекта 290–480 мм;

– диапазон одиночного скана 200×150 мм;

– разрешение камеры 1,31 мега пикселей.

Рисунок 1.2 – Настольный сканер EinScan-SE [2]

1.1.3 Ручной сканер MetraSCAN 3D 750 – это высокотехнологичный 3D-сканер от канадской компании Creaform, предназначенный для промышленного использования [3]. Он разработан для решения задач, требующих высокой точности и надежности, таких как контроль качества в цехах, реверс-инжиниринг и прототипирование. MetraSCAN 3D выделяется высокой точностью и возможностью сканирования объектов любых форм и размеров. Представлен на рисунке 1.3.

Этот сканер использует технологию лазерного триангуляционного сканирования, принцип работы которого основан на проецировании лазерных линий на поверхность, с последующей их фиксацией камерами и анализом искажения [18] . На основе этого вычисляется точная 3D-геометрия объекта. и обеспечивается быстрое получение данных с высокой детализацией. MetraSCAN 3D может работать в условиях сложных производственных и лабораторных обстановок, где важно получать точные 3D-модели, даже с учетом нестандартных и сложных поверхностей.

MetraSCAN 3D относится к «англ. Creaform optical CMM (Coordinate Measuring Machine)» (рус. «Координатно-измерительная машина с оптической технологией»). Ее особенностью является возможность использования оптического трекера C-Track, который отслеживает положение сканера и объекта в режиме реального времени.

Одним из наиболее важных преимуществ метрологической измерительной системы является возможность как сканирования, так и проведения измерений контактным или оптическим зондом HandyPROBE, что обеспечивает универсальность использования. Возможности системы сканирования и зондирования могут использоваться независимо или интегрироваться в рабочий процесс измерений.

Основные характеристики представлены ниже:

– точность до 0,030 мм;

– разрешение: 0,050 мм;

– глубина резкости 200 мм;

– рабочее расстояние 300 мм;

– область сканирования 275×250 мм;

– скорость сканирования до 1 000 000 точек в секунду.

Из дополнительных возможностей можно выделить следующие:

– сканирование с помощью 7 перекрещивающихся лазеров;

– совместимость с CAD-системами и интеграция с SolidWorks, CATIA и другими;

– возможность дополнения системой Max Shot 3D Photogrammetry для повышения точности при работе с крупными объектами от 2 до 10 м, что обеспечивает точность до 0,015 мм/м.

Рисунок 1.3 – Ручной сканер MetraSCAN 3D 750 [3]

1.2 Требования к проектируемому устройству

Процесс обработки собираемых сканером данных и их преобразование в формат трехмерного представления является довольно ресурсозатратным. В связи с этим устройство, выполняющее вычисления, должно быть достаточно высокопроизводительным. На данную роль не подходят микроконтроллеры, только полноценные процессоры, из чего вытекает два основных способа реализации проекта. Первый вариант – спроектировать устройство с использованием миникомпьютера, например, Raspberry Pi, что выполняло бы функции как сбора данных, так и их последующей обработки. Второй вариант заключается в создании аппаратного устройства, отвечающего именно за получение параметров объектов, в совокупности с прикладной программой, которая после получения данных отвечала бы за их обработку. Первый вариант идеально подходит для встраиваемых и «самостоятельных» решений. Однако построение на основе миникомпьютера является довольно не гибким, по причине чего принято решение использовать второй способ реализации.

Исходя из поставленных задач, следует, что устройство для 3D сканирования должно иметь конструкцию, обеспечивающую для датчиков максимальный «угол обзора» в 360°. Для этого конструкция должна иметь подвижную платформу. Обычные моторы для точной ориентации в пространстве в виду

«неконтролируемого» угла поворота не подходят. Остаются сервоприводы и шаговые двигатели. Однако, большинство сервоприводов имеют максимальный угол поворота 180°, следовательно, они не обеспечат конструкцию должной «гибкостью». По итогу наиболее предпочтительным вариантом является использование шаговых электродвигателей.

Как было сказано в пункте 1.1 по принципу работы 3D сканеры бывают лазерными и оптическими. Основное отличие вытекает из названия. Первые для получения данных используют лазеры, вторые – цифровые камеры. Также существуют и смешанные решения. В дипломном проектировании будет использован последний способ в виде RGB-D камеры Orbbec Astra, указанной в исходных данных.

Прикладная программа будет представлять собой инструмент, предназначенный в том числе для обработки данных о физических параметрах объекта, полученных с RGB-D камеры. На их основе пользователю будет предоставлена возможность сформировать облака точек объекта, полученных с разных ракурсов, выполнить их сшивку с использованием специализированных алгоритмов, задав необходимые параметры, а также преобразовать полученное облако в полигональную модель (Mesh) и сохранить результат в одном из популярных форматов – PLY, STL или OBJ.

Также с помощью этого ПО пользователь должен иметь возможность задавать настройки аппаратного устройства по беспроводному подключению через протокол Wi-Fi.

В качестве вспомогательных визуальных сигналов для обозначения различных режимов работы аппаратного устройства, а также указателя на изменение его состояния могут служить светодиоды. Для вывода полезной пользователю информации – дисплей. Для управления устройством достаточно использование кнопок и/или резистора переменного осевого.

Проектируемое устройство должно быть портативным. Следовательно, в качестве источника питания необходимо использовать аккумуляторные батареи.

1.3 Обзор технологий

Далее будут рассмотрены задействованные в проекте технологии, среди которых два основных способа 3D сканирования, обзор структуры данных Mesh и формата STL-файла, полезные библиотеки для языка программирования Python.

1.3.1 Подходы к сканированию можно условно разделить на оптическое и лазерное. Оптическое 3D-сканирование использует структурированный свет или фотограмметрию [4]. В процессе сканирования с помощью проектора и двух (иногда одной) камер с разных ракурсов на объект проецируется световой узор (сетка или полосы), а камеры фиксируют деформацию этого рисунка, что дает информацию о глубине и кривизне поверхности, форме предмета. Принцип работы оптического 3D сканирования представлен на рисунке 1.4.

Рисунок 1.4 – Принцип работы оптического 3D сканирования [5]

Структурированные системы освещения могут захватывать тысячи точек одновременно, что делает данный метод значительно быстрее, чем лазерный. Немаловажным преимуществом оптического сканирования является возможность работы с цветными текстурами.

Из недостатков оптических сканеров можно выделить ограничение по сканированию крупных объектов, где необходимо использовать наклейки-маркеры, небольшие черно-белые метки, для создания опорных точек, которые в дальнейшем необходимы для сшивания в единую 3D-модель крупная деталь, отсканированную по частям. Также данный метод требует контроля освещения и может испытывать трудности при сканировании прозрачных или блестящих поверхностей.

Лазерное 3D-сканирование основано на принципе триангуляции и измерении расстояния до объекта с помощью лазерного луча [6]. Лазерный датчик излучает луч, который отражается от поверхности объекта, а система фиксирует угол отклонения и/или время возврата луча, после чего формируется облако точек, каждая из которых содержит информацию о координатах в пространстве 34 , после чего и собирается 3D модель. Принцип

работы лазерного 3D сканирования представлен на рисунке 1.5.

Рисунок 1.5 – Принцип работы лазерного 3D сканирования [5]

Лазерные сканеры менее зависимы от освещения и обеспечивают высокую точность. Также они хорошо справляются со сканированием больших объектов и площадей на расстоянии. Данная способность делает лазерные сканеры незаменимыми в строительстве, топографии и крупномасштабных промышленных приложениях.

Однако данная технология плохо справляется с отражающими и пропускающими свет поверхностями – блестящими, зеркальными и прозрачными.

1.3.2 Структура Mesh является основой для визуализации, редактирования и дальнейшего использования 3D-моделей, состоящей из набора вершин, ребер и граней [7]. Вместе эти элементы определяют форму и детали поверхности объекта. Ключевые компоненты структуры представляют собой:

– вершины: точки в трехмерном пространстве, определяющие форму объекта;

– ребра: отрезки линий, соединяющие пары вершин, образующие границы;

– грани: поверхности, ограниченные множеством ребер, обычно треугольниками или четырехугольниками, которые определяют внешнюю поверхность объекта.

Файл сетки состоит из ряда небольших треугольных граней, которые определяются вершинами и ребрами. Каждая из граней имеет две общие вершины и ребро с соседними граням и определяется координатами трех вершин и вектором нормали, который указывает направление поверхности грани.

Перпендикулярное направление каждого треугольника направлено наружу. При наблюдении за объектом снаружи вершины располагаются в порядке против часовой стрелки, как показано на рисунке 1.6.

Рисунок 1.6 – Строение грани в «Mesh» структуре [8]

Процесс создания Mesh структуры начинается с получения облака точек, представляющего координаты поверхности объекта. Далее на основе этих точек формируется полигональная сетка, которая упрощает геометрию модели и позволяет проводить различные операции, такие как сглаживание, устранение шумов и текстурирование. Пример представлен на рисунке 1.7.

Рисунок 1.7 – Пример полигональной сетки [8]

После формирования полигональной сетки модель можно экспортировать в такие форматы, как STL, OBJ, PLY для дальнейшего использования в САПР системах, рендеринге или 3D печати. Качественная генерация Mesh структуры позволяет добиться высокой детализации и точности воспроизведения объекта.

1.3.3 Формат файла STL (англ. STereoLithography) представляет объект в виде набора треугольных полигонов, описывающих его поверхность. Однако, в отличие от, например, OBJ файла [8], формат не хранит информацию о цветах, текстурах и других атрибутах, что делает его неподходящим для визуализации и сложных рендеров.

STL-файл может быть представлен в двух форматах: бинарном и текстовом. Файл первого типа занимает меньше места и быстрее обрабатывается. Второго – подходит при необходимости переноса данных на компьютер с иным числовым представлением, а также удобен для чтения и редактирования вручную [9].

Текстовый файл начинается с заголовка solid <name>, за которым следует список треугольных граней. Каждая грань описывается вектором нормали и трёхмерными координатами вершин [10]. Координаты заданы в декартовой системе и представлены числами с плавающей запятой:

solid example

facet normal 0.0 0.0 1.0

outer loop

vertex 0.0 0.0 0.0

vertex 1.0 0.0 0.0

vertex 0.0 1.0 0.0

endloop

endfacet

endsolid example

Бинарный файл начинается с заголовка, за которым следует число, указывающее количество треугольников. Далее идут записи о каждой грани, состоящие из координаты нормали, координат трёх вершин и дополнительной информации – атрибутов. Последние на практике обычно не используются и просто заполняются нулями, но в некоторых расширенных версиях STL могут хранить метаданные [11]. Условно содержимое файла может быть изображено следующим образом:

struct stl_bin_format

{

char header [80]; // заголовок - 80 байт

uint32_t num; // число граней - 4 байта

triangle tr_arr [num]; // массив граней - 50 байт * num

};

struct triangle

{

float normal [3]; // вектор нормали - 12 байт

float vertex [3][3]; // координаты вершин - 36 байт

uint16_t attribute; // атрибуты - 2 байта

};

1.3.4 OpenCV [12] (англ. Open Source Computer Vision Library) – это библиотека для обработки изображений и компьютерного зрения, написанная на C++ с возможностью привязки Python, Java и других языков. Также OpenCV способна поддерживать аппаратное ускорение с помощью таких технологий, как OpenCL и CUDA.

Из основных возможностей OpenCV [13] можно выделить:

– фильтрация, морфологические операции, коррекция цветовой гаммы для обработки изображений;

– захват и обработка видео в реальном времени;

– обнаружение объектов, выделение контуров, сегментация в сфере компьютерного зрения;

– интеграция с такими библиотеками глубокого обучения, как TensorFlow, PyTorch, ONNX;

– работа с облаками точек и восстановление глубины сцены.

Хотя сама библиотека не предоставляет средств для взаимодействия с RGB-D камерами напрямую, в контексте работы с данными устройствами, которые предоставляют не только цветное изображение (RGB), но и карту глубины (D), OpenCV позволяет обрабатывать получаемые данные, применять фильтрацию, улучшать качество изображений и создавать трёхмерные представления сцены [14].

Затрагивая работу лишь с RGB-D изображениями, OpenCV предлагает следующий функционал:

– выравнивание RGB и глубины, необходимое по причине аппаратных особенностей камеры, из-за которых получаемые изображения могут быть несовмещёнными, с помощью определенных методов калибровки;

– уменьшение количества шумов, часто содержащихся на картах глубины, посредством специальных алгоритмов фильтрации глубинных данных.

– обнаружение и сегментация, позволяющие точнее выделять объекты на сцене, что особенно полезно для компьютерного зрения и робототехники.

– построение 3D-модели сцены на основе данных глубины путем формирования облаков точек.

Пример обработки цветного изображения и карты глубины представлен на рисунке 1.8.

Рисунок 1.8 – Пример обработки RGB-D данных в OpenCV. Цветное изображение, карта глубины и их совмещение [14]

1.3.5 OpenNI (англ. Open Natural Interaction) [15] – это программный фреймворк с открытым исходным кодом, предназначенный для работы со специализированными аппаратными сенсорами.

На изображении, представленном на рисунке 1.9, видно, что OpenNI выполняет роль связующего слоя между аппаратными устройствами (сенсоры, такие как камеры глубины и микрофоны), промежуточным ПО (middleware-компоненты, например, для трекинга жестов) и конечным приложением, использующим данные от сенсоров. OpenNI получает данные с сенсоров, передаёт их на обработку в middleware-компоненты, а затем передаёт результаты в конечные приложения, такие как игры, браузеры и ТВ-порталы.

Рисунок 1.9 – Устройство систем, задействующих OpenNI [16]

Одной из часто используемых возможностей OpenNI является поддержка трекинга жестов и движений. Используя камеры глубины, OpenNI может идентифицировать и отслеживать основные суставы тела: колени, локти, плечи и голову – в реальном времени, что представлено на рисунке 1.10, а в новой версии 2.0 – распознавать управляющие речевые команды, что делает OpenNI часто используемым решением в робототехнике, дополненной реальности и 3D-сканировании.

Рисунок 1.10 – Пример отслеживания скелета с помощью OpenNI [17]

Кроме работы с жестами, OpenNI выступает в роли SDK и драйвера для некоторых моделей камер глубины, что позволяет разрабатывать программные решения, используя фреймворк для получения и обработки данных глубины, RGB и инфракрасного изображения.

Как упоминалось в пункте 1.3.4 OpenCV не умеет напрямую работать с RGB-D камерами, но может обрабатывать их данные. Для этого ему нужен интерфейс получения информации от сенсора, и таким интерфейсом является OpenNI. Именно OpenNI позволяет OpenCV получать карты глубины и цветные изображения, после чего OpenCV выполняет их обработку.

1.3.6 Open3D [18] – это открытая библиотека, разработанная для работы с 3D-данными, включая облака точек, 3D-модели, изображения и видео. Она предоставляет мощные инструменты для работы с 3D-геометрией, компьютерным зрением, и обработки данных. Пример использования данной библиотеки представлен на рисунке 1.11.

Рисунок 1.11 – Пример проекта, использующего Open3D [18]

Из основных инструментов можно выделить:

– работа с облаками точек, включающая в себя их чтение, визуализацию, фильтрацию, сегментацию и выравнивание;

– построение полигональных сеток, или mesh-структур, из облаков точек при помощи алгоритма Poisson Surface Reconstruction, который восстанавливает поверхность объекта, основываясь на данных точек;

– экспорт и импорт данных за счет поддержки различных форматов файлов для обмена данными, включая «.ply», «.stl», «.obj».

При необходимости формирования 3D-модели объекта, «снятого» на камеру глубины, проводится преобразование нескольких карт глубины, полученных с разных ракурсов, в облака точек. Затем для слияния этих облаков используется алгоритм выравнивания ICP (англ. Iterative Closest Point), который находит оптимальное совпадение между точками разных облаков. После их объединения проводится дополнительная обработка данных, включая фильтрацию и очистку от шума, и выбросов, что позволяет создать точную и чистую 3D-модель объекта 15 .

2 СТРУКТУРНОе проектирование

2.1 Описание основных аппаратных блоков устройства

В контексте структурного проектирования в реализуемом программно-аппаратном комплексе можно выделить следующие аппаратные блоки:

– блок обработки команд, а именно микроконтроллер;

– блок контроля положения платформы, представленный шаговым электромотором;

– блок ручного управления платформой;

– блок отображения информации;

– блок индикации, представленный светодиодами;

– блок питания, состоящий из аккумуляторов, обеспечивающих питание устройства вне сети;

– блок основного управления и обработк 36 и данных;

– блок захвата RGB изображения;

– блок взаимодействия с ИК излучением.

2.1.1 Блок обработки команд, представленный микроконтроллером, является основным элементом части комплекса, на котором выполняется сканирование объекта. Он необходим для организации приёма, анализа и необходимой обработки входящих сигналов управления как с панели самого устройства, так и по беспроводному каналу для последующей передачи команд и преобразованных сигналов на другие блоки комплекса, относящиеся к установке с платформой.

Блок обработки команд получает данные от пользователя, а именно сигналы управления, через блок ручного управления платформой для ее первичной настройки, отвечающей за состояние платформы, и блок основного управления и обработки данных – для вторичной, необходимой для задания параметров сканирования и подключения к сети.

Получая специальные данные, блок обработки команд формирует определенные команды, требующие своего исполнения блоком контроля положения платформы, блоком индикации, а также блоком отображения информации.

2.1.2 Блок контроля положения платформы необходим, чтобы, обрабатывая полученный от пользователя запрос на сканирование, микроконтроллер формировал требуемые к исполнению команды на поворот платформы. Расположенный на ней объект должен быть отсканирован с разных ракурсов, для чего необходимо изменить его позиционирование на определенный угол, за что и ответственен рассматриваемый блок.

Для поворота платформы на нужный угол используется шаговый электродвигатель. После того как проведено сканирование с одного ракурса, шаговый мотор вращает платформу, чтобы выполнить следующее сканирование.

2.1.3 Блок ручного управления платформой предназначен для локального взаимодействия пользователя с устройством без необходимости использования беспроводного соединения. Он включает в себя две кнопки и переменный осевой резистор. Основной функционал блока – предоставление пользователю возможности контролировать состояние устройства и изменять параметры его работы в реальном времени.

Переменный осевой резистор используется для навигации по параметрам из блока отображения информации. Одна из кнопок предназначена для взаимодействия с меню. В разделе, отвечающем за подключение, она позволяет переключать устройство между режимами: подключение к сети пользователя или создание собственной точки доступа, что необходимо для ввода параметров сети. Вторая кнопка служит для ручного запуска и остановки вращения платформы. При доработке устройства м их функционал может быть расширен.

Данный блок позволяет пользователю быстро настраивать платформу без необходимости подключения к внешнему программному обеспечению, что упрощает управление устройством и повышает его автономность.

2.1.4 Блок отображения информации предназначен для текстового представления текущего состояния платформы и параметров её работы. Он реализован в виде дисплея, на котором по выбору пользователя может отображаться статус подключения, прогресс поворота платформы, а также общая информация о платформе.

Режим вывода информация о подключении включает в себя сообщения «Подключено к сети пользователя: [Название]» и «Режим точки доступа. Подключено [Количество] устройств». При выводе информации о процессе сканирования отображается прогресс поворота платформы с сообщением «Позиция X / Y»).

При возможной будущей доработке дополнительно может быть добавлено отображение предупреждений об ошибках, статуса выполнения команд и других вспомогательных данных, что предоставит пользователю возможность более гибкого контроля устройства.

2.1.5 Блок индикации обеспечивает визуальный контроль за состоянием устройства с помощью двух светодиодов. Один из индикаторов сигнализирует о включении питания, подтверждая, что установка готова к работе. Второй отображает ее активность, загораясь во время вращения платформы.

Быстрое мигание второго светодиода может быть использовано для сигнализации о возникновении некорректности в работе устройства.

Такой способ индикации может позволить быстро оценить текущее состояние устройства.

2.1.6 Блок питания является неотъемлемой частью любых электронных устройств. Исходя из необходимости в портативности проектируемого устройства возникает требование использовать автономный режим питания.

Автономное питание подразумевает под собой использование в качестве источника питания аккумуляторных батарей.

Кроме аккумуляторов блок питания должен включать в себя:

– bms (англ. battery management system), представленную платой, с целью защиты аккумуляторов от ухода в глубокий разряд;

– понижающие напряжение платы, необходимые для обеспечения требуемых отдельными элементами уровней напряжений;

– кнопку питания для запуска устройства.

Блок питания должен обеспечивать стабильное и безопасное питание каждого элемента в устройстве платформы.

2.1.7 Блок основного управления и обработки данных представляет собой ПК пользователя, который взаимодействует с платформой через специально разработанное ПО. Этот блок отвечает за настройку всех параметров работы комплекса и управление процессом

сканирования. Через ПО пользователь может подключиться к платформе, задать параметры сканирования.

Данный блок также отвечает за взаимодействие пользователя с RGB-D камерой, которая захватывает изображения в процессе сканирования. После того как пользователь определяет нужные параметры (например, угол поворота или количество снимков), рассматриваемый блок передает команды на платформу для поворота, а также активирует камеру для выполнения снимков.

Блок основного управления и обработки данных является неотъемлемой частью комплексного взаимодействия всех элементов системы. Он объединяет все части системы, синхронизируя работу установки с платформой и камеры для получения точных данных, необходимых для получения 3D модели объекта.

2.1.8 Блок захвата RGB изображения представляет собой цифровую камеру, которая является частью RGB-D камеры и используется для получения цветных изображений сканируемых объектов. Эти изображения помогают улучшать визуализацию 3D модели, а также могут быть использованы для текстурирования модели после формирования основной 3D геометрии на основе данных карт глубины. Данные с RGB камеры также могут служить вспомогательным средством при «сшивании» данных с нескольких ракурсов объекта, предоставляя дополнительную информацию о внешнем виде, которая может быть наложена на 3D модель [13] .

Работа блока RGB камеры тесно связана с блоком глубины, поскольку для полноценного создания 3D модели объект должен быть представлен как в цвете (RGB), так и в глубине (с помощью ИК излучения). Камера RGB выполняет роль детализированного захвата визуальной информации, дополняя данные о форме объекта, полученные с помощью карт глубины. Это взаимодействие обеспечивает более высокое качество получаемых моделей, позволяя точно реконструировать объекты с реалистичными текстурами.

2.1.9 Блок взаимодействия с ИК излучением состоит из инфракрасного проектора и приемника, встроенного в RGB-D камеру. Проектор излучает инфракрасные импульсы, которые, отражаясь от объектов, возвращаются к приемнику камеры. Эти отражения используются для вычисления расстояния до поверхности объекта и создания карты глубины. Инфракрасный излучатель формирует «сетку» на поверхности объекта, где каждый импульс генерирует точку на этой сетке. Эти данные позволяют точно определить форму и размеры объекта, создавая карту глубины, необходимую для дальнейшего формирования 3D модели [18] .

Совокупность «снимков глубины» используется для дальнейшего формирования облака точек, которое в свою очередь послужит для создания детализированной 3D модели объекта. В сочетании с получаемыми цифровыми снимками такой подход сбора данных обеспечивает точность в определении формы объекта, его структуры и размеров, а также улучшает качество реконструированных моделей.

2.1.10 Структурная электрическая схема разрабатываемого комплекса приведена на чертеже ГУИР.400201.099 Э1.

2.2 Описание основных программных блоков устройства

В контексте структурного проектирования в реализуемом программно-аппаратном комплексе можно выделить следующие программные блоки для прикладного ПО на ПК пользователя:

– графический интерфейс;

– управление RGBD камерой;

– обработка полученных от RGBD камеры данных;

– связь с платформой.

Для программы микроконтроллера выделяются следующие программные блоки:

– прием/передача данных;

– контроль положения платформы;

– обработка ввода;

– вывод информации.

2.2.1 Графический интерфейс (GUI) является ключевым элементом взаимодействия пользователя с прикладным программным обеспечением. Он представляет собой визуальную оболочку, которая позволяет пользователю настраивать и контролировать процесс 3D-сканирования объекта с помощью своей системы. Сам интерфейс включает в себя множество компонентов, таких как кнопки, поля ввода, панели состояния и отображение получаемых данных, что делает управление устройством удобным и интуитивно понятным.

В интерфейсе должны быть представлены различные меню и опции, через которые пользователь сможет настраивать угол поворота платформы или количество снимков, которые необходимо сделать, настройки RGBD камеры, подключение к платформе, а также параметры обработки как полученных снимков, так и сформированных облаков точек.

В интерфейсе предусмотрены также элементы для отображения изображений, полученных как от RGB камеры, так и камеры глубины для более точной визуализации объекта. Эти изображения позволяют пользователю отслеживать процесс сканирования объекта в реальном времени и корректировать настройки при необходимости. Информация о текущем статусе камеры и процессе съемки должна помочь пользователю убедиться, что устройство функционирует корректно и в соответствии с заданными параметрами.

Интерфейс должен поддерживать функции сохранения и экспорта для возможности дальнейшей работы с полученными данными, что облегчит их использование в других приложениях или дальнейший анализ.

2.2.2 Блок управления RGBD-камерой отвечает за обнаружение, настройку и контроль работы камеры. Когда необходимо, система должна выполнять проверку подключения камеры, и, если устройство не обнаружено, выдавать пользователю соответствующее уведомление. В противном случае пользователю будет предоставлена возможность задать и настроить необходимые параметры.

Пользователь может задать разрешение для цветового и глубинного изображений. Предусмотрена возможность отключения зеркального отображения изображения. Для обеспечения пространственной согласованности между каналами доступна возможность включения функции совмещения глубины и цвета, для получения согласованных во времени кадров – синхронизации потоков RGB и глубины.

2.2.3 Обработка полученных от RGBD камеры данных начинается после завершения съемки со всех ракурсов. Пользователю предоставляется возможность раздельной обработки изображений и облаков точек, что позволяет более гибко управлять каждым этапом на пути формирования 3D-копии объекта. В обоих случаях возможно как использование предоставляемых средств обработки, так и подключение собственных модулей.

Первым этапом является работа с исходными RGB и depth изображениями. Присутствует возможность корректировать совмещение цветового и глубинного каналов как вручную, так и с использованием внутренних параметров камеры (интринсиков). В процессе обработки применяются: фильтрация для устранения шумов, методы выделения объекта на фоне и формирование основного контура объекта, по которому изолируются глубинные и цветовые данные, относящиеся только к целевому объекту. На основе отфильтрованных данных создаётся RGBD изображение, из которого генерируется облако точек рассматриваемого ракурса с учётом установленных пользователем параметров.

Следующим шагом является формирование облака точек, которое представляет трехмерное распределение координат объекта. Процесс включает сшивание ракурсов и выравнивание точек, что позволяет создать целостную геометрическую модель. Предусматриваются средства для предобработки облаков, включающие фильтрацию, удаление шумов, даунсэмплинг, или разрежение облака точек, и расчёт нормалей поверхности. Для совмещения облаков могут быть использованы методы как глобальной регистрации, так и точной локальной коррекции (ICP). Завершающим этапом является постобработка объединённого облака точек, включающая, при необходимости, дополнительную фильтрацию и сглаживание. Результат может быть преобразован в полигональную модель, пригодную для визуализации или последующего экспорта.

2.2.4 Блок связи с платформой представляет собой условный интерфейс, который обеспечивает коммуникацию между прикладным программным обеспечением и установкой с вращающейся платформой. Его основная задача является установка соединения совместно с обеспечением возможности обмена данными между ними для выполнения различных операций и функций.

В процессе подключения существуют два возможных сценария. В первом случае микроконтроллер работает как точка доступа, к которой пользователь подключается через приложение, вводя данные своей сети, а именно SSID и пароль. После того как микроконтроллер подключается к сети, прикладное ПО автоматически обнаруживает его, и становится возможным дальнейшее взаимодействие между устройствами. После приложение отправляет команды микроконтроллеру, включающие в себя информацию о дальнейшем процессе сканирования, и запросы на поворот платформы, что происходит после того, как камера сделает снимок.

Именно блок связи отвечает не только за управление устройством, но и за дистанционную синхронизацию работы компонентов разрабатываемого комплекса со стороны прикладного ПО.

2.2.5 Блок приема и передачи данных обеспечивает стабильную беспроводную связь между микроконтроллером и устройством пользователя с прикладным ПО.

Когда микроконтроллер находится в режиме точки доступа, он ожидает подключения устройства пользователя для получения данных о его сети. После успешного выполнения предыдущего действия пользователь может ввести имя и пароль своей сети, которые будут сохранены во внутренней памяти микроконтроллера.

При переключении режима соединения установка использует сохраненные данные для подключения к заданной сети. После установления соединения платформа становится доступной для взаимодействия, начиная принимать команды на поворот платформы или обновление данных о процессе сканирования. Таким образом, код управляет синхронизацией работы комплекса, обеспечивая точную координацию между всеми её компонентами со стороны микроконтроллера в установке с вращающейся платформой.

2.2.6 Блок контроля положения платформы отвечает за точное вращение шагового электродвигателя, обеспечивая достижение требуемых углов поворота. В данном блоке микроконтроллер принимает управляющую команду, анализирует заданный угол и вычисляет необходимое количество шагов для двигателя. Затем, с учетом полученных данных, он поочередно активирует нужные обмотки двигателя, задавая как направление, так и скорость вращения.

После завершения каждого поворота блок формирует сообщение об успешном выполнении команды, позволяя прикладному ПО своевременно получать актуальную информацию о состоянии.

2.2.7 Блок обработки ввода отвечает за обработку сигналов от элементов управления, позволяя пользователю напрямую взаимодействовать с установкой с вращающейся платформой. Микроконтроллер отслеживает нажатие кнопок и изменение положения осевого переменного резистора, преобразуя аналоговые и дискретные сигналы в управляющие команды, что позволяет в реальном времени настраивать устройство и корректировать процесс его работы.

2.2.8 Блок вывода информации предназначен для информирования пользователя о текущем состоянии устройства. Микроконтроллер управляет светодиодами и дисплеем, обеспечивая визуальную обратную связь.

Микроконтроллер взаимодействует с этими элементами через специальные команды, которые устанавливают необходимые логические уровни на соответствующих выводах, тем самым управляя состоянием светодиодов и отображением текстовой информации на дисплее и обеспечивая пользователя своевременной информацией, описанной в пунктах 2.1.4 и 2.1.5.

2.2.9 Структурная схема разрабатываемого комплекса приведена на чертеже ГУИР.400201.099 С1.

3 Функциональное проектирование

3.1 Аппаратные блоки устройства

3.1.1 Микроконтроллер играет центральную роль в установке с вращающейся платформой, обеспечивая выполнение ее основных функций и управление периферийными устройствами. Ниже рассматриваются два самых популярных решения, используемых вне промышленного производства.

На рынке микроконтроллеров самыми популярными являются микроконтроллеры ATmega компании Atmel. Из их особенностей можно выделить:

– минимальный уровень энергопотребления;

– высокая производительность;

– простота программирования.

Данные микроконтроллеры пользуются спросом как у новичков, только осваивающих азы программирования микроконтроллеров, так и у профессионалов, использующих микроконтроллеры ATmega как в домашних, так и стартап проектах.

Пример микроконтроллера компании Atmel ATmega8 [19] представлен на рисунке 3.1.

Рисунок 3.1 – Микроконтроллер ATmega8 в корпусе DIP [19]

Большой популярностью также пользуются микроконтроллеры ESP компании Espressif Systems.

ESP – это семейство недорогих микроконтроллеров, самой главной особенностью которых являются интегрированные модули радиосвязи Wi-Fi, Bluetooth и Thread. Обладают большими производительными способностями по сравнению с микроконтроллерами ATmega.

Микроконтроллеры возможно использовать как на самодельных платах, так и на готовых заводских. Пример микроконтроллера Esp32-Wroom-32 на отладочной плате [20] представлен на рисунке 3.2.

Сравнение характеристик микроконтроллеров Esp32-Wroom-32 [21] и ATmega328P представлено в таблице 3.1

Рисунок 3.2 – Плата с микроконтроллером Esp32-Wroom-32 [20]

Таблица 3.1 – Сравнение характеристик микроконтроллеров ESP32 и ATmega328P

Микроконтроллер Esp32-Wroom-32 ATmega328P

Архитектура процессора RISK RISK

Разрядность процессора, бит 32 8

Тактовая частота, МГц 240 16

ОЗУ, КБ 520 2

EEPROM, КБ Отсутствует 1

Число циклов перезаписи EEPROM – 100000

Flash-память, КБ 4000 32

Число циклов перезаписи FLASH-памяти 10000 10000

Рабочее напряжение, В 3,3 5

Количество цифровых выводов 34 14

Количество аналоговых выводов 18 6

Количество выводов с поддержкой ШИМ 28 6

Максимальный ток с выводов, мА 40 40

Поддерживаемые интерфейсы UART, GPIO, ADC, DAC, SDIO, SD card, PWM, I2C, I2S UART, I2C, SPI

Несмотря на то, что микроконтроллер ATmega328P более энергоэффективен и может быть дополнен модулем Wi-Fi, Esp32-Wroom-32 является более предпочтительным выбором в данном проекте. Во-первых, он уже оснащен встроенным Wi-Fi, что упрощает реализацию беспроводной связи. Во-вторых, ESP32 значительно превосходит ATmega328P по вычислительной мощности, имеет два ядра, более высокую тактовую частоту и значительно больший объем памяти.

Одной из плат, на которой установлен микроконтроллер ESP32, является плата Esp32-WROOM-32 DevKit v1. Она представляет собой компактный и удобный для интеграции модуль, оснащенный встроенной антенной и разъемами для внешних подключений. Поддерживается программирование через UART, а наличие встроенного стабилизатора питания позволяет подключать модуль к источникам напряжения 3,3В. На отладочной плате расположены две тактовые кнопки: «Boot» – для ручного запуска режима прошивки и «Reset» – для ручного перезапуска контроллера. Также на плате присутствуют индикатор питания и адресный RGB-светодиод.

Параметры отладочной платы Esp32-WROOM-32 DevKit v1 представлены в таблице 3.2.

Таблица 3.2 – Характеристики платы Esp32-WROOM-32 DevKit v1

Плата Esp32-WROOM-32 DevKit v1

Напряжение питания, В 5–14

Выходное напряжение, В 3,3

Рабочее напряжение, В 3,3

Количество цифровых выводов 21 (вход/выход) + 4 (только вход)

Количество выводов с наличием АЦП 15

Количество выводов с поддержкой ШИМ 21

Максимальный потребляемый ток 21 , мА 500

Интерфейсы UART, I2C, SPI, I2C

Интерфейс для прошивки и отладки USB

Габариты, мм 51×28

3.1.2 С целью обеспечить установку с платформой возможностью вращения на 360° используются шаговые электродвигатели.

Шаговые электродвигатели – это специальный тип двигателей, которые вращают свой вал «шаг за шагом». Важным преимуществом шаговых двигателей является их способность перемещаться на точно заданный угол (или шаг), что делает их идеальным выбором для применения в проектах, где требуется точное управление.

Устройство и принцип работы шаговых двигателей основаны на использовании электромагнитов для создания вращающего момента. Шаговый электродвигатель состоит из ротора (обычно имеющего перечень зубцов или зубчатый диск) и набора статорных обмоток, которые контролируются в определенной последовательности для достижения вращения с заданными углом и скоростью. Эта последовательность подается с помощью электрических импульсов, изменяющих состояние электромагнитов в статоре. Путем изменения последовательности подачи электрических сигналов становится возможным точно контролировать угол вращения или перемещения шагового двигателя.

В связи со своей универсальностью шаговые двигатели бывают самых разных форм и размеров. Как вариант для использования были взяты две наиболее распространенные модели, используемые в любительских проектах.

Самым популярным компактным вариантом является униполярный шаговый двигатель 28BYJ-48 [22], представленный на рисунке 3.3.

Рисунок 3.3 – Шаговый электродвигатель 28BYJ-48 [22]

Для управления шаговым двигателем 28BYJ-48 используют один из двух режимов подключения, использующих ранее описанный принцип:

– полношаговый режим – 4 ступени импульсов на 1 шаг;

– полушаговый режим – 8 ступеней импульсов на 1 шаг 31 .

Визуализация второго режима представлена на рисунке 3.4.

Рисунок 3.4 – Визуализация второго режима работы 28BYJ-48 [22]

Популярным вариантом мощного шагового электродвигателя является модель 17HS8401 [23], внешний вид которой представлен на рисунке 3.3.

Рисунок 3.5 – Шаговый электродвигатель 17HS8401 [24]

Модель 17HS8401 имеет типоразмер Nema17 и относится к шаговым двигателям с высоким крутящим моментом. Основное назначение – привод небольших станков (в том числе ЧПУ), 3D-принтеров и плоттеров.

17HS8401 является биполярным шаговым двигателем, что означает, что у него две обмотки, и для его работы необходимо поочередное переключение направления тока на каждой из них. Подобная деталь позволяет биполярным двигателям обеспечивать более высокий крутящий момент по сравнению с униполярными при том же размере Условное изображение обмоток представлено на рисунке 3.6.

Рисунок 3.6 – Условное обозначение обмоток 17HS8401 [25]

Характеристики обоих шаговых двигателей представлены в таблице 3.3.

Таблица 3.3 – Характеристики рассматриваемых электродвигателей

Шаговый электродвигатель 17HS8401 28BYJ-48

Напряжение питания, В 10–24 5 или 12

Максимальный потребляемый ток, А 1,7 0,32

Номинальный ток, А 1,68 0,10

Число фаз 2 4

Количество шагов ротора 200 64

Крутящий момент, кг·см 5,3 0,45

Для повышения качества и надежности платформы принято решение использовать более мощный шаговый электродвигатель Nema 17 17HS8401.

По ряду причин не принято подключать шаговый двигатель непосредственно к микроконтроллеру. Для управления мотором используется специальная микросхема – драйвер. Причины его использования заключаются в следующем:

– драйверы обеспечивают защиту контроллера в связи с возможным увеличением тока на обмотках шагового мотора;

– управление током работает и в обратную сторону, так как, чтобы корректно выполнять «шаги», шаговые двигатели требуют точного контроля тока;

– драйвер необходим для управления скоростью и задания необходимого числа шагов, на которое необходимо повернуть вал;

– обеспечение защиты мотора от перегрева и короткого замыкания;

– использование драйвера позволяет управлять шаговым мотором с помощью микроконтроллера или другого устройства, используя стандартные сигналы управления, такие как STEP (шаг), DIR (направление) и другие.

Для выбранного шагового мотора принято решение использовать драйвер DRV8825 [26], плата которого представлена на рисунке 3.7.

Рисунок 3.7 – Плата драйвера DRV8825 [26]

DRV8825 – это драйвер для управления биполярными шаговыми двигателями, который имеет встроенный преобразователь для удобства работы, который позволяет управлять шаговым двигателем с помощью всего лишь 2 контактов микроконтроллера, один из которых предназначен для управления направлением вращения, а другой – для управления шагами [29] .

Данный драйвер обеспечивает пять различных ступенчатых режимов: полный, средний, четвертьступенчатый, восьмиступенчатый и шестнадцатиступенчатый. Кроме того, он оснащен потенциометром для регулировки выходного тока, теплового отключения при перегреве и защиты от перекрестного тока. С помощью потенциометра ограничим ток для выбранного электродвигателя до 1.7А. Для обеспечения эффективного теплоотведения и предотвращения перегрева, рекомендуется использовать радиатор, особенно при высоких токах нагрузки.

Из характеристик можно выделить:

– напряжение питания: от 5 В до 45 В;

– максимальный ток: до 2,5 А (пиковый ток на обмотку)

– максимальный ток, потребляемый платой: 10мА.

3.1.3 Для отображения текстовой информации принято решение использовать монохромный 0.96-дюймовый OLED-дисплей модуль DSM-OLEDv2-0.96-4P-YB [27], представленный на рисунке 3.8. Его особенностью является работа без подсветки, и связанное с этим, низкое электропотребление. Выбранная модель является двухцветной с доступными желтым и синим цветами.

Данный дисплей имеет компактные размеры, а также обладает высокой четкостью изображения и удобством использования, что делает его отличным и наиболее предпочтительным выбором для данного проекта.

Рисунок 3.8 – Модуль DSM-OLEDv2-0.96-4P-YB [27]

Из характеристик можно выделить:

– разрешение: 128×64;

– напряжение питания: 3,3/5В;

– максимальный ток потребления: 50мА;

– интерфейс: I2C.

3.1.4 Источником питания для проектируемой установки с вращающейся платформой будут являться аккумуляторы, для выбора которых необходимо провести анализ потребления энергии элементами устройства. Напряжение, ток и мощность, потребляемые отдельными компонентами устройства, представлены в таблице 3.4.

Таблица 3.4 – Затраты энергии

Элемент Напряжение питания, В Максимальный потребляемый ток, мА Потребляемая мощность, Вт

Esp32-WROOM-32 DevKit v1 5 500 2,5

NEMA 17 17HS8401 12 1700 20,4

DRV8825 5 10 0,05

DSM-OLEDv2-0.96-4P-YB 5 50 0,25

Максимально потребляемая мощность: 23,2 Вт

Потребляемая отдельными элементами мощность рассчитывалась по формуле

Рпотр=Iмакс потр Uпит .

(3.1)

Максимально потребляемая мощность была получена суммированием всех потребляемых отдельными элементами.

Далее необходимо рассчитать значение максимального выходного тока блока питания с помощью формулы

IБП макс потр =Рмакс потрUмакс пит .

(3.2)

Получаем IБП макс потр, равное 1,94. Далее необходимо увеличить полученное значение на 20 %, чтобы обеспечить запас по току:

I БП макс потр*=1.2IБП макс потр=2.32А.

(3.3)

Таким образом, вновь рассчитывая значение мощности по формуле (3.1), получаем РБП макс потр, равное 27,84 Вт.

Теперь необходимо подобрать аккумулятор. Хорошим вариантом является Li-ion аккумулятор 18650 с максимальным током разряда 20 А. Номинальное напряжение у данных аккумуляторов 3.7 В. Соединив последовательно два таких аккумулятора, получаем блок питания подходящей мощности.

3.1.5 Необходимость в защите обусловлена тем, что Li-ion аккумуляторы наряду с многочисленными достоинствами имеют слабую сторону – чувствительность к перезаряду выше 4,2 В и разряду ниже допустимой границы в 2,5 В на элемент. Поэтому для безопасного использования литий-ионные батареи снабжаются системами контроля заряда-разряда – BMS платами управления, обеспечивающими защиту и балансировку элементов питания в сборке.

BMS – это электронная система, управляющая зарядно-разрядными процессами в автономном источнике питания, которая:

– обеспечивает безопасную работу батареи при подключении и отключении нагрузки и ЗУ;

– в ходе зарядки АКБ распределяет токи между ячейками;

– контролирует их температуру, напряжение и сопротивление;

– отслеживает разрядный ток;

– выполняет балансировку (равномерное распределение энергии) аккумуляторов в сборке;

– обеспечивает их защиту от перенапряжения, КЗ, токовых перегрузок, перегрева, критического разряда и других опасных состояний.

Поэтому с целью безопасного использования АКБ и по максимуму увеличивания срока ее службы 14 в проекте используется специальная плата BMS-3S 40A с балансировкой, представленная на рисунке 3.9.

Рисунок 3.9 – Плата BMS-3S 40A с балансировкой [28]

3.1.6 Блок питания в виде трех аккумуляторов Li-ion 18650 обеспечивает напряжение питания в районе от 11 В до 12,6 В.

Опираясь на данные из пункта 3.1.4, можно выделить, что шаговый мотор может питаться напрямую от аккумуляторов. Плате с микроконтроллером, драйверу и дисплею необходимо напряжение питания, равное 5В.

Для устройств, электропитание которых зависит от батареек или аккумуляторов, изменение напряжения до требуемой величины возможно с использованием DC/DC преобразователей – электронных схем или электромеханических устройств, обеспечивающих изменение выходного напряжения источника тока как в большую, так и в меньшую сторону.

Преобразователи напряжения бывают линейные и импульсные.

Линейные преобразователи работают по принципу использования управляемого элемента, который регулирует путь тока от входа к выходу с определенным напряжением. Преобразование напряжения происходит путем рассеивания избыточной энергии в виде тепла.

Одно из основных преимуществ линейных преобразователей – это их простота и низкая стоимость. Линейные стабилизаторы также проще для повторения, и часто обеспечивают меньший уровень помех и пульсаций.

Однако их эффективность невысока, особенно при больших различиях входного и выходного напряжения. Это означает, что они могут быть неэффективными и генерировать большое количество тепла при работе с высокими нагрузками.

Импульсный преобразователь – устройство, элемент управления которого функционирует в импульсном состоянии, постоянно замыкаясь и размыкаясь. Благодаря данной особенности работы подача тока происходит порционно. Эти преобразователи имеют более высокий КПД и малую тепловую нагрузку по сравнению с линейными, особенно при больших различиях входного и выходного напряжения.

Именно импульсные преобразователи включают в себя различные типы, такие как Step-Down (или Buck), Step-Up (или Boost), Step-Up/Step-Down (или Buck-Boost) и другие, которые обеспечивают различные сочетания входных и выходных напряжений.

Для получения 5В для питания элементов и их логики используется плата линейного преобразователя напряжения DC-DC MINI-360, представленная на рисунке 3.10 со следующими характеристиками:

– входное напряжение: 4.75–18 В;

– выходное напряжение: 0,9–15 В;

– выходной ток: 1,8 А(рабочий), 3 А (пиковый);

– эффективность: 96 % (макс).

Выбор данной платы обусловлен отсутствием генерации в питании шумов и помех, к которым чувствительны управляющая логика и особенно Esp32-WROOM-32.

Подобные схемы, оснащенные радиомодулями и Wi-Fi модулями, имеют строгие требования к стабильности напряжения и тока, так как работают на довольно высоких частотах и могут быть чувствительны к электромагнитным помехам (EMI) или помехам, которые могут присутствовать в источнике питания. Недостаточно чистое или стабильное питание может вызвать нежелательные помехи, что приведет к плохому качеству сигнала или даже потере связи при использовании беспроводного соединения.

Рисунок 3.10 – Изображение платы DC-DC MINI-360 [29]

3.1.7 Orbbec Astra [30] – это 3D-датчик глубины, использующий технологию Structured Light (структурированное освещение) для захвата информации о расстоянии до объектов. В основе работы лежит проектор, который отображает инфракрасный узор на объекте, а камера анализирует искаженные изображения этого узора, определяя расстояние до поверхности. RGBD камера представлена на рисунке 3.11.

Рисунок 3.11 – 3D-датчик глубины Orbbec Astra [30]

Для обработки данных Orbbec Astra использует собственный ASIC (англ. Application-Specific Integrated Circuit), специализированный чип, оптимизированный для вычислений, связанных с анализом 3D-глубины, что значительно повышает производительность и уменьшает нагрузку на центральный процессор.

Схематичное представление камеры изображено на рисунке 3.12. Из основных компонентов для сбора данных можно выделить следующие:

1 Проектор. Используется для отображения инфракрасного узора на сканируемом объекте. Этот узор необходим для вычисления глубины путём анализа его искажений.

2 IR-сенсор. Это камера, которая захватывает искажения инфракрасного узора, что позволяет вычислить расстояние до объектов.

3 Система защиты глаз, которая фильтрует нежелательное инфракрасное излучение, предотвращая его попадание в зону видимости пользователя, что важно для безопасного использования устройства.

4 RGB-камера. Захватывает цветные изображения, которые можно комбинировать с данными о глубине для получения полноцветных 3D-изображений.

Рисунок 3.12 – Схематичное представление Orbbec Astra

Камера обладает следующими техническими характеристиками:

– длина волны: 850 нм;

– диапазон измерения глубины: 0,6–8 м (точность ±3 мм);

– разрешение датчика глубины:640×480;

– угол обзора датчика глубины: H58,4° V45,5°;

– разрешение RGB-камеры: 640×480;

– угол обзора RGB-камеры: H63,1° V49,4°;

– частота кадров: 30 fps;

– инерциальный измерительный блок (IMU): отсутствует;

– интерфейс USB 2.0 Type-A;

– триггер (синхронизация съемки): отсутствует;

– потребляемая мощность: в среднем <2,4 Вт;

– условия эксплуатации: 10–40°, в помещении, 10–85 % влажности.

3.1.8 Функциональная электрическая схема разрабатываемого комплекса приведена на чертеже ГУИР.400201.099 Э2.

3.2 Программные блоки устройства

3.2.1 Графический интерфейс пользователя (GUI) является связующим звеном между пользователем и программным обеспечением, обеспечивая доступ к функциям разрабатываемой системы в удобной визуальной форме. Интерфейс реализован с использованием библиотеки PyQt6, предоставляющей инструменты для разработки полноценных графических настольных приложений с кроссплатформенной поддержкой.

Ключевым элементом структуры интерфейса является класс QMainWindow, выступающий в роли основного окна приложения, которое разделено на центральную область и строку состояния. Внутри основного окна размещён виджет QTabWidget, реализующий систему вкладок для навигации между основными функциональными модулями программы.

Для взаимодействия с пользователем активно используются стандартные элементы ввода и отображения информации, включающие в себя:

– QPushButton – элемент для запуска команд по нажатию кнопки;

– QCheckBox – «булевый» флажок для переключения параметров;

– QLabel – компонент для отображения текстовой или графической информации;

– QLineEdit – однострочное текстовое поле для ввода данных;

– QSpinBox/QDoubleSpinBox – числовые поля с возможностью пошагового изменения значения (целые и вещественные числа);

– QComboBox – выпадающий список для выбора одного из заранее заданных вариантов;

– QTextBrowser – элемент для отображения текстовой информации, включая форматированный текст и ссылки;

– QTableWidget – таблица с возможностью отображения и редактирования данных в табличной форме.

Для гибкой и адаптивной компоновки элементов используются Layout-менеджеры, а именно QVBoxLayout, QHBoxLayout и QGridLayout, а также QSpacerItem для регулирования пространства между элементами графической оболочки. Именно данные компоненты обеспечивают масштабируемость интерфейса при изменении размеров окна.

В местах, где предполагается размещение большого количества элементов, применяется QScrollArea, позволяющая прокручивать содержимое вкладки при нехватке видимой области.

Виджет QTabWidget «делит» окно на пять вкладок:

– «Scanning»;

– «Platform»;

– «Cam Check»;

– «Frames Processing»;

– «Clouds Processing».

Вкладка «Scanning» представлена на рисунке 3.13. В ней пользователь имеет возможность установить настройки для процесса сканирования, среди которых:

– выбор папки для сохранения снимков;

– ввод имени сканируемого объекта для формирования имен файлов;

– кнопка для сохранения фона;

– «флажок» для открытия/закрытия окон с выводом камеры;

– ввод скорости и ускорения платформы, количества необходимых снимков и угла поворота;

– кнопка для начала сбора снимков и прерывания в процессе.

Рисунок 3.13 – Вкладка «Scanning» графического интерфейса

Вкладка «Platform» представлена на рисунке 3.14. В ней пользователь имеет возможность подключиться к платформе и настроить параметры соединения, среди которых:

– порт для подключения;

– IP-адрес;

– SSID (название) точки доступа пользователя;

– пароль для подключения к точке доступа пользователя.

Рисунок 3.14 – Вкладка «Platform» графического интерфейса

Вкладка «Cam Check» представлена на рисунке 3.15. В ней пользователь имеет следующие возможности:

– проверить вывод камеры;

– задать настройки камеры;

– сделать и сохранить тестовые фото в выбранном каталоге под заданным именем.

Настройки камеры будут рассмотрены более подробно в подразделе 3.2.2.

Рисунок 3.15 – Вкладка «Cam Check» графического интерфейса

Вкладка «Frames Processing» представлена на рисунке 3.16 и предназначена для ввода данных и параметров для получения облаков точек из RGBD данных. Более подробно процесс обработки и соответствующие параметры будут рассмотрены в подразделах 3.2.3 и 6.2.5.

На вкладке пользователь может:

– указать необходимый каталог и имя сканируемого объекта, на основе чего будут выбраны нужные файлы;

– вручную скорректировать сдвиг и растяжение изображения depth камеры относительно RGB, если аппаратного согласования недостаточно;

– включить/выключить визуализацию для проверки корректности выделения объекта на фоне, как в видеорежиме с выводом камеры в реальном времени, так и в фоторежиме, где выбирается фото с заданного угла

– включить/выключить отображение контура объекта;

– выбрать интринсики какой камеры использовать (или обеих) и задать их значения;

– выбрать количество облаков для визуализации после обработки;

– задать расстояние, объекты, лежащие дальше которого будут отсечены;

– задать размер морфологического ядра для сглаживания;

– выбрать, визуализировать ли оси при отображении облаков;

– выбрать каталог, куда сохранить результат и указать его имя;

– указать использовать пользовательскую обработку.

Вкладка «Clouds Processing» представлена на рисунке 3.17. В ней пользователю представлены инструменты для работы с облаками точек.

В верхнем левом углу находится кнопка, с помощью которой пользователь сможет открыть 3D пространство, в котором будут отображены все выбранные облака точек. Оставшееся пространство в левой части можно поделить на несколько модулей:

– «Upload & Filtration» для задания настроек фильтрации, сдвига и загрузки облаков;

– «Manual Adjustment» для ручной корректировки положения;

– «Processing & Merging» для задания настроек предобработки, регистрации и постобработки «сшиваемых» облаков;

– «Save & Export» для задания настроек сохранения и экспорта результатов.

Более подробно процесс обработки и соответствующие параметры будут рассмотрены в подразделах 3.2.3 и 6.2.6.

Рисунок 3.16 – Вкладка «Frames Processing» графического интерфейса

Рисунок 3.17 – Вкладка «Clouds Processing» графического интерфейса

В верхнем правом углу вкладки «Clouds Processing» находится таблица, строки которой содержат имена всех загруженных облаков точек, а также результатов регистрации и «флажки» для отображения облаков в 3D пространстве и указания тех, что необходимо «сшить». В нижнем правом углу находится текстовое поле для логирования результатов.

Так как для удобства пользователя задействованы «прокручиваемые» области на рисунке 3.17, интерфейс вкладки «Clouds Processing» отображен не полностью. На рисунке 3.18 изображены скрытые модули «Manual Adjustment» и «Save & Export». На рисунке 3.19

представлены не отображенные части модуля «Processing & Merging».

Рисунок 3.18 – Скрытые модули вкладки «Clouds Processing»

Рисунок 3.19 – Не отображенные части модуля «Processing & Merging»

3.2.2 Блок управления RGBD-камерой отвечает за инициализацию, настройку и получение данных с устройства. В данном проекте для работы с Orbbec Astra используется фреймворк OpenNI2 через модуль primesense для Python. Для работы с OpenNI2 необходимо установить соответствующую библиотеку с помощью команды «pip install primesense», а также добавить в корень проекта содержимое папки Orbbec_OpenNI_v2.3.0.86-beta6_windows_release, взятой из официальной сборки Orbbec [31]. Эта папка содержит необходимые библиотеки (.dll, .lib) и конфигурационные файлы (.ini) для корректной работы с устройством.

Перед началом работы с камерой необходимо инициализировать модуль OpenNI2 с помощью команды openni2.initialize(), которая загружает драйверы и подготавливает систему к взаимодействию с устройством [32].

Подключение камеры осуществляется вызовом команды openni2.Device.open_any(), которая позволяет найти и открыть первое доступное устройство. После успешного подключения необходимо создать потоки depth_stream для получения данных о глубине и color_stream – для цветных изображений:

depth_stream = dev.create_depth_stream()

color_stream = dev.create_color_stream()

Перед началом захвата данных задаются настройки потоков: формат, разрешение и частота кадров. Для потока глубины применяется формат PIXEL_FORMAT_DEPTH_1_MM, при котором значения соответствуют расстоянию в миллиметрах, для цветного – формат PIXEL_FORMAT_RGB888, соответствующий стандартному 24-битному RGB изображению. Разрешение определяется техническими характеристиками камеры.

depth_stream.set_video_mode(openni2.VideoMode(

pixelFormat=openni2.PIXEL_FORMAT_DEPTH_1_MM,

resolutionX=640,

resolutionY=480,

fps=30))

color_stream.set_video_mode(openni2.VideoMode(

pixelFormat=openni2.PIXEL_FORMAT_RGB888,

resolutionX=640,

resolutionY=480,

fps=30))

Для аппаратного совмещения каналов (Depth to Color), наложения карты глубины на цветное изображение с целью обеспечения пространственной согласованности данных может быть использована команда:

dev.set_image_registration_mode(openni2.IMAGE_REGISTRATION_DEPTH_TO_COLOR)

Для включения аппаратной синхронизации, обеспечивающей одновременный захват RGB и глубины, нужна команда:

dev.set_depth_color_sync_enabled(True)

Для управления зеркальным отображением, которое камера включает по умолчание, могут быть использованы команды:

depth_stream.set_mirroring_enabled(False)

color_stream.set_mirroring_enabled(False)

После настройки параметров могут быть запущены потоки:

depth_stream.start()

color_stream.start()

Захват изображений осуществляется поочерёдным считыванием кадров из каждого потока. Полученные данные представляют собой буферы, которые преобразуются в массивы NumPy и подготавливаются к дальнейшей обработке:

depth_frame = depth_stream.read_frame()

color_frame = color_stream.read_frame()

depth_data = depth_frame.get_buffer_as_uint16()

color_data = color_frame.get_buffer_as_uint8()

depth_raw = np.frombuffer(depth_data, dtype=np.uint16).reshape((480, 640))

color_raw = np.frombuffer(color_data, dtype=np.uint8).reshape((480, 640, 3))

depth = depth_raw[0:user.resolutionY, 0: user.resolutionX]

color = color_raw[0:user.resolutionY, 0: user.resolutionX]

color_bgr = cv2.cvtColor(color, cv2.COLOR_RGB2BGR)

3.2.3 Обработка полученных от RGB-D камеры данных условно делится на два этапа, где первый – предварительная работа с RGB и depth снимками и формирование облаков точек, а второй – их обработка [33]. Более подробно алгоритмы и параметры рассматриваются в разделе «Разработка программного обеспечения», в данном подразделе описываются применяемые программные средства и ключевые методы.

Основные библиотеки: OpenCV и Open3D. Для их использования предварительно требуется установить зависимости с помощью следующих команд:

pip install opencv

pip install open3d

На первом этапе RGB и глубинные снимки преобразуются в облако точек, описывающее объект с определенного ракурса, для чего используются следующие методы OpenCV:

– cv2.medianBlur(source, kernel.size) – медианная фильтрация глубины для устранения шумов;

– cv2.absdiff(image, background) – разность изображений для вычитания фона;

– cv2.threshold(..., cv2.THRESH_BINARY) – бинаризация по заданному порогу (отделение объекта), где cv2.THRESH_BINARY указывает, что применяется именно бинарная пороговая фильтрация;

– cv2.morphologyEx(..., cv2.MORPH_OPEN/MORPH_CLOSE, kernel) – морфологическая обработка для устранения шумов и закрытия разрывов в маске с применением эрозии и дилатации, где kernel является морфологическим ядром;

– cv2.findContours(source, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE) – поиск контуров, где cv2.RETR_EXTERNAL ограничивает поиск внешними контурами, что позволяет выделить только основной объект, а cv2.CHAIN_APPROX_SIMPLE – упрощает описание контуров (сохраняя лишь ключевые точки);

– cv2.drawContours(...) – метод для отрисовки контуров.

Облако точек формируется из RGBD-данных средствами Open3D с помощью:

depth_o3d = o3d.geometry.Image(depth_masked)

color_o3d = o3d.geometry.Image(color_rgb)

rgbd = o3d.geometry.RGBDImage.create_from_color_and_depth(

color_o3d, depth_o3d,

depth_scale=1000.0,

depth_trunc=user_trunc,

convert_rgb_to_intensity=False

)

pcd = o3d.geometry.PointCloud.create_from_rgbd_image(rgbd, intrinsic)

Метод create_from_color_and_depth объединяет цвет и глубину в один формат RGBDImage. Параметр depth_scale указывает масштаб глубины в метрах, depth_trunc отсекает точки за указанным пределом, convert_rgb_to_intensity=False отключает преобразование RGB в значения яркости. Далее на основе RGBDImage и интринсиков камеры создаётся облако точек с помощью метода create_from_rgbd_image.

Интринсики камеры – это параметры внутренней калибровки, включающие фокусные расстояния и координаты центра проекции. Для Orbbec Astra в документации они явно не указаны. Пользователь может задать их вручную, однако по умолчанию используются приближённые значения, рассчитанные на основе разрешения и углов обзора камеры:

depth_intrinsics = {

"fx": 554.26, # фокусное по X

"fy": 579.41, # фокусное по Y

"cx": 320.0, # центр X

"cy": 240.0, # центр Y

}

rgb_intrinsics = {

"fx": 515.23, # фокусное по X

"fy": 534.06, # фокусное по Y

"cx": 320.0, # центр X

"cy": 240.0, # центр Y

}

После формирования облаков точек с отдельных ракурсов выполняется их последовательная обработка и объединение. Этот этап включает фильтрацию, коррекцию положения и выравнивание, сглаживание, и постобработку с возможностью сохранения и экспорта финальной модели. Все обозреваемые ниже процедуры реализованы с использованием библиотеки Open3D.

На предварительном этапе выполняется удаление шумов методом статистической фильтрации:

filtr_cl = cloud.remove_statistical_outlier(nb_neighbors, std_ratio)

remove_statistical_outlier позволяет исключить точки, находящиеся далеко от локального окружения. Параметр nb_neighbors определяет количество соседей, участвующих в статистике, а std_ratio, или «Outlier Sensitivity», как представлено в пользовательском интерфейсе, задаёт допустимое отклонение от среднего – чем ниже значение, тем строже фильтрация.

При необходимости облако может быть сдвинуто и повернуто в пространстве с использованием заданных трансформаций:

rotation_matrix = np.array([

[R11, R12, R13, tx],

[R21, R22, R23, ty],

[R31, R32, R33, tz],

[0, 0, 0, 1]

])

cloud.transform(rotation_matrix)

rotation_matrix – 4×4 матрица преобразования в однородных координатах, включающая в себя матрицу линейного преобразования и вектор трансляции.

Для даунсэмплинга, разрежения точек с помощью воксельной сетки необходимо использовать:

cloud = cloud.voxel_down_sample(voxel_size)

Параметр voxel_size задаёт размер ячеек в метрах. Каждая ячейка (воксель) покрывает объём voxel_size3. Все точки, попавшие в один воксель, заменяются одной «средней».

После даунсэмплинга при необходимости могут быть вычислены нормали, которые необходимы для определенных методов регистрации облаков и восстановления поверхности модели:

cloud.estimate_normals(o3d.geometry.KDTreeSearchParamHybrid(

radius=R, max_nn=N))

radius определяет радиус области поиска соседей, max_nn – максимальное число соседей.

Для грубого совмещения облаков может быть использована глобальная регистрация, для реализации которой предусмотрено 2 метода: RANSAC на FPFH-признаках с использованием метода registration_ransac_based_on_feature_matching и Fast Global Registration (FGR), для которого используется метод registration_fgr_based_on_feature_matching.

Для обоих методов даунсэмплирование с разными множителями voxel_size, а также вычисляются нормали с FPFH-признаками (Fast Point Feature Histograms), что описывают локальную геометрию вокруг точки и позволяют сопоставлять области между облаками при начальной (грубой) регистрации, с помощью метода:

fpfh = o3d.pipelines.registration.compute_fpfh_feature(

cloud, o3d.geometry.KDTreeSearchParamHybrid(

radius=R, max_nn=N) 27)

radius_feature представляет собой радиус поиска соседей для построения локального дескриптора, max_nn – максимальное число соседей, участвующих в расчёте.

Для метода RANSAC необходимо применить метод:

result = o3d.pipelines.registration.registration_ransac_based_on_feature_matching(

source_down, target_down,source_fpfh, target_fpfh,

mutual_filter=True, max_correspondence_distance=voxel_size * multiplier,

estimation_method=o3d.pipelines.registration.TransformationEstimationPointToPoint(False),

ransac_n=N,

checkers=[...],

criteria=o3d.pipelines.registration.RANSACConvergenceCriteria(

max_iterations, confidence)

)

Метод использует параметры:

– source_down, target_down – сравниваемые облака точек;

– source_fpfh, target_fpfh их FPFH-признаки;

– mutual_filter проверяет сохраняются ли только взаимные соответствия;

– max_correspondence_distance – максимальная допустимая дистанция между соответствующими точками;

– ransac_n – число выборок случайных точек («Random Samples» в пользовательском интерфейсе);

– checkers – фильтры для отбрасывания недопустимых соответствий, где edgeLength проверяет согласованность масштаба, distance – расстояния между сопоставленными точками, normal – согласованность ориентации нормалей;

– criteria – условия остановки алгоритма RANSAC, где max_iteration – максимальное число итераций, а confidence – вероятность нахождения корректного соответствия.

Метод FGR используется, когда допустима меньшая устойчивость, но важна высокая скорость, для чего применяется метод:

FastGlobalRegistrationOption(

maximum_correspondence_distance,

division_factor,

max_iterations)

division_factor представляет собой коэффициент, определяющий степень разбиения пространства признаков: большее значение ускоряет поиск соответствий, но снижает точность [18] . max_iterations – есть число итераций, maximum_correspondence_distance – максимальное допустимое расстояние между точками.

Для точного совмещения используется алгоритм ICP, поддерживающий следующие трансформации:

– TransformationEstimationPointToPoint() – минимизация расстояния между точками;

– TransformationEstimationPointToPlane()– учёт нормалей;

– TransformationEstimationForColoredICP() – дополнительный учёт цветовой информации.

Для вызова используется:

result = o3d.pipelines.registration.registration_icp(

source, target, threshold, current_transform, estimation, criteria)

Параметр threshold задаёт допустимое расстояние между соответствующими точками. fitness – доля корректных соответствий, rmse – среднеквадратичное отклонение (в метрах).

Для постобработки модели на финальном этапе могут быть применены:

– заполнение дыр;

– фильтрация артефактов;

– сглаживание;

– оптимизация, необходимая для устранения накопленных ошибок при попарном совмещении отдельных облаков.

Для заполнения дыр используется:

mesh = cloud.create_from_point_cloud_ball_pivoting(

radii=[postprocessing.hole_size * 0.5, postprocessing.hole_size, postprocessing.hole_size * 2])

Для фильтрации артефактов может быть применен кластерный анализ, который удаляет мелкие скопления точек:

clusters = cloud.cluster_dbscan(

eps=artifact_removal.tolerance,

min_points=artifact_removal.min_cluster)

Параметр min_cluster задаёт минимальный размер сохраняемых кластеров, tolerance определяет максимальное расстояние между точками в кластере.

Для сглаживания необходимо реализовать следующие методы:

cloud = cloud.smooth_mls(radius=smoothing.mls_radius, iterations=smoothing.iterations)

cloud = cloud.filter_smooth_laplacian(number_of_iterations=smoothing.iterations)

cloud = cloud.filter_smooth_taubin(

number_of_iterations=smoothing.iterations,

lambda=smoothing.taubin_lambda,

mu=smoothing.taubin_mu)

Необходимые параметры:

– mls_radius – область влияния Moving Least Squares;

– iterations – количество проходов алгоритма;

– taubin_lambda – коэффициент сглаживания;

– taubin_mu – компенсирующий коэффициент сохранения формы.

Для оптимизации используется:

option = GlobalOptimizationOption(

edge_prune_threshold=optimization.edge_prune_threshold,

use_geodesic_distance=optimization.use_geodesic,

max_correspondence_distance=VOXEL_SIZE * N)

Параметр edge_prune_threshold – это порог отбраковки неточных рёбер, use_geodesic учитывает геодезические расстояния.

В качестве алгоритмов оптимизации могут быть задействованы такие методы, как GlobalOptimizationLevenbergMarquardt() и GlobalOptimizationGaussNewton().

Для создания полигональной сетки и сохранения результатов могут быть использованы методы:

if method == "poisson":

mesh, densities = o3d.geometry.TriangleMesh.create_from_point_cloud_poisson(

pcd, depth=POISSON_DEPTH)

densities = np.asarray(densities)

density_threshold = np.percentile(densities, 10)

mesh = mesh.select_by_index(np.where(densities > density_threshold)[0])

elif method == "ball_pivoting":

radii = o3d.utility.DoubleVector(BALL_PIVOTING_RADII)

mesh = o3d.geometry.TriangleMesh.create_from_point_cloud_ball_pivoting(

pcd, radii)

elif method == "alpha_shape":

tetra_mesh, pt_map = o3d.geometry.TetraMesh.create_from_point_cloud(pcd)

mesh = o3d.geometry.TriangleMesh.create_from_point_cloud_alpha_shape(

pcd, ALPHA_VALUE, tetra_mesh, pt_map)

o3d.io.write_triangle_mesh("output/mesh_result.stl", mesh)

o3d.io.write_triangle_mesh("output/mesh_result.obj", mesh)

o3d.io.write_triangle_mesh("output/mesh_result.ply", mesh)

Полученные таким образом данные могут быть использованы для дальнейшей обработки или визуализации в сторонних приложениях, примером которого может послужить такая утилита, как MeshLa 13 b [34].

3.2.4 Блок связи с платформой реализует сетевое взаимодействие между управляющим прикладной программой на ПК и микроконтроллером. В качестве канала связи используется стандартный TCP-протокол. Связь реализуется с использованием модуля socket, входящего в стандартную библиотеку Python.

Модуль socket предоставляет интерфейс низкоуровневого управления сетевыми соединениями. На стороне клиента, представленного прикладным ПО, для создания объекта сокета необходимо использовать:

socket.socket(socket.AF_INET, socket.SOCK_STREAM)

AF_INET указывает на использование адресов IPv4, SOCK_STREAM – протокола TCP.

Для установки соединения с сервером, представленным ESP32, необходимо использовать метод:

connect(ESP_IP, ESP_PORT)

ESP_IP и ESP_PORT – IP-адрес и порт сервера соответственно.

Для отправки данных служит метод sendall(data), где data – буфер данных, представленный в байтовом формате.

Приём данных осуществляется с помощью метода recv(bufsize), где bufsize – максимальное количество байтов, которое планируется принять за один вызов. Полученные данные возвращаются в виде байт и требуют последующего декодирования.

Для установки таймаута операций приема может быть использован метод settimeout(seconds).

3.2.5 Для реализации сетевого взаимодействия на стороне микроконтроллера в блоке «Прием/передача данных» используется библиотека <WiFi.h>, предоставляющая интерфейс для подключения к Wi-Fi-сетям и организации TCP/UDP-соединений. Так как в рамках разрабатываемой системы микроконтроллер может работать и как точка доступа (Access Point), и как клиент существующей Wi-Fi-сети (Station), необходимо задействовать соответствующие параметры и функции.

Для создания точки доступа необходимо использовать:

WiFi.softAP(ssid, password);

ssid – имя создаваемой сети, password – пароль доступа к ней. Для получения IP-адрес микроконтроллера используется метод WiFi.softAPIP().

Для подключения к существующей сети, переключения в режим STA, необходим вызов:

WiFi.begin(ssid, password);

ssid и password – имя сети и ее пароль. После инициализации модуль пытается подключиться к указанной точке доступа. Успешность соединения проверяется через метод:

WiFi.status() == WL_CONNECTED

IP-адрес, полученный от роутера, доступен с помощью метода WiFi.localIP().

Для приёма соединений и обработки команд используется объект WiFiServer, при инициализации которого необходимо указать используемый порт:

WiFiServer server(port);

Для запуска сервера используется метод server.begin(). Вызов server.available() позволяет проверить наличие входящего подключения. При подключении клиента обмен данными осуществляется через объект WiFiClient, в котором чтение принимаемых данных осуществляется через методы client.read() и client.available(), а передача через client.println().

3.2.6 Для контроля положения платформы используется библиотека <AccelStepper.h> [35], предоставляющая интерфейс управления шаговыми двигателями с поддержкой задания ускорения и скорости. Управление осуществляется через объект класса AccelStepper, создаваемый с указанием используемого интерфейса и пинов управления:

AccelStepper stepper(AccelStepper::DRIVER, stepPin, dirPin);

В качестве первого параметра задаётся режим управления. Так как используемый шаговый двигатель управляется по интерфейсу STEP/DIR, выбирается параметр AccelStepper::DRIVER. В данном режиме подача импульса на stepPin инициирует один шаг, а уровень сигнала на dirPin определяет направление вращения.

После создания объекта необходимо настроить параметры движения с помощью методов:

stepper.setAcceleration(value);

stepper.setMaxSpeed(value);

stepper.setSpeed(value);

Методы выше устанавливают скорость, ее максимальное значение и ускорение соответственно. Скорость движения измеряется в шагах двигателя в секунду, а ускорение – в шагах на секунду в квадрате.

Для движения используются методы:

stepper.moveTo(position);

stepper.move(offset);

stepper.run();

stepper.runToPosition();

stepper.runSpeed();

moveTo() устанавливает целевую абсолютную позицию, move() – смещение относительно текущего положения. run() – метод, который должен регулярно вызываться для пошагового выполнения движения на заданную позицию. runToPosition() является блокирующей версией run(). runSpeed() используется для движения с постоянной скоростью, заданной методом setSpeed().

Для получения текущей позиции используется метод currentPosition(), изменения текущей позиции без движения – setCurrentPosition(value).

3.2.7 Блок обработки ввода обеспечивает считывание сигналов с элементов физического пользовательского интерфейса, а именно потенциометра и кнопок. Для получения значений с потенциометра используется функция analogRead(pin). При необходимости масштабирования значения может быть применена функция map(val, in_min, in_max, out_min, out_max), где val – исходное значение, in_min и in_max – границы входного диапазона, out_min и out_max – границы требуемого выходного диапазона.

Считывание состояния кнопки осуществляется с помощью digitalRead(pin). Для подавления дребезга используется программная фильтрация, в которой с помощью функции millis() фиксируется время при каждом изменении сигнала, и новое состояние устанавливается только тогда, когда его значение остается стабильным в течение заданной задержки debounceDelay:

if (reading != lastState)

lastChangeTime = millis();

if (millis() - lastChangeTime > debounceDelay)

if (reading != buttonState) {

buttonState = reading;

if (buttonState == LOW)

toggleFlag();

}

3.2.8 Блок вывода информации использует связку библиотек <Adafruit_GFX.h> и <Adafruit_SSD1306.h> [36], обеспечивающих поддержку графического вывода и управления дисплеями на контроллерах SSD1306. Библиотека <Adafruit_GFX.h> представляет собой базовый графический движок, реализующий функции отрисовки текста, примитивов и графики, а <Adafruit_SSD1306.h> – необходима для обеспечения взаимодействия с конкретным дисплейным контроллером через интерфейсы I²C или SPI.

Инициализация дисплея выполняется путем создания объекта класса Adafruit_SSD1306 с указанием параметров дисплея, а именно ширины, высоты, используемого интерфейса и номера пина сброса со значением -1, если тот отсутствует:

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, reset_pin);

После создания объекта дисплей инициализируется методом:

display.begin(SSD1306_SWITCHCAPVCC, i2c_address);

Параметр SSD1306_SWITCHCAPVCC указывает на тип используемого источника питания дисплея, а i2c_address – I²C-адрес устройства. Для уточнения адреса применяется специальный диагностический код «I2C Scanner», предоставляемый разработчиками платформы Arduino. Данная программа осуществляет перебор всех возможных адресов на шине и выводит адреса подключённых устройств в порт Serial.

После успешной инициализации может быть выполнена очистка экрана с помощью метода clearDisplay(). Для отображения информации применяется метод display().

Для вывода текстовой информации используются следующие функции:

– setTextSize(n) – установка масштаба шрифта, где n – целое число;

– setTextColor(color) – установка цвета текста;

– setCursor(x, y) – позиционирование курсора по координатам в пикселях;

– print()/println() – вывод строки или значения.

Кроме вывода текста, библиотека поддерживает отрисовку графических элементов с помощью методов:

– drawLine(x0, y0, x1, y1, color) – отрисовка линии;

– drawRect(x, y, w, h, color) и fillRect(...) – отрисовка прямоугольника с заливкой или без соответственно;

– drawCircle(x, y, r, color) – отрисовка окружности.

Для вывода изображения, закодированного в виде массива, может быть применена функция drawBitmap(x, y, bitmap, w, h, color). Массив bitmap содержит побитовое представление изображения, где каждый бит соответствует одному пикселю и отображается в указанном цвете color. Координаты x и y задают положение верхнего левого угла изображения, w и 17 h – его ширину и высоту в пикселях.

4 ПРИНЦИПИАЛЬНОЕ ПРОЕКТИРОВАНИЕ

4.1 Плата микроконтроллера

Конфигурация выводов на плате Esp32-WROOM-32 DevKit v1 представлена на рисунке 4.1.

Рисунок 4.1 – Конфигурация выводов на плате [20]

Выводами, необходимыми для питания платы микроконтроллера, являются выводы GND и VIN. Необходимым и достаточным напряжением питания является напряжение 5В. Параллельно этим выводам максимально близко к плате ставятся конденсаторы керамический 100 нФ для фильтрации высокочастотных помех и электролитический 47мкФ для сглаживания низкочастотных пульсаций и скачков напряжения.

На плате Esp32-WROOM-32 DevKit v1 поддерживаются следующие интерфейсы: два I2C, три SPI, три UART, два I2S.

Подключение по интерфейсу I2C по умолчанию осуществляется при помощи выводов SCL (Clock) и SDA (Data) – аналоговых выводов D22 и D21 соответственно.

Выводы (21) 1–5, 12–19, 21–23, 25–27, 32 и 33 являются цифровыми и могут быть настроены как входы/выходы. Выводы (4) 34–36 и 39 могут быть настроены только на вход.

ШИМ (широтно-импульсную модуляцию) поддерживают все контакты ввода-вывода. Позволяет выводить аналоговые значения в виде ШИМ-сигнала с разрядность 16 бит. Максимальное количество каналов 16 38 .

Также присутствуют АЦП (аналогово-цифровой преобразователь) на 15-ти выводах и ЦАП (цифро-аналоговой преобразователь) на двух выводах.

Описание задействованных в проектировании выводов платы будет приведено далее.

На принципиальной схеме при обозначении на шине следует задействовать собственное обозначение выводов платы Esp32-WROOM-32 DevKit v1. Соответствие между названиями контактов микроконтроллера их условным обозначением на разрабатываемой принципиальной схеме приведено в таблице 4.1.

Таблица 4.1 – Соответствие обозначений контактов на плате и схеме

Обозначение контакта на плате Esp32-WROOM-32 DevKit v1 Условное обозначение контакта на схеме

GND 001

GND 002

VIN 003

3V3 004

D2 102

D4 104

D5 105

D12 112

D13 113

D14 114

D15 115

D16 116

D17 117

D18 118

D19 119

D21 121

D22 122

D23 123

D25 125

D26 126

D27 127

D32 132

D33 133

D34 134

D35 135

TX0 201

RX0 202

VP 203

VN 204

EN 301

4.2 Шаговый электродвигатель

Шаговый электродвигатель Nema 17 17HS8401 имеет две версии исполнения: четыре и шесть выводов. В данном проекте используется версия со следующими контактами:

– A+, первая половина первой обмотки;

– A-, вторая половина первой обмотки;

– COM A, общий вывод первой обмотки;

– B+, первая половина второй обмотки;

– B-, вторая половина второй обмотки;

– COM B, общий вывод второй обмотки.

На рисунке 4.2 представлены различные типы шаговых двигателей: биполярный, униполярный, 6-выводной и 8-выводной. Особенностью 6-выводного двигателя, используемого в данном проекте, является возможность работы в обоих режимах – биполярном и униполярном.

Если центральные выводы обмоток (COM A и COM B) оставить неподключенными, двигатель функционирует в биполярном режиме, используя всю длину обмоток для максимального крутящего момента. При соединении этих выводов и подключении их к GND двигатель переходит в униполярный режим, в котором ток проходит только через половину обмотки, упрощая управление, но снижая мощность.

Рисунок 4.2 – Типы шаговых электродвигателей [37]

Как было указано ранее, контакты шагового мотора не управляются микроконтроллером напрямую. Для этого используется драйвер DRV8825, выводы которого отображены на рисунке 4.3. Список контактов представлен ниже:

– EN, активирует и отключает драйвер (включен на низком уровне);

– M0, M1 и M2, выбор делителя шага (от 1 до 1/32);

– RST, сброс драйвера (на низком уровне);

– SLP, переводит драйвер в спящий режим (на низком уровне);

– STEP, подача импульсов для выполнения шага;

– DIR задает направление вращения;

– VMOT и GND MOT, питание шагового двигателя от 8.2 до 45 В;

– A1 и A2, первая обмотка двигателя (A+ и A-);

– B1 и B2, вторая обмотка двигателя (B+ и B-);

– FAULT, сигнал ошибки (низкий уровень при ошибке);

– GND LOGIC, заземление микроконтроллера.

Рисунок 4.3 – Контакты DRV8825 [38]

Драйвер DRV8825 предназначен для управления биполярными шаговыми двигателями, поэтому центральные отводы мотора Nema 17 17HS8401 (COM A и COM B) остаются неиспользованными. Для подключения выбираются только A+, A-, B+, B-, что позволяет полностью задействовать обмотки и обеспечивать максимальный крутящий момент.

Для стабильной работы драйвера на вход питания двигателя необходимо установить конденсатор емкостью 100 мкФ (VMOT), чтобы сгладить пиковые токи и защитить модуль от возможных скачков напряжения. Для предотвращения перегрева на чип драйвера должен быть установлен радиатор.

Конечная схема подключения драйвера и двигателя представлена на рисунке 4.4.

Рисунок 4.4 – Соединение двигателя Nema 17 и платы DRV8825 [39]

Выводы плат DRV8825 и Esp32-WROOM-32 DevKit v1 соединяются следующим образом:

– STEP модуля DRV8825 подключается к выводу D12 на микроконтроллере для выполнения шагов (важным аспектом является поддержка ШИМ);

– DIR модуля DRV8825 подключается к выводу D14 на микроконтроллере для задания направления вращения вала двигателя.

4.3 Дисплей

В качестве устройства вывода текстовой информации используется модуль DSM-OLEDv2-0.96-4P-YB, расположение контактов которого представлено на рисунке 4.5.

Рисунок 4.5 – Расположение контактов на дисплее OLED 0.96 [40]

Модуль OLED 0.96 является дисплеем на органических светодиодах, в основе которого лежит контроллер SSD1306.

SSD1306 – это специализированная микросхема, управляющая 128×64 (или 128×32) пикселями дисплея. Контроллер принимает команды от микроконтроллера и преобразует их в изображение на экране. Среди выполняемых функций можно выделить:

– хранение изображения во встроенной памяти (1 КБ буфера кадров);

– управление яркостью и контрастностью;

– аппаратные функции, такие как прокрутка, инверсия цветов, отражение по горизонтали/вертикали;

– поддержка интерфейсов I2C и SPI.

Модуль DSM-OLEDv2-0.96-4P-YB представляет собой схему, обеспечивающую работу контроллера SSD1306 посредством интерфейсных соединений, стабилизации питания и удобства интеграции с микроконтроллерами.

DSM-OLEDv2-0.96-4P-YB (I2C версия) имеет выводы:

– VCC;

– GND;

– SCL, тактовая линия;

– SDA, линия данных.

Выводы дисплея и платы микроконтроллера соединяются следующим образом:

– SDA дисплея и D21 Esp32-Wroom;

– SCL дисплея и D22 Esp32-Wroom.

4.4 Подключение светодиодов

В проектируемой установке используется 2 светодиода.

Для ограничения тока, проходящего через светодиод, используется резистор, номинал которого высчитывается по формуле:

RД =UП-UДIПР,

(4.1)

где UП – напряжение питания;

UД – напряжение, падающее на светодиоде;

IПР – прямой ток светодиода.

В установке используются светодиоды со следующими параметрами:

– UД = 2.1 В;

– IПР = 20 мА.

За подачу питания на светодиоды отвечает микроконтроллер, обеспечивающий напряжение питания 3.3В.

Расчет сопротивления для ограничивающего через светодиоды ток:

RД =3.3-2.10,02=60 Ом.

(4.1)

В связи с отсутствием резисторов с рассчитанным номиналом и наличием резисторов номиналом 62 Ом, используем вторые по причине не критичной разницы.

Анод светодиодов подключается напрямую к выходу платы микроконтроллера, катод – к резистору, ведущему к земле, что представлено на рисунке 4.6.

Рисунок 4.6 – Подключение светодиода [41]

Зеленый и синий светодиоды подключаются к выводам платы Esp32-WROOM-32 DevKit v1 D25 и D26 соответственно.

4.5 Подключение кнопок

Возможно два основных способа подключения кнопок к микроконтроллеру: с подтяжкой линии порта к высокому или низкому логическому уровню через резистор – как показано на рисунке 4.7.

Первый способ – это подтяжка к высокому уровню (Pull-up), где один вывод кнопки подключается к земле, а другой – к входу микроконтроллера, который через резистор подтянут к питанию. В этом случае, когда кнопка не нажата, на входе находится высокий уровень (логическая «1»), а при нажатии на нее – низкий уровень (логический «0»).

Подтяжка к низкому уровню (Pull-down) осуществляется путем подключения одного вывода кнопки к питанию, а другого – к входу микроконтроллера, который через резистор подтянут к земле. При таком способе при не нажатой кнопке на входе будет низкий уровень (логический «0»), в обратном случае – высокий (логическая «1»).

Резистор в данной схеме выполняет роль подтягивающего элемента, который предотвращает неопределённое состояние входа, когда кнопка разомкнута. Без него вход микроконтроллера может оказаться в плавающем состоянии, улавливая случайные помехи, что приведёт к некорректной работе устройства.

Рисунок 4.7 – Варианты подключения кнопок к микроконтроллеру [42]

В данной схеме реализован вариант подключения кнопок с подтяжкой к низкому уровню (Pull-down).

Подключение кнопок выполнено следующим образом: один из контактов каждой кнопки соединён с линией питания платы микроконтроллера, равной 3.3В (3V3), а второй – одновременно с цифровым входом микроконтроллера и через резистор с большим сопротивлением – с землёй (GND). В качестве подтягивающих элементов используются резисторы номиналом 10 кОм.

На плате Esp32-WROOM-32 DevKit v1 кнопки подключены к цифровым входам D18 и D19, что позволяет микроконтроллеру регистрировать нажатия, отслеживая переходы сигнала с низкого на высокий уровень. При не нажатой кнопке на соответствующем входе фиксируется логический ноль, а при нажатии – логическая единица.

4.6 Подключение резистора переменного осевого

Потенциометр (он же переменный резистор) представляет собой резистивный делитель напряжения с подвижной средней точкой. При подключении крайних выводов (первый и третий) на источник напряжения (GND и 3.3V соответственно) на среднем выводе (втором) появится напряжение (относительно GND), пропорциональное положению ручки потенциометра. Схематичное представление потенциометра представлено на рисунке 4.8.

В устройстве используется резистор переменный осевой с максимальным сопротивлением, равным 5кОм.

Выводы потенциометра и платы микроконтроллера соединяются следующим образом:

– первый вывод потенциометра и GND Esp32-Wroom;

– второй вывод потенциометра и D33 Esp32-Wroom;

– третий вывод потенциометра и 3V3 Esp32-Wroom.

Рисунок 4.8 – Схематичное представление потенциометра [43]

4.7 Питание

Плата BMS 3S необходима для управления зарядом и разрядом аккумуляторных батарей, состоящих из трёх последовательно соединённых элементов (3S). Она защищает аккумуляторы от перезаряда, глубокого разряда, короткого замыкания и перегрузки по току, обеспечивая безопасную эксплуатацию устройства.

Плата рассчитана на работу с литий-ионными (Li-Ion) или литий-полимерными (Li-Po) аккумуляторами и поддерживает номинальное напряжение 11,1 В

(максимум 12,6 В). Встроенная схема балансировки следит за тем, чтобы все элементы заряжались равномерно, продлевая срок службы батарей.

Плата имеет следующие контакты:

– B-, отрицательный вход аккумуляторной батареи;

– B1+ и B2+, промежуточные входы для подключения ячеек;

– B3+, положительный вход батарей;

– P+, положительный выход;

– P-, отрицательный выход (GND).

Схема подключения трех последовательно соединенных аккумуляторов к плате BMS 3S представлено на рисунке 4.9.

На положительном выходе P+ расположен переключатель, который замыкает и размыкает цепь питания, тем самым отвечая за включение и выключение устройства.

Рисунок 4.9 – Схема подключения трех аккумуляторов к плате BMS 3S [43]

Mini360 – это компактный и эффективный модуль понижающего преобразователя (buck-конвертера) на основе микросхемы MP2307.

Микросхема MP2307 – это высокоэффективный синхронный понижающий DC-DC преобразователь с частотой переключения 340 кГц. Она имеет встроенные MOSFET-транзисторы, что позволяет добиться КПД до 95%, снижая тепловые потери и улучшая стабильность работы модуля.

Модуль принимает входное напряжение в диапазоне от 4,75 В до 23 В DC и способен выдавать регулируемое выходное напряжение от 1 В до 17 В DC. Максимальный кратковременный ток нагрузки составляет 3 А, однако для стабильной работы рекомендуется не превышать 1,8 А.

Регулировка выходного напряжения осуществляется с помощью встроенного подстроечного резистора. Поворот по часовой стрелке увеличивает напряжение, а против часовой – уменьшает.

Плата имеет следующие контакты:

– IN +, положительный вход;

– IN -, отрицательный вход (GND);

– OUT +, положительный выход;

– OUT -, отрицательный выход (GND).

На входы MINI-360 подключаются выходы платы BMS 3S P+ и P-, а с выводов осуществляется питание устройства.

Элементы и выводы платы MINI-360 представлены на рисунке 4.10.

Рисунок 4.10 – Элементы и выводы платы MINI-360 [44]

4.8 Принципиальная схема

Принципиальная электрическая схема установки с вращающейся платформой разрабатываемого комплекса приведена на чертеже ГУИР.400201.099 Э3.

4.9 Перечень элементов

Перечень необходимых электронных компонентов установки с вращающейся платформой разрабатываемого комплекса приведен в приложении А.

5 МОДЕЛИРОВАНИЕ

5.1 Проектирование печатных плат

5.1.1 Для соединения всех ранее упомянутых электронных компонентов решено использовать печатные платы собственного изготовления.

Разработка печатных плат выполнялась в среде EasyEDA в соответствии с приведённой ранее принципиальной схемой. EasyEDA – это онлайн сервис EDA, для создания и редактирования печатных плат и электрических схем. Она может использоваться на любом железе и работать под любой операционной системой – Linux, Windows или Mac OS. Для её использования требуется любой совместимый браузер: Chrome, Firefox, Microsoft Edge, Opera или Safari. Также есть возможность установить десктопную версию данной программы.

Разработка печатной платы включает в себя несколько последовательных этапов:

1 Определение всех входных и выходных сигналов, а также всех задействованных электронных компонентов.

2 Создание в среде EasyEDA электрической схемы, на которой все компоненты соединяются логическими связями в соответствии с приведенной ранее принципиальной схемой.

3 После создания схемы следует этап размещения элементов на плате, где важно обеспечить компактность, а также учесть требования к разводке, чтобы исключить пересечения дорожек и взаимные помехи между цепями.

4 Когда все элементы размещены, проводится трассировка дорожек, соединяющих контакты компонентов.

5 Перед переходом к изготовлению выполняется проверка платы на наличие ошибок (DRC-проверка) – коротких замыканий, незамкнутых соединений и других потенциальных проблем. Также может применяться ручная визуальная проверка.

6 После окончательной проверки проводится подготовка к изготовлению, а именно формируется топология платы (Gerber-файлы) или, в случае самостоятельного изготовления, экспортируется рисунок дорожек для переноса на текстолит.

7 Далее следует изготовление платы, которое может включать в себя как производственное изготовление, так и собственноручное, включающее технологии «лазерно-утюжную» и «фоторезист».

8 На завершающем этапе производится монтаж всех компонентов на плату и их припайка согласно схеме.

9 После завершения сборки проводится проверка работоспособности схемы и устранение возможных дефектов монтажа.

Для снижения расхода травящего раствора и ускорения процесса травления свободные участки платы заливаются общим полигоном земли. Это позволяет минимизировать открытую медь, которую необходимо удалить при изготовлении.

Толщина дорожек зависит от назначения. Для подбора толщины используются значения, полученные путем соблюдения стандарта IPC-2221 и сохранения запаса, в том числе для удобства собственноручного травления. Для питания платы микроконтроллера, модуля дисплея, логики, а также подключения линий их управления, светодиодов, кнопок, потенциометра, используется ширина дорожек, равная 1 мм. Для дорожек питания шагового электродвигателя берется ширина 3 мм.

5.1.2 Изготовление плат происходит собственноручно по «лазерно-утюжной» технологии. Для получения плат по данной технологии необходимо выполнить следующие этапы:

1 Печать рисунка дорожек в зеркальном виде на специальную термобумагу (может быть заменена глянцевой или фотобумагой) с помощью лазерного принтера. Необходимо установить высокое качество печати и максимальную плотность тонера.

2 Подготовка подложки. Стеклотекстолит вырезается с небольшим запасом по краям, тщательно зачищается мелкой наждачной бумагой и обезжиривается ацетоном.

3 Перенос изображения. Для этого бумага с рисунком укладывается на медную сторону платы и прогревается утюгом в течение 1–3 минут, равномерным прижимом по всей площади. После охлаждения бумага размачивается и аккуратно удаляется, оставляя тонер на меди.

4 Далее следует процесс травления. Плата погружается в раствор, приготовленный из 100 мл трёхпроцентной перекиси водорода, 30 г лимонной кислоты и 7 г соли. Травление продолжается до полного удаления открытого медного слоя, при этом раствор периодически перемешивается.

5 После травления тонер смывается растворителем, сверлятся отверстия под выводы компонентов, залуживаются дорожки и производится пайка всех элементов согласно схеме.

5.1.3 При учитывании предполагаемой конструкции корпуса появляется необходимость в создании следующих плат:

– плата для установки BMS 3S;

– плата с разведением питания и микроконтроллером;

– плата «панели управления», содержащая модуль дисплея, резистор переменный осевой, кнопки и светодиоды.

Также отдельно можно выделить батарейный отсек, представленный на рисунке 5.1, с тремя последовательно соединенными аккумуляторами Li-Ion 18650 и дополнительными выводами для платы BMS 3S.

Схема платы для установки BMS 3S представлена на рисунке 5.2 и включает в себя:

– разъем (вход) для подключения питания от аккумуляторов (GND, 3.7V, 7.4V, 11,1V);

– место под разъемы для размещения bms-платы;

– разъем (выход) для вывода питания (GND, 12V).

Рисунок 5.1 – Батарейный отсек с дополнительными выводами

Рисунок 5.2 – Схема платы для установки BMS 3S

Изображение заготовки платы для установки BMS 3S представлено на рисунке 5.3.

Изображение вытравленной платы для установки BMS 3S с размещенными на ней компонентами представлено на рисунке 5.4.

Рисунок 5.3 – Заготовка платы для установки BMS 3S

Рисунок 5.4 – Плата для установки BMS 3S

Схема платы с разведением питания и микроконтроллером представлена на рисунке 5.5 и включает в себя:

– разъем (вход) для подключения питания (GND, 12V);

– место под разъемы для подключения переключателя, замыкающего цепь питания;

– место под разъемы для размещения платы DC-DC MINI-360;

– место под разъемы для размещения платы микроконтроллера;

– место под разъемы для размещения платы драйвера шагового электродвигателя;

– разъе 22 м (выход) для подключения платы «панели управления» (GND, 3V3, 5V, а также семь цифровых выводов микроконтроллера);

– разъем (выход) для подключения шагового электродвигателя (A+, A-, B+, B-).

Рисунок 5.5 – Схема платы с разведением питания и микроконтроллером

Изображение заготовки платы с разведением питания и микроконтроллером представлено на рисунке 5.6.

Изображение вытравленной платы с разведением питания и микроконтроллером с размещенными на ней компонентами представлено на рисунке 5.7.

Рисунок 5.6 – Заготовка платы для установки BMS 3S

Рисунок 5.7 – Плата для установки BMS 3S

Схема платы «панели управления» представлена на рисунке 5.8 и включает в себя:

– разъем (вход) для подключения питания и управляющих сигналов микроконтроллера (GND, 3V3, 5V и семь цифровых выводов микроконтроллера);

– место под разъемы для подключения модуля дисплея;

– место под разъемы для подключения кнопок и связанных с ними резисторов;

– место под разъемы для подключения резистора переменного осевого;

– место под разъемы для подключения светодиодов и ограничивающих проходящий через них ток резисторов.

Рисунок 5.8 – Схема платы «панели управления»

Изображение заготовки платы «панели управления» представлено на рисунке 5.9.

Изображение вытравленной платы «панели управления» с размещенными на ней компонентами представлено на рисунке 5.10.

Рисунок 5.9 – Заготовка платы «панели управления»

Рисунок 5.10 – Плата «панели управления» с монтажом компонентов

Схемы плат приведены в приложении Б на чертеже ГУИР.400201.099 МЭ.

5.2 Разработка корпуса

5.2.1 Для размещения компонентов устройства и формирования устойчивой механической конструкции решено использовать сборный корпус, напечатанный на 3D-принтере.

Конструкция корпуса должна состоять из отдельных функциональных модулей, соответствующих назначению компонентов, описанных в предыдущих разделах. Такой подход обеспечит удобство при сборке, отладке, а также возможной модернизации устройства.

Проектирование и редактирование деталей производится в среде FreeCAD – бесплатной системе автоматизированного проектирования с открытым исходным кодом. Программа поддерживает параметрическое моделирование и позволяет точно настраивать размеры, форму и конструктивные особенности деталей в соответствии с заданными требованиями.

Изготовление корпуса осуществляется с помощью 3D-принтера кафедры ЭВМ. Печать конструкции выполняется поэтапно: каждый элемент изготавливается отдельно, а затем соединяется с остальными, формируя цельное изделие.

Для проекта планируется изготовление двух отдельных конструкций: установки с вращающейся платформой и регулируемого штатива для RGB камеры.

В качестве основы используются открытые 3D-модели, размещённые на платформе Thingiverse: [45] и [46]. Некоторые детали оригинальных моделей были отредактированы в соответствии с особенностями проекта: внесены изменения в элементы крепления, а также скорректированы габариты отдельных частей.

5.2.2 Основой установки с вращающейся платформой является алюминиевый конструкционный профиль. Все необходимые детали представлены на рисунке 5.11 и перечислены ниже:

– 5 деталей типа V20×40 длиной 150 мм;

– 2 детали типа V20×20 длиной 265 мм;

– 1 деталь типа V20×20 длиной 150 мм;

Для размещения «панели управления» спроектировано три детали, две из которых предназначены для крепления к профилю типа V20×40. Лицевая деталь содержит специально выделенные пазы под размещение компонентов соответствующей платы. Модели данных деталей представлены на рисунке 5.12.

Для соединения профилей типа V20×40, V20×20 и обоих типов используются «угловая», «Т-образная» и «совмещенная» детали соответственно, представленные на рисунке 5.13, а также винты M5.

Для создания стабильной и обеспечивающей скольжение опоры непосредственно самой платформы на каждую «угловую» и «совмещенную» детали также крепятся винт M8, гайка M8, шайба M8 и подшипник 698ZZ с внешним и внутренним диаметром 19 и 8 мм соответственно.

Для фиксации проводов на профилях используются детали «крючок» и «зацеп», представленные на рисунке 5.14, и винты M5.

Рисунок 5.11 – Подготовленные профили

а б в

Рисунок 5.12 – Модели деталей для размещения «панели управления»: а – боковая левая; б – лицевая; в – боковая правая

а б в

Рисунок 5.13 – Детали для соединения профилей: а – «угловая»;

б – «Т-образная»; в – «совмещенная»

а б

Рисунок 5.14 – Детали для фиксации проводов на профилях: а – «крючок»; б – «зацеп»

Деталь держателя отсека под аккумуляторы и платы с BMS 3S представлена на рисунке 5.15. Для крепления платы используются винты M3, для непосредственно детали – винты M5.

Деталь держателя платы с микроконтроллером представлена на рисунке 5.16. Для крепления платы используются винты M3 и пластиковые стойки M3, для непосредственно детали – винты M5.

Рисунок 5.15 – Деталь держателя отсека под аккумуляторы и платы с BMS 3S

Рисунок 5.16 – Деталь держателя платы с микроконтроллером

Деталь держателя шагового электродвигателя представлена на рисунке 5.17. Для крепления мотора используются винты M3, для непосредственно детали – винты M5. В саму деталь устанавливается подшипник 6800 2RS с внешним и внутренним диаметром 19 и 10 мм соответственно.

Для обеспечения контроля поворота платформы используется пара конических шестерен [47] с количеством зубьев, равным 15, у малой, и 45 – у большой. Их внутренние диаметры равняются 5 и 10 мм соответственно. Малая шестерня насаживается на вал шагового электродвигателя. Большая – на алюминиевый вал, который в свою очередь вставляется во внутреннее кольцо подшипника. На вале для крепления площадки вырезается шпонка. Конструкция с шестернями представлена на рисунке 5.17.

Рисунок 5.17 – Конструкция с шестернями

Площадка для размещения предмета представляет собой круглый лист плиты ПВХ диаметром 35 см и толщиной 3 мм с прикрепленной деталью-пазом для закрепления на основной конструкции. Площадка и собранная конструкция представлены на рисунке 5.18.

а б

Рисунок 5.18 – Установка с вращающейся платформой: а – основание; б – платформа

5.2.3 В конструкции штатива можно выделить следующие ключевые элементы:

– «основание»;

– «шарнир основания»;

– 2 «плеча»;

– «шарнир камеры».

«Основание» представляет собой «треногу» с углублением под установку крепления для «шарнира основания. Деталь представлена на рисунке 5.19.

Рисунок 5.19 – Деталь «Основания»

«Шарнир основания», «плечо» и «шарнир камеры» представлены на рисунке 5.20.

Для соединения «шарниров» и «плеч» используются винты, шайбы и гайки типа M4.

Крепеж камеры вставляется в специальный паз соответствующего «шарнира» и закрепляется с помощью винтов типа M2,5.

Собранная конструкция штатива представлена на рисунке 5.21.

а б в

Рисунок 5.20 – Детали штатива: а – «шарнир основания»; б – «плечо»; в – «шарнир камеры»

Рисунок 5.21 – Штатив с камерой в собранном виде

6 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

6.1 Программное обеспечение для микроконтроллера

6.1.1 Схема программы микроконтроллера представлена на чертеже ГУИР.400201.099 ПД.1.

Исходный код программы микроконтроллера представлен в приложении В.

6.1.2 Код микроконтроллера начинается с объявления директив препроцессора, определяющих логическую привязку функциональных элементов системы к конкретным выводам микроконтроллера ESP32, а также ряд постоянных значений, используемых в логике работы устройства.

С помощью директив #define заданы:

– аппаратные привязки выводов микроконтроллера, подключенных к кнопкам, светодиодам, потенциометру и драйверу;

– параметры OLED-дисплея (ширина и высота экрана);

– параметры сетевого взаимодействия, а именно порт TCP-сервера;

– константы, используемые при управлении шаговым двигателем и считывании потенциометра, а именно шаги, скорость и максимальное значение АЦП.

Для структурной организации кода переменные сгруппированы по функциональному назначению.

Для каждой из кнопок MENU_BUTTON и ACTION_BUTTON определены переменные, обеспечивающие программную фильтрацию дребезга контактов:

– bool action_button_last_state, bool action_button_state, bool menu_button_last_state, bool menu_button _last_state;

– unsigned long action_button_last_debounce_time и menu_button_last_debounce_time.

Для хранения текущего состояния светодиода MOTOR_LED, отображающего активность мотора переменная используется переменная bool motor_led_state. Второй светодиод POWER_LED всегда активен после инициализации компонентов устройства и в отдельной переменной не нуждается.

Для управления OLED-дисплеем используется Adafruit_SSD1306 display.

Управление шаговым двигателем осуществляется с помощью AccelStepper stepper.

Для сетевого взаимодействия ключевыми являются переменные:

– WiFiServer server и WiFiClient activeClient – объекты для реализации TCP-соединений по Wi-Fi;

– ap_ssid, ap_password, sta_ssid, sta_password – строковые константы для настройки Wi-Fi в режимах точки доступа и станции;

– bool currentModeIsAP – текущее сетевое состояние устройства;

– bool clientConnected – наличие подключённого TCP-клиента.

Для работы с меню и пользовательским интерфейсом используются:

– DeviceMode current_device_mode – текущий режим устройства;

– const char* menuItems[] – строки для отображения названий пунктов меню;

– uint8_t potenciometer_selected_item – выбранный на основе положения потенциометра пункт меню;

DeviceMode – это перечисление возможных режимов работы устройства:

```
enum DeviceMode {
MOTOR_TEST, // Режим проверки шагового электродвигателя
WIFI_CONFIG, // Режим работы с подключением
SCAN_MODE, // Режим сканирования
DEVICE_INFO, // Режим вывода общей информации
MENU_ITEMS_NUM, // Общее количество пунктов меню
MAIN_MENU_OUTPUT // Режим вывода меню
};
```

В режиме тестирования мотора используются флаг bool test_mode_motor_is_running и параметр int motor_test_speed.

В режиме сканирования из ключевых переменных, обеспечивающих контроль параметров и состояния процесса сканирования, можно выделить:

– bool scanning_outp_is_position – флаг вида отображаемой информации (число позиций или угол).

– параметры движения мотора int motor_scan_speed, int motor_scan_acceleration, int scan_turn_delta;

– переменные-флаги статуса процесса сканирования bool scan_in_progress, bool scan_abort_request;

– переменные-счетчики int scan_number_of_turns_to_do и int scan_number_of_turns_completed.

6.1.3 Для организации многозадачности в программе микроконтроллера создаются две независимые задачи: buttonTask и serverTask. Их запуск производится с помощью функции FreeRTOS:

```
xTaskCreate()
functionPointerTask, // Функция задачи
"Task Name", // Имя задачи
STACK_SIZE, // Размер стека
parameters, // Параметры
priority, // Приоритет
taskDeskriptor // Дескриптор задачи
);
```

Задача buttonTask используется для обработки нажатия кнопок. Порядок действий представлен в схеме программы. Переключение состояния запуска и остановки шагового двигателя реализуется через изменение значения переменной test_mode_motor_is_running на противоположное. Для смены режимов AP и STA используется функция toggleWiFiMode(), для смены типа выводимой информации –change_scan_outp().

Задача serverTask обрабатывает подключение клиентов и их запросы с помощью функции handleClientConnections().

Для предотвращения конфликтов доступа к общим данным применяется механизм межзадачной синхронизации – двоичный семафор FreeRTOS xMutex, инициализируемый следующим образом:

xMutex = xSemaphoreCreateMutex();

Режимы выводов задаются следующим образом:

pinMode(MENU_BUTTON_PIN, INPUT);

pinMode(ACTION_BUTTON_PIN, INPUT);

pinMode(POTENCIOMETER_PIN, INPUT);

pinMode(POWER_LED_PIN, OUTPUT);

pinMode(MOTOR_LED_PIN, OUTPUT);

6.1.4 Как указано в перечислении DeviceMode, устройство может находиться в четырех режимах:

– режим вывода меню;

– режим проверки шагового электродвигателя;

– режим работы с подключением;

– режим сканирования;

– режим вывода общей информации.

В режиме вывода меню на дисплее отображаются все доступные для выбора пункты с выделением текущего выбранного на основе значения потенциометра с помощью функции output_main_menu().

В режиме проверки шагового двигателя выполняется функция motor_test(), в которой происходит регулирование скорости шагового двигателя на основании значения потенциометра. На дисплей в данном режиме выводится текущая скорость в градусах в секунду (°/c).

В режиме работы с подключением на дисплей выводятся:

– режим подключения AP или STA;

– IP адрес установки или «X.X.X.X», если не удалось подключиться к сети пользователя;

– «Client is Connected» или «Client is not Connected».

Работа с подключением осуществляется в функции connection_management(). Установка режимов AP и STA выполняется в функциях initAPMode() и initSTAMode() соответственно.

Для проверки условия «Устройство в сети» используется код:

WiFi.status() == WL_CONNECTED || currentModeIsAP;

Для проверки состояния клиента используются:

– activeClient.connected() – проверка подключения клиента;

– server.available() – проверка запроса на подключение;

– activeClient.available() – проверка на наличие данных от клиента.

Микроконтроллер принимает сообщения от клиента в виде массива байт фиксированной длины uint8_t [65]. Первый байт каждого сообщения служит идентификатором команды, определяющим тип запроса, в зависимости от которого интерпретируются остальные данные. Структура входящих пакетов представлена в таблице 6.1.

Таблица 6.1 – Структура принимаемых микроконтроллером пакетов

Код (байт 0) Запрос Байты

[1-16] Байты

[17-32] Байты

[17-48] Байты

[49-64]

01 Начать процесс сканирования Число поворотов

(uint_16) Угол поворота

(uint_16) Скорость

°/c

(uint_16) Ускорение

°/c2

(uint_16)

02 Продолжить процесс сканирования Не используются

03 Прервать процесс сканирования Не используются

04 Установить параметры сети пользователя SSID сети пользователя

char [32] Пароль сети пользователя

char [32]

После каждого запроса микроконтроллер отправляет сообщения клиенту в виде массива байт фиксированной длины uint8_t [3]. Первый байт каждого сообщения определяет тип ответа. Структура исходящих пакетов представлена в таблице 6.2.

Таблица 6.2 – Структура отправляемых микроконтроллером пакетов

Код (байт 0) Ответ Байты [1-16]

01 Сделан N-ный поворот Номер выполненного поворота

(uint_16)

02 Процесс сканирования прерван Не используются

03 Данные сети пользователя сохранены Не используются

Процесс сканирования и, соответственно, поворота платформы осуществляется в функции scanning_process(). В режиме сканирования на дисплей выводятся:

– состояние процесса сканирования: «Scanning: In Progress/ Not In Progress»;

– прогресс процесса поворота платформы в одном из двух форматов: «Position {K}/{N}», где K – число выполненных поворотов и N – число поворотов, которые необходимо сделать, либо «Rotation Angle: {X}/{Y}», где X – текущий угол поворота платформы и Y – конечный угол поворота.

Для удобства пользователя параметры вращения платформы, скорость и ускорение, задаются в градусах в секунду (°/c). Однако при управлении шаговым двигателем через драйвер в коде используются не углы, а количество необходимых шагов, для чего выполняется преобразование градусов в шаги электродвигателя.

Один полный оборот двигателя соответствует 200 шагам. С учётом передаточного отношения зубчатой передачи 15/45 (в 3 раза), один шаг на выходе соответствует:

1\ шаг =360°200·1545 =0,6°.

(6.1)

Таким образом, для преобразования градусов в шаги используется формула:

Nшагов =θ0,6.

(6.2)

Для вывода информации об устройстве используется функция info_outp(). Данная информация включает в себя названия ключевых комплектующих устройства: микроконтроллера, дисплея, шагового электродвигателя и его драйвера.

6.2 Программное обеспечение для персонального компьютера

6.2.1 Схема программы прикладного программного обеспечения представлена на чертеже ГУИР.400201.099 ПД.2.

Исходный код прикладного программного обеспечения представлен в приложении Г.

Пользовательский интерфейс разрабатывался в графическом редакторе Qt Creator в формате «.ui». Для интеграции с основным кодом интерфейс был преобразован в Python-скрипт с помощью утилиты pyuic6:

pyuic6 -o mainwindow.py mainwindow.ui

В схеме программы представлен общий алгоритм взаимодействия компонентов. Далее подробнее рассмотрим функционал пользовательского интерфейса по вкладкам.

Для обработки взаимодействия пользователя с элементами интерфейса используются методы, реализованные в классе главного окна.

Во всех именах элементов интерфейса присутствует префикс, соответствующий названию вкладки, который для краткости опускается в дальнейшем описании. Таблица префиксов представлена в таблице 6.3.

Таблица 6.3 – Префиксы имен элементов интерфейса

Вкладка Префикс

Platform platformConnectionSettings_

Cam Check cam_check_

Scanning scanning_process_

Frames Processing framesProcessing_

Clouds Processing cloudsProcessing_

6.2.2 На вкладке «Platform» при нажатии на кнопку «Send New AP Data» вызывается метод sendNewAPData(), используемый для отправки установке с платформой данных о пользовательской сети и ожидания от нее ответа. Метод использует значения, введённые в поля:

– ApSSID_lineEdit;

– APPassword_lineEdit.

При нажатии на кнопку «Connect»/«Disconnect» вызывается метод connect_disconnect_button_process(), реализующий установку/разрыв соединения с микроконтроллером. Метод использует значения, введённые в поля:

– platformPort_lineEdit;

– platformIP_lineEdit.

6.2.3 На вкладке «Cam Check» за включение/выключение вывода камеры отвечает cameraOutput_checkBox, а за ее настройки – флажки:

– depthToColor_checkBox;

– syncIsEnabled_checkBox;

– mirroringIsEnabled_checkBox.

resolutionX_spinBox и resolutionY_spinBox используются для задания разрешения сохраняемых снимков. Максимальное разрешение обеих камер: 640×480.

Для предоставления пользователю выбора способа отображения изображения глубины используется выпадающий список depthDisplayMethod_comboBox. Из представленных вариантов:

– cv2.COLORMAP_JET – стандартная палитра с переходом «От синего к красному»;

– cv2.COLORMAP_BONE – черно-белая палитра;

– cv2.COLORMAP_PLASMA – палитра с переходом «От фиолетового к желтому».

Нажатие на кнопку «Choose Folder» вызывает choose_folder(), который использует метод getExistingDirectory() класса QFileDialog для открытия диалогового окна с выбором каталога для сохранений фото:

dir_path = QFileDialog.getExistingDirectory(self, "Выберите каталог", "")

Методы обработки нажатий на кнопки с аналогичным действием также используют данный метод.

При нажатии на кнопку «Make Photo» вызывается метод choose_folder(), сохраняющий depth и RGB снимки с заданными настройками в выбранный каталог под именем, указанным в photoName_lineEdit, с расширениями «.png» и «.npy» соответственно.

6.2.4 На вкладке «Scanning» кнопка «Choose Folder» используется для выбора каталога, куда сохранять снимки, поле «Object Name» – для ввода имени объекта, которое будет использоваться в шаблонах сохранения, загрузки и обработки файлов.

Кнопка «Save Background» с помощью метода save_background() делает и сохраняет одиночную пару снимков «{object_name}_rgb_background.png» и «{object_name}_depth_background.npy».

При нажатии на кнопку «Start» запускается процесс сканирования вызовом метода scanning_process(), внутри которого создаётся отдельный поток QThread scanning_process_thread для выполнения всех операций, связанных с процессом сканирования. Кнопка «Start» при этом переименовывается в «Abort», что позволит пользователю прервать выполнение процесса. По завершении сканирования, вне зависимости от способа окончания, – кнопка возвращается к первоначальному состоянию «Start».

В начале работы scanning_process_thread считываются заданные пользователем параметры, представленные в таблице 6.4.

Далее выполняется и сохраняется первая пара снимков. После этого осуществляется сетевое взаимодействие с установкой. На первом этапе передаётся команда, содержащая параметры сканирования: скорость, ускорение, число снимков, угол поворота. После, в зависимости от действий пользователя, передаются команды «продолжить» или «прервать».

Получив от микроконтроллера подтверждение завершения поворота, приложение делает очередную пару снимков. Снимки сохраняются в выбранный каталог с именами «{object_name}_rgb_{angle}.png» и «{object_name}_depth_{angle}.npy», где {object_name} – имя объекта, указанное пользователем, {angle} – текущий угол поворота платформы.

Таблица 6.4 – Параметры сканирования

Поле интерфейса Параметр Значение по умолчанию Диапазон

speed_spinBox Скорость поворота платформы, °/с 10 1–60

acceleration_spinBox Ускорение поворота, °/с² 3 1–20

numberOfShots_spinBox Количество снимков 18 ≥1

rotationAngle_spinBox Угол поворота за один шаг

в градусах 20 1–360

6.2.5 Во вкладке «Frames Processing» функционал кнопки «Choose Folder» и поля «Object Name» идентичен представленным во вкладке «Scanning» за исключением того, что в данной вкладке эти данные используются для загрузки изображений.

Смена значений флажков manualAdjustment_checkBox, videoMode_checkBox, onOffImage_checkBox, showCountour_checkBox обрабатывается методами frameProc_ManAdjustment(), frameProc_Set_camOutpMode(), frameProc_Set_contourOutpMode() соответственно.

По нажатию на кнопку «Apply» вызывается метод get_clouds_from_frames(), в котором для каждой пары RGB- и глубинных снимков выполняется следующий алгоритм:

1. Применение медианного фильтра к изображению глубины для подавления шума.

2. Выполнение вычитания фонового изображения глубины из изображения сс объектом.

3. Бинаризация результата вычитания для отделения объекта от фона.

4. Морфологическая обработка для устранения шумов и закрытия разрывов в маске с применением эрозии и дилатации.

5. Определение контуров объекта.

6. Обрезание изображений по выделенной маске.

7. Построение облаков точек на основе полученных обрезанных изображений с учётом заданной дистанции отсекания.

После обработки визуализируется в отдельном окне заданное пользователем число облаков. Все облака сохраняются в выбранный с помощью «Folder To Save» каталог под именем «{object_name}_cloud_{angle}.ply». Характеристики задаваемых пользователем параметров приведены в таблице 6.5.

При активных флажках использования интринсиков depthCameraIntrins_checkBox и RGBCameraIntrins_checkBox в обработке используются либо параметры одной выбранной камеры, либо обеих с учетом назначения каждой. Значения параметров извлекаются из соответствующих полей ввода параметров камер: width, height, Fx, Fy, Cx, Cy. Значения по умолчанию приведены в подразделе 3.2.3.

Таблица 6.5 – Параметры обработки RGB и Depth изображений

Поле интерфейса Назначение Значение по умолчанию Диапазон

depthCameraIntrins_checkBox Использовать интринсики камеры глубины True Bool

CameraIntrins_

checkBox Использовать интринсики RGB-камеры False Bool

numberOfClouds_

spinBox Количество облаков для отображения после обработки 0 ≥0

depthTrunc_

spinBox Порог отсечения точек по глубине (в метрах) 1,5 0,6–8

morphKernelSize_

spinBox Размер ядра медианного фильтра 5 3–15 (нечётные)

Если установлен флажок useCustomProcessing_checkBox, применяется пользовательская обработка изображений.

Для реализации пользовательской обработки изображений необходимо определить функцию с именем «custom_rgbd_images_process», которая должна принимать три аргумента:

– depth_list: список изображений глубины np.ndarray типа uint16;

– color_list: список цветных изображений в формате np.ndarray типа uint8;

– angles_list: список углов в формате np.ndarray типа uint8.

Возвращать функция должна список объектов pcd_list, содержащих облака точек типа o3d.geometry.PointCloud.

Имя файла с пользовательской функцией указывается в поле framesProcessing_useCustomProcessing_lineEdit. Загрузка и подключение модуля осуществляется методом load_user_rgbd_proc_function(), в котором выполняется:

– spec = importlib.util.spec_from_file_location (name, filepath) – создание спецификации модуля;

– user_module = importlib.util.module_from_spec (spec) – создание модуля из спецификации;

– spec.loader.exec_module(user_module) – загрузка модуля;

– user_func = getattr (user_module, "custom_rgbd_ images_process", None) – извлечение функции.

Далее вызывается apply_user_rgbd_proc_function (user_func, depth_list, angles_list, color_list).

6.2.6 Во вкладке «Clouds Processing» по нажатию на кнопку «Open3D space» происходит вызов метода open_3d_space(), в котором происходит создание окна для вывода всех облаков точек .

Для загрузки, фильтрации и начального смещения облаков точек используется кнопка «Choose Files and Upload», вызывающая метод choose_n_upload_clouds(), в котором вначале вызывается диалоговое окно для выбора файлов формата «.ply», где пользователь должен выбрать нужные файлы с облаками точек «{object_name}_{angle}.ply», после этого происходит фильтрация шумов и по дальности, а также смещение с вращением, если заданы соответствующие параметры.

Характеристики задаваемых пользователем параметров загрузки и предобработки приведены в таблице 6.6.

Таблица 6.6 – Параметры фильтрации и смещения облаков точек

Поле интерфейса Назначение Значение по умолчанию Диапазон

uploadFiltration_

checkBox Применить фильтрацию после загрузки True Bool

filtrationNeighbors_

spinBox Число соседей фильтра 40 20–100

outlierSensitivity_

doubleSpinBox Агрессивность фильтра 1.0 0,5–2,0

enableDistanceFilter_

checkBox Применить фильтр по расстоянию True Bool

uploadFilterMax

Distance_doubleSpinBox Минимальная дальность, м 0,6 0,6–8

uploadFilterMin

Distance_doubleSpinBox Максимальная дальность, м 8 0,6–8

preloadShift_checkBox Применить начальное смещение False Bool

Смещение и вращение облаков задаются через значения Shift_X/Y/Z для смещения вдоль осей (в метрах) и Rotation_XY/XZ/YZ – поворота в соответствующих плоскостях (в градусах), которые задаются соответствующих числовых полях.

Для корректировки положения отдельных облаков используется флажок manualAdjustmentIsEnabled_checkBox. При его активации становятся доступны ползунки смещения и поворота, такие как manualAdjustmentShiftX_horizontalSlider, manualAdjust mentRotShiftXY_horizontalSlider и аналогичные.

Корректировка работает следующим образом: по выбранной пользователем строке в таблице определяется активное облако, текущие значения его положения отображаются на ползунках. При изменении положения ползунков применяется соответствующее смещение и/или поворот облака.

По нажатию на кнопку «Apply and Process» вызывается метод megre_clouds(), в котором выполняется сшивание облаков, отмеченных флажком «Merge» в таблице по алгоритму ниже, если заданы соответствующие параметры:

1. Разрежение точек с помощью воксельной сетки.

2. Вычисление нормалей.

3. Глобальная регистрация.

4. ICP регистрация.

5. Заполнение дыр в облаке.

6. Фильтрация артефактов.

7. Сглаживание.

8. Оптимизация трансформаций для устранения накопленных ошибок при попарном совмещении отдельных облаков.

Глобальная регистрация может выполняться методом как RANSAC, так и FGR, или использовать гибридную опцию. Количество пробегов задается числом множителей, используемых для вычисления максимальной допустимой дистанции между соответствующими точками и заданных в поле ввода «Feature Radius Multipliers» как строка вида «x3.0, x5.0, x8.0». Наилучший результат выбирается по критерию fitness.

ICP-регистрация может включать два этапа: coarse (грубая) и fine (точная), если они активированы соответствующими флажками. Характеристики задаваемых пользователем параметров сшивания облаков и постобработки приведены в таблице 6.7.

Таблица 6.7 – Параметры сшивания и постобработки облаков точек

Поле интерфейса Назначение Значение по умолчанию Диапазон

1 2 3 4

downsamplingMode_

comboBox Метод вокселизации «fixed» «fixed» | «adaptive»

downsampling_vovel

Size_doubleSpinBox Размер вокселя (м) для «fixed» 0.002 0,001–0,1

downsamplingMinPnts

PerVoxel_spinBox Минимальное число точек в вокселе для «adaptive» 5 1–20

normalsCalculation_

checkBox Выполнить расчет нормалей True Bool

normalEstimation

Radius_doubleSpinBox Радиус области поиска соседей, r×voxel_size 2.0 1.0–10.0

globalRegistrationIsEnabled_checkBox Выполнить глобальную регистрацию 8 0,6–8

featureRadius

Multipliers_lineEdit Множители радиуса глобальной регистрации «x3.0, x5.0, x8.0» x2.0–x10.0

aligmentMethod_

comboBox Метод глобальной регистрации «RANSAC» «RANSAC» | «FGR» | «HYBRID»

ransacMaxIterations_

spinBox Максимально количество итераций RANSAC, K 1000 10–5000

Продолжение таблицы 6.7

1 2 3 4

ransacConfidence

Level_doubleSpinBox Вероятность нахождения корректного соответствия 0,999 0,8–0,999

ransacRandomSamples_

spinBox Максимальное расстояние между соседями в RANSAC, d×voxel_size 7 5–20

fgrMaxIterations_

spinBox Максимальное количество итераций FGR, K 64 30–200

fgrDivisionFactor_

doubleSpinBox Максимальное расстояние между соседями в FGR, мм 1,4 1,2–4,0

edgeLengthVerifi

cationCheck_checkBox Проверить согласованность масштаба True Bool

distanceVerificationCheck_checkBox Проверить расстояния между сопоставленными точками False Bool

normalsVerification

Check_checkBox Проверить согласованность ориентации нормалей False Bool

icpMethod_comboBox Метод ICP «P2Plane» «P2Point» | «P2Plane»

useCoarseIcpRegist

ration_checkBox Использовать «грубый» ICP True Bool

coarseIcpMax

Iterations_spinBox Максимально количество итераций Coarse ICP 60 30–100

useAbsoluteValues_

coarse_checkBox Использовать абсолютные расстояния True Bool

coarseThresholdAb

solute_spinBox Абсолютное допустимое расстояние между точками в Coarse ICP, мм 1 1–50

coarseThresholdMultiplier_doubleSpinBox Множитель допустимого расстояния между точками в Coarse ICP, n×voxel_size 3 3,0–6,0

coarseIcpRelativeFitness_doubleSpinBox Доля корректных соответствий в Coarse ICP 0,4 0,1–1

coarseIcp

RelativeRmse_spinBox Среднеквадратичное отклонение в Coarse, мм 5 1–40

useFineIcpRegistra

tion_checkBox Использоват 37 ь «точный» ICP True Bool

fineIcpMaxItera

tions_spinBox Максимальное количество итераций Fine ICP 30 20–50

useAbsoluteValues_

checkBox Использовать абсолютные расстояния True Bool

fineThresholdAbsolute_doubleSpinBox Абсолютное допустимое расстояние между точками в Fine ICP, мм 0,5 0,1–5

Продолжение таблицы 6.7

1 2 3 4

coarseThresholdMultiplier_spinBox Множитель допустимого расстояния между точками в Fine ICP, n×voxel_size 1 1,0–5,0

fineIcpRelativeFitness_doubleSpinBox Доля корректных соответствий в Fine ICP 0,9 0,5–1

fineIcpRelativeRmse_doubleSpinBox Среднеквадратичное отклонение в Fine, мм 1 0,5–3

useColorIcp_checkBox Использовать информацию о цвете True Bool

shapeImportance_

doubleSpinBox Вес геометрической составляющей 0,9 0,5–1,0

colorImportance_

doubleSpinBox Вес цветовой составляющей 0,1 0,0–0,5

postprocessingVoxel

Size_doubleSpinBox Множитель размера вокселя для постобработки 1 0,0–4

fillHoles_checkBox Выполнить заполнение дыр True Bool

postprocessingHole

Size_doubleSpinBox Максимальный размер дыр, м 0,05 0,01–0,1

filtering_checkBox Выполнить фильтрацию артефактов True Bool

minArtifactSize_

spinBox Минимальный размер сохраняемых кластеров 100 100–500

filtrArtifactSensitivity_doubleSpinBox Максимальное расстояние между точками в кластере, м 0,02 0,01–0,05

keepLargest_checkBox Сохранить наибольший True Bool

smoothingMethod_

comboBox Метод сглаживания «none» «none» | «mls» | «laplasian»

smoothingRadius_

doubleSpinBox Область влияния Moving Least Squares, м 0,03 0,01–0,05

smoothingIterations_doubleSpinBox Количество проходов алгоритма 2 1–5

edgePreserve_

checkBox Сохранять границы True Bool

smoothingTaubin

Lambda_doubleSpinBox Коэффициент сглаживания для «laplasian» 0,3 0,1–0,5

smoothingTaubinMu_

doubleSpinBox Коэффициент сохранения формы для «laplasian» -0,3 -0,5–0,0

optimization_

checkBox Использовать оптимизацию True Bool

optimizationAlgorithm_comboBox Алгоритм оптимизации «L-M» «L-M» | «G_N» | «Levenberg»

Продолжение таблицы 6.7

1 2 3 4

useLoopClosure_

checkBox Применить петлевые замыкания True Bool

minOverlapPoints_

spinBox Минимальное число точек для замыкания 500 100–1000

edgePruneThreshold_

doubleSpinBox Порог обрезки неточных ребер 0,25 0,1–0,5

useGeodesic_checkBox Использовать геодезические расстояния True Bool

Если установлен флажок useCustomProcessing_checkBox, применяется пользовательская обработка облаков.

Для реализации пользовательской обработки изображений необходимо определить функцию с именем «custom_clouds_process», которая должна принимать список объектов pcd_list, содержащий облака точек типа o3d.geometry.PointCloud, и возвращать облако-результат.

Загрузка модуля аналогична той, что рассматривалась в подразделе 6.2.5.

Кнопка «Save» вызывает метод save_cloud(), выполняющий сохранение выбранного облака точек в файл. Из доступных для сохранения форматов представлены: «PLY», «STL» и «OBJ». При выборе последних двух перед сохранением облако преобразуется в Mesh согласно заданным параметрам реконструкции, представленным в таблице 6.8.

Таблица 6.8 – Параметры преобразования облаков точек в Mesh

Поле интерфейса Назначение Значение по умолчанию Возможные варианты

exportReconstructionMethod_comboBox Метод реконструкции « Poisson» «Ball-Pivot» | «Poisson» | «Alpha-Shapes» | «Delaunay»

exportMeshResolution_comboBox Разрешение Mesh'a «Average» «Low» | «Average» | «High»

exportFinalSmoothing_comboBox Сглаживание «None» «None» | «Laplacian» | «Taubin»

7 Технико-экономическое обоснование разработки и реализации на рынке ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА ДЛЯ 3D СКАНИРОВАНИЯ ОБЪЕКТОВ

7.1 Характеристика программно-аппаратного комплекса

Разработанный дипломный проект является программно-аппаратны м комплексо м для 3D сканирования объектов. Это совокупность физического устройства, представленного вращающейся платформой, управляемой микроконтроллером, и камерой глубины, снимающей параметры объекта, и программного обеспечения, выполняющего сбор и обработку данных для последующего формирования 3D модели , а также реализующего соединение с аппаратной частью и его настройку.

Данный дипломный проект разрабатывается для нужд кафедры ЭВМ УО БГУИР по предложению одного из ее преподавателей, в частности для использования в дисциплинах, смежных с темой обработки изображений. Кроме образовательных целей, проект также может быть использован для сбыта обычным пользователям

для 3D-сканирования объектов и получения оцифрованной трехмерной модели объекта, пригодной для последующей обработки или 3D-печати .

Реализуемое устройство относится к категории стационарных 3D сканеров, использующих смешанный тип сканирования: лазерный и оптический.

Из преимуществ данной работы перед аналогами можно выделить следующие характеристики:

– бюджетность;

– необходимость в лишь одном USB порте в отличие от других стационарных аналогов, которые занимают два USB порта (первый – для камеры и второй – для управления платформой), так как у реализуемого устройства управление платформой происходит посредством беспроводного подключения;

– модульность (реализованные по отдельности блоки получения и обработки данных);

– возможность использовать получаемые данные не только для формирования файла с трехмерной моделью, но и любым другим способом по усмотрению пользователя.

7.2 Расчет экономического эффекта от производства программно-аппаратного комплекса

Для определения результата от вложения инвестиций в разработку и создание программно-аппаратного комплекса необходимо определить его отпускную цену на основе расчёта затрат на производство аппаратной части и разработку программной части.

7.2.1 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 рыночных цен. Исходные данные и результаты расчётов заносятся в таблицы 7.1 и 7.2.

Расчёт затрат по статье «Основные и вспомогательные материалы», в которую включается стоимость необходимых для изготовления изделия основных и вспомогательных материалов (как для конструкции, так и для печатной платы), осуществляется по формуле

$$РМ=Ктрi=1nНpiЦотпi,$$

(7.1)

где Ктр – коэффициент транспортных расходов;

n\ – номенклатура применяемых материалов;

Нpi – норма расхода материала i-го вида на единицу изделия, нат. ед./шт.;

Цотпi – цена за единицу материала i-го вида, р.

Таблица 7.1 – Расчёт затрат на основные и вспомогательные материалы

Наименование материала Ед. изм. Норма расхода материала Цена за единицу материала, р. Сумма, р .

1 2 3 4 5

1 Припой 0.8/1000г S-Sn60Pb40 SW26/3/2.5 % г 50,00 0,17 8,50

2 ЛТИ-120 LUX флюс для пайки (бесспиртовой) мл 25,00 0,31 7,75

3 Пластик ABS г 1500,00 0,05 75,00

4 Провод ПВАМ-1.0 коричневый / до 48 В м 7,00 0,85 5,95

5 Профиль станочный 2020 V-слот м 0,9 19,98 17,99

6 Профиль станочный 2040 V-слот м 0,68 29,97 20,38

7 Винт М5х8 мм полусф. головка, цинк, кл.пр. 5.8, DIN 7985 STARFIX шт 50 0,09 4,32

8 Винт М8х35 мм полусф. головка, цинк, кл.пр. 5.8, DIN 7985 STARFIX шт 6 0,62 3,70

Продолжение таблицы 7.1

1 2 3 4 5

9 Винт с полукр.гол. М 3х8 оцинк.DIN 7985 FixHaus шт 30 0,08 2,38

10 Винт М4х30 мм полусф. головка, цинк, кл.пр. 5.8, DIN 7985 STARFIX шт 5 0,22 1,10

11 Шайба М8 усиленная, цинк, прочн.4,8 DIN9021 EKT B006706 шт 10 0,19 1,90

12 Гайка М8 шестигр., нерж.сталь (А2), DIN 934 STARFIX шт 6 0,47 2,80

13 Гайка М 4 оцинк. DIN 934 FixHaus шт 5 0,05 0,03

14 Втулка резьбовая М4.0, D=5.2, L=6 / гайка, латунь шт 5 0,14 0,70

15 Гайка-М3-DIN555-PA66 шт 7 0,40 2,8

16 KLS8-0215 M3-15 (НТР-315), Стойка пластмассовая для печатных плат 15мм шт 7 0,57 3,99

Итого 162,82

Всего с учетом транспортных расходов (1,1) (Рм) 179,20

Расчёт затрат по статье «Покупные комплектующие изделия, полуфабрикаты», в которую включается стоимость необходимых для изготовления изделия комплектующих механических и электронных изделий, а именно интегральных микросхем совместно с полупроводниковыми приборами, осуществляется по формуле

РК=Ктрi=1mNiЦотпi,

(7.2)

где Ктр – коэффициент транспортных расходов;

m – номенклатура применяемых комплектующих;

N_i – количество комплектующих i-го вида на единицу изделия, нат. ед./шт.;

Цотпi – цена за единицу комплектующего i-го вида, р.

Таблица 7.2 – Расчё 1 т затрат на комплектующие изделия

Наименование комплектующего Количество на изделие, шт. Цена за единицу, р. Сумма 4 , р.

1 Аккумулятор Sony US18650VTC3 (Li-ion, 1600mAh, 3.7V) 3 10,30 30,90

2 16-0803-9, отсек батарейный Li-ion, 3x18650 (на плату) 1 12,00 12,00

3 Переключатель клавишный RS-7201-C-C3 / с фиксацией, 250VAC, без подсветки красный 1 1,00 1,00

4 BMS 3S (12.6В 40А BALANCE) контроллер заряда с защитой и балансировкой на 3 АКБ 1 14,00 14,00

5 Модуль Mini-360 понижающий DC-DC преобразователь напряжения 1 1,18 1,18

6 Драйвер шагового двигателя Reprap Drv8825, фиолетовый текстолит 1 6,00 6,00

7 Шаговый мотор NEMA 17HS8401-01 54 мм, 2А 1 103,36 103,36

8 Модуль WROOM ESP32-S WiFi 4MB, 38 контактов 1 22,44 22,44

9 Камера глубины Orbbec Astra 1 295,00 295,00

10 DSM-OLEDv2-0.96-4P-YB, 0.96-дюймовый модуль с дисплеем OLED, разрешение 128x64 пикселей, интерфейс I2C, драйвер SSD1306, желто-синий 1 14,85 14,85

11 KAN1211-1001B 12x12x10 mm Кнопка тактовая 3 0,16 0,48

12 Резистор переменный осевой R-0904N-B100K L20 1 0,36 0,36

13 Подшипник 1000098 ZZ (619/8 ZZ, 698 ZZ) 6 1,01 6,06

14 Подшипник 61800 2RS (6800 2RS), размер 10x19x5 1 4,80 4,80

15 Резистор выводной 470 0.25W ±5 % 4 0,04 0,16

16 Резистор выводной 10K 0.25W ±5 %, CF1, 4W-10K 3 0,07 0,21

17 Конденсатор электролитический выводной 100uF 16V 5x11 105C (JWCO) 2 0,05 0,10

18 Светодиод выводной GNL-5013UBC (14V), 14В 3 0,47 1,41

Итого 514,31

Всего с учетом транспортных расходов (1,1) (Рк) 565,73

Расчёт общей суммы прямых затрат на создание аппаратной части представлен в таблице 7.3.

Таблица 7.3 – Расчёт общей суммы прямых затрат на создание аппаратной части

Показатель Сумма, р.

1 Сырье и материалы 179,20

2 Покупные комплектующие изделия 565,73

Всего прямых затрат на производство аппаратной части (Зрач)

744,93

7.2.2 Расчёт затрат на основную заработную плату необходимых специалистов осуществляется исходя из состава и численности команды, размера месячной заработной платы каждого участника команды, а также трудоемкости работ.

Расчёт затрат на основную заработную плату команды сборщиков представлен в таблице 7.4.

Так как при расчёте заработной платы используется среднемесячная заработная плата в Республике Беларусь для сотрудников ИТ-отрасли, то премии и иные стимулирующие выплаты не рассчитываются.

Таблица 7.4 – Расчёт затрат на основную заработную плату специалистов

Категория

разработчика Месячная

заработная плата, р. Часовая

заработная плата, р. Трудоемкость

работ, ч Итого, р.

Инженер-электроник 1600,00 10,00 4,00 40,00

Программист 1600,00 10,00 4,00 40,00

Итого 80,00

Всего затрат на основную заработную плату разработчиков (Зо)

80,00

7.2.3 Расчёт общей суммы затрат на разработку программной части программно-управляемого комплекс 1 а 4 совместно с его сборкой представлен в таблице 7.5.

Дополнительная заработная плата специалистов рассчитывается по 4 формуле

Зд=Зо· Нд100,

(7.3)

где Нд – норматив дополнительной заработной платы. В расчётах примем за 15%.

Отчисления на социальные нужды рассчитывается по формуле

Рсоц= (Зо+ Зд)·Нсоц 100,

(7.4)

где Нсоц – ставка отчислений в ФСЗН, размер которой составляет 35 % (29 % предназначено для пенсионного страхования, 6 % – социального страхования).

Расходы на разработку и сборку изделия рассчитывается по формуле

Зр= Зо+ Зд+ Рсоц 11 .

(7.5)

Таблица 7.5 – Расчёт затрат на разработку и сборку комплекса

Наименование статьи затрат Расчёт по формуле

(в таблице) Сумма, р.

1 Основная заработная плата специалистов Таблица 7.4 80,00

2 Дополнительная заработная плата 10 специалистов 80,00·15100

12,00

3 Отчисления на социальные нужды (80,00+12,00)·35 100

32,20

Затраты на 1 разработку и сборку комплекса (Зрпч)

124,20

7.2.4 Формирование отпускной цены программно-аппаратного комплекса осуществляется в соответствии с методикой, представленной в таблице 7.6.

Расчёт суммы затрат на производство программно-аппаратного комплекса осуществляетс 9 я по формуле

Зпр =Зрач+ Зрпч,

(7.6)

где Зрач и Зрпч – затраты на производство со сборкой аппаратной части и разработку с настройкой программной части, представленные в таблицах 7.3 и 7.5 соответственно.

Расчёт накладных расходов осуществляется по формуле

Рнакл=Зпр ·Ннакл100,

(7.7)

где Ннакл – норматив накладных расходов (по фактическим данным предприятия или на уровне 60–70 %). В расчётах используется значение Ннакл, равное 60 %.

Расчёт расходов на реализацию осуществляется по формуле

Рреал=Зпр ·Нреал100,

(7.8)

где Нреал – норматив расходов на реализацию. В расчётах используется значение Нреал, равное 20 %.

Расчёт полной себестоимости осуществляется по формуле

Сп =Зпр + Рнакл+ Рреал,

(7.9)

Расчёт плановой прибыли, включаемой в цену, осуществляется по формуле

Пед= Сп·Рпр100,

(7.10)

где Рпр – рентабельность продукции. В расчётах используется значение Рпр, равное 35 %.

Расчёт отпускной цены осуществляется по формуле

Цотп= Сп+ Пед .

(7.11)

Таблица 7.6 – Расчёт отпускной цены

Показатель Расчёт по формуле (в таблице) Сумма, р.

Затраты на производство аппаратной части (Зрач)

Таблица 7.3 744,93

Затраты на разработку программной части и сборку комплекса(Зрпч)

Таблица 7.5 124,20

Сумма затрат на производство программно-аппаратного комплекса 794,93+124,20 919,13

Накладные расходы 919,13·60,00100

551,48

Расходы на реализацию 919,13·2,00100

18,38

Полная себестоимость 919,13+551,48+18,38 1488,99

Плановая прибыль, включаемая в цену Пед= 1488,99·35,00100

521,15

Отпускная цен 1 а 1488,99+521,15 2010,14

7.2.5 Результатом создания и продажи программно-аппаратного комплекса является прирост чистой прибыли, полученный от их реализации, который рассчитывается по формуле

ΔПч=Пед·Nп1-Нп100,

(7.12)

где Nп – объем производства и реализации аппаратного модуля, 40 шт.;

Пед – прибыль, включаемая в цену, р.;

Нп – ставка налога на прибыль согласно действующему законодательству, % (по состоянию на 2025 г. – 25 %).

ΔПч=521,15·40·1-25100=15634,50 р.

(7.13)

7.3 Расчет инвестиций в производство программно-аппаратного комплекса

Затраты на производство нового изделия включают в общем случае: инвестиции на его разработку, инвестиции в прирост основного капитала и инвестиции в прирост собственного оборотного капитала 1 . 11

Инвестиции в разработку (Ир) изделия могут быть оценены по затратам на разработку нового изделия инженерами предприятия-производителя 11 . 4 Расчёт затрат на основную заработную плату команды разработчиков, представленный в таблице 7.8, осуществляется исходя из состава и численности команды, размера месячной заработной платы каждого участника команды, а также трудоемкости работ.

Расчёт затрат на основную заработную плату команды разработчиков представлен в таблиц 28 е 7.7.

Таблица 7.7 – Расчёт затрат на основную заработную плату команды разработчиков

Категория

разработчика Месячная

заработная плата, р. Часовая

заработная плата, р. Трудоемкость

работ, ч Итого, р 1 .

Инженер-радиотехник 2000,00 12,50 72,00 900,00

Инженер-программист 2000,00 12,50 72,00 900,00

Итого 1800,00

Всего затрат на основную заработную плату разработчиков 1800,00

Инвестиции на разработку нового изделия рассчитывается по формуле

Зр= Зо+ Зд+ Рсоц.

(7.13)

Таблица 7.8 – 10 Расчёт инвестиций на разработку комплекса

Наименование статьи затрат Расчёт по формуле

(в таблице) Сумма, р.

Основная заработная плата разработчиков Таблица 7.7 1800,00

Дополнительная заработная плата разработчиков 1800,00·15100

216,00

Отчисления на социальные нужды (1800,00+216,00)·35 100

705,60

Затрат 9 ы на разработку (Ир) 1800,00+216,00+705,60 2721,60

Инвестиции в прирост основного капитала не требуются, так как производство нового изделия планируется осуществлять на действующем оборудовании в связи с наличием свободных производственных мощностей.

Так как создается уникальный аппаратный модуль, требуется приобретение нового оборотного капитала. Расчёт инвестиции в прирост собственного оборотного капитала осуществляются следующим образом:

1 Определяется потребность в материалах по формуле

Пм=Рм·Nп,

(7.14)

где Рм – затраты на материалы на единицу продукции, расчёт которых представлен в таблице 7.1.

2 Определяется потребность в комплектующих изделиях по формуле

Пк=Рк·Nп,

(7.15)

где Рк – затраты на комплектующие изделия на единицу продукции, расчёт которых представлен в таблице 7.2.

3 Вычисляются инвестиции в прирост собственного оборотного капитала в процентах от годовой потребности в материалах и комплектующих изделиях β (20%) по формул 1 е

Ис.о.к.= β·(Пм+Пк).

(7.16 10)

Вычислим годовую потребность в материалах подставив значения в формулу (7.14), получим

Пм=179,20·40=7168,00.

(7.13)

Вычислим годовую потребность в комплектующих изделиях подставив значения в формулу (7.15), получим

Пк=565,73·40=22629,20.

(7.13)

Вычислим годовую потребность в комплектующих изделиях подставив значения в формулу (7.16), получим

Ис.о.к.= 20 %·(7168,00+22629,20)=5959,44.

(7.13)

7.4 Расчет показателей экономической эффективности инвестиций в производство программно-аппаратного 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.

Вычислим инвестици 1 и в производство программно-аппаратного комплекса по формуле

Ипр=Ир+Ис.о.к.= 2721,60 + 5959,44 =8681,04.

(7.13)

Сравним инвестиции в производство программно-аппаратного комплекса и полученный годовой прирост чистой прибыли 30 . Инвестиции в производство Ипр составляют 8681,04 р.; годовой прирост чистой прибыли ΔПч составляет 15634,50 р. Тогда

Ипр< ΔПч.

(7.13)

Поскольку сумма инвестиций в производство меньше полученного годового прироста чистой прибыли, оценка их экономической эффективности осуществляется на основе расчёта рентабельности инвестиций (затрат) (Return Of Investment, ROI) по формуле

\ ROI=\ ΔПч -(Ис.о.к+Ир)Ис.о.к+Ир ·100 %,

(7.17)

где ΔПч – прирост чистой прибыли от производства и реализации новых изделий, р.;

Ир – инвестиции в разработку нового изделия, р.;

Ис.о.к. – прирост собственного оборотного капитала, р 1 .

Вычислим оценку экономической эффективности, подставив значения в

формулу (7.17), получим

ROI=\ \frac{15634,50\ -(5959,44+2721,60)}{5959,44+2721,60}\ ·100 %=80,10 %.

7.5 Вывод об экономической эффективности программно-аппаратного комплекса по результатам расчетов

Результатом проведения экономического обоснования инвестиций в разработку и реализацию аппаратного комплекса стали полученные выше значения показателей эффективности:

Прирост чистой прибыли, полученной в результате разработки и реализации аппаратного комплекса, составляет порядка 15634,50 белорусских рублей.

Рентабельность инвестиций на разработку аппаратного комплекса достигла отметки в 80,1 %.

В качестве итога можно сделать заключение о том, что разработка и производство программно-аппаратного комплекс 9 а для 3D сканирования объектов является экономически эффективным вложением для организации-разработчика, а реализация проекта с экономической точки зрения целесообразна с точки зрения использования продукта как в образовательных, так и коммерческих целей.

8 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

В качестве источника питания установки с вращающейся платформой необходимо использовать три аккумулятора Li-Ion 18650, место под которые предусмотрено в соответствующем отсеке. Для включения устройства следует нажать круглую красную кнопку, расположенную сбоку панели управления. О подаче питания и готовности к работе сигнализирует зажженный зеленый светодиод.

После включения устройство автоматически переходит в главное меню, вывод которого представлен на рисунке 8.1. Для навигации в нем используется цилиндрическая ручка, поворотом которой выбирается нужный пункт меню. Переход в соответствующий режим работы осуществляется нажатием верхней кнопки. Эта же кнопка используется и для возвращения в меню.

Рисунок 8.1 – Вывод главного меню на дисплее

При переходе в режим «Motor test» на дисплей выводится информация, представленная на рисунке 8.2. В данном режиме ручка используется для задания скорости вращения платформы относительно среднего положения: поворот влево вызывает вращение в одну сторону, вправо – в другую. Нажатие на нижнюю кнопку приостанавливает и возобновляет вращение. Синий светодиод является индикацией движения платформы. Данный режим может быть полезен как для теста электродвигателя, так и подбора значения предпочитаемой скорости.

Рисунок 8.2 – Вывод дисплея в режиме «Motor test»

При переходе в режим «Connection Config» на дисплей выводится информация, представленная на рисунке 8.3. По умолчанию устройство запускается в режиме точки доступа (AP), к которой возможно подключиться, используя имя сети «ESP32-AP» и пароль «12345678».

Нажатие на нижнюю кнопку переключает режим работы между точкой доступа (AP) и режимом станции (STA), в котором устройство подключается к существующей сети пользователя. Название сети и пароль в этом случае загружаются из энергонезависимой памяти и могут быть изменены через прикладное ПО, что будет подробнее описано далее. Вывод «IP: X.X.X.X» означает, что подключение к сети не установлено. Для связи с устройством используются указанный IP-адрес и порт 1234.

Рисунок 8.3 – Вывод дисплея в режиме «Connection Config»

При переходе в режим «Scanning Mode» на дисплее отображается интерфейс, представленный на рисунке 8.4. Только когда устройство находится в этом режиме, оно принимает команду, начинающую сканирование. В других режимах данная команда игнорируется. При выходе из режима, если процесс сканирования был активен, он немедленно прерывается.

Сканирование представляет собой серию поворотов платформы на указанный угол заданное количество раз. Подробнее далее в описании работы с прикладным ПО.

Нажатие нижней кнопки в этом режиме переключает формат отображения прогресса поворота платформы между двумя вариантами, обозначающими либо число выполненных поворотов, либо текущий угол вращения платформы.

а б

Рисунок 8.4 – Вывод дисплея в режиме «Scanning Mode»: а – отображение числа поворотов; б – отображение угла вращения

При переходе в режим «Device Info» на дисплей выводится техническая информация об устройстве, представленная на рисунке 8.5.

Рисунок 8.5 – Вывод дисплея в режиме «Device Info»

Прикладное ПО имеет следующие минимальные системные требования:

− операционная система: Linux/Windows 10/11 (64-разрядная версия);

− процессор: Intel Core i3-4150/AMD FX-6300;

− оперативная память: 8 ГБ;

− видеокарта: NVIDIA GeForce GTX 660/AMD HD 7870;

− свободное место на диске: 10 ГБ.

Для работы с Orbbec Astra должен быть установлен драйвер из архива «astra-win32-driver-4.3.0.22.zip», полученного с официального сайта производителя.

Программа может быть запущена двумя способами.

Первый способ заключается в распаковке содержимого «3DScanPlatform.zip» и запуске файла «Rgbd3DScan.exe». Важно, чтобы рядом с исполняемым файлом находились все остальные файлы из архива.

Второй способ заключается ручном запуске скипта «MainRgbd3DScan.py» из «3DScanPlatform_Source.zip», содержащего все исходные файлы. Присутствие в том же каталоге остальных файлов из архива также обязательно. Кроме вышеперечисленного для второго способа необходимо наличие Python версии 3.9+ и библиотек:

– PyQt6;

– Primesense;

– OpenCV;

– Open3D.

При запуске прикладного ПО открывается окно с пятью вкладками, первой из которых является вкладка «Platform», представленная на рисунке 8.6. Данная вкладка предназначена для работы с подключением к установке с вращающейся платформой.

Рисунок 8.6 – Вкладка «Platform»

Чтобы подключиться к платформе необходимо ввести порт и IP адрес, указанный на дисплее физического устройства, в поля «Platform Port» и «Platform IP» соответственно и нажать кнопку «Connect». Информация о результате подключения будет выведена в отдельном информационном окне. При успешном подключении будет изменен статус «Platform Status», и кнопка «Connect» станет «Disconnect», нажатие которой разрывает подключение.

Чтобы передать платформе имя и пароль сети пользователя, необходимо ввести эти данные в поля «AP SSID» и «AP Password» и нажать кнопку «Send New AP Data».

Используя вкладку «Cam Check», представленную на рисунке 8.7, можно проверить работу камеры и задать ее настройки.

Рисунок 8.7 – Вкладка «Cam Check»

С помощью флажка «Camera Output On/Off» можно включить/выключить вывод RGB камеры и камеры глубины. Пример представлен на рисунке 8.8. Флажки ниже определяют настройки: наложение карты глубины на цветное изображение, включение аппаратной синхронизации двух камер и зеркальное отображение

Рисунок 8.8 – Вывод RGBD данных

В полях под «Resolution» задается разрешение сохраняемых RGB и Depth изображений. В поле «Depth Display Method» можно выбрать метод отображения изображения глубины.

Сохранить выводимую пару изображений можно с помощью кнопки «Make Photo». Путь и имя файла берутся из полей выше. Каталог также может быть задан с помощью кнопки «Choose Folder», открывающей диалоговое окно проводника.

На вкладке «Scanning», представленной на рисунке 8.9 происходит управление процессом сканирования, заключающегося в получении набора RGBD данных сканируемого объекта с разных ракурсов.

Рисунок 8.9 – Вкладка «Scanning»

Все полученные файлы будут сохранены в каталог, указанный в поле «Folder», и который можно задать с помощью кнопки «Choose Folder». Имена файлов

с RGB и Depth данными формируются по шаблонам «{object_name}_rgb_{angle}.png» и «{object_name}_depth_{angle}.npy» соответственно, где {object_name} – имя объекта, указанное в поле «Object Name», {angle} – соответствующий угол ракурса.

Для дальнейшей обработки необходимо сохранить фон без объекта, для чего нужно нажать кнопку «Save Background» – и будут сохранены файлы «{object_name}_rgb_background.png» и «{object_name}_depth_background. npy».

С помощью флажка «Show Cam Output» можно включить/выключить вывод RGB камеры и камеры глубины.

В полях «Speed», «Acceleration», «Number of Shots», «Rotation Angle» задаются скорость поворота платформы, ускорение число снимков и угол поворота соответственно.

Процесс сканирования, а точнее сбора данных, запускается по нажатию на кнопку «Start», которая при активном процессе меняется на «Abort» для его прерывания.

Например, с параметрами, представленными на рисунке 8.9 будут сохранены файлы: «panda_rgb_background.png», «panda_depth_background. npy», «panda_rgb_0.png», «panda_rgb_18.png» ... «panda_rgb_342.png», «panda_depth_0.npy», «panda_depth_18.npy» ... «panda_depth_342.npy».

Обработка RGBD данных и получение облаков точек объекта происходит во вкладке «Frames Processing», представленной на рисунке 8.10.

Рисунок 8.10 – Вкладка «Frames Processing»

Вверху вкладки задаются каталог и имя объекта для загрузки данных.

При активном флажке «Show Cam Output» можно включить/выключить вывод RGB камеры и камеры глубины.

Для получения облака точек объекта его необходимо предварительно обнаружить и выделить на фоне. Для визуализации используется флажок «Show Contour», позволяющий отобразить границы выделенного объекта. В случае смещения каналов, возникающего из-за несовершенства функции аппаратного совмещения, может потребоваться ручная корректировка. При активном флажке «Manual Adjustment» становятся доступны поля для задания сдвига и масштабирования изображения, позволяющие устранить расхождения. Вывод RGBD данных во вкладке «Frames Processing» представлен на рисунке 8.11.

а б в

Рисунок 8.11 – Вывод RGBD данных во вкладке «Frames Processing»: а – изображение глубины; б – цветное изображение; в – цветное изображение со смещением

По нажатию кнопку «Apply» выполняется обработка и преобразование всех пар RGB и Depth снимков в соответствующие облака точек. В обработке применяется медианная фильтрация, для которой в поле «Morph Kernel Size» возможно задать размер ядра.

В поле «Depth Trunc» можно указать порог отсечения точек по глубине в метрах. В «Number of Clouds» – число облаков, которое необходимо вывести после обработки. Пример двух полученных облаков представлен на рисунке 8.12. Полученные облака точек сохраняются с именем формата «{object_name}_{angle}.ply» .

Рисунок 8.12 – Пример двух полученных облаков после обработки RGBD данных

Помимо обработки RGBD способом по умолчанию, предусмотрена возможность использовать пользовательскую обработку. Для этого необходимо реализовать функцию с именем «custom_rgbd_images_process», принимающую три аргумента:

– depth_list: список изображений глубины np.ndarray типа uint16;

– color_list: список цветных изображений в формате np.ndarray типа uint8;

– angles_list: список углов в формате np.ndarray типа uint8.

Возвращать функция должна список объектов pcd_list, содержащих облака точек типа o3d.geometry.PointCloud.

Имя файла с пользовательской функцией указывается в поле «Custom Processing». При этом сам файл должен быть размещен в корневом каталоге программы. Для активации пользовательской обработки необходимо установить флажок «Use Custom Processing».

Для работы с облаками используется вкладка «Clouds Processing», представленная на рисунке 8.13. Поскольку интерфейс вкладки содержит области прокрутки, на изображении показан не весь функционал. Остальные элементы, доступ к которым осуществляется путем прокрутки, представлены в подразделе 3.2.1.

Кнопка «Open 3D Space» используется, чтобы открыть отдельное окно для визуализации облаков точек в 3D-пространстве.

По нажатию на кнопку «Choose Files and Upload» открывается диалоговое окно выбора файлов с именем формата «{object_name}_{angle}.ply» . После файлы загружаются в программу.

Дополнительно могут быть заданы параметры фильтрации и смещения. Их назначение следует из наименований, а подробное описание приведено в таблице 6.6.

Все облака точек заносятся в таблицу в верхнем правом углу интерфейса.

Рисунок 8.13 – Вкладка «Clouds Processing»

Флажки show в таблице используются для указания, какие облака точек необходимо отображать.

При выборе строки в таблице, доступны ручная корректировка положения соответствующего облака с помощью ползунков блока «Manual Adjustment» и его удаление с помощью клавиши Del. Также доступна возможность переименовывать облака с условием сохранения в начале имени строки «{object_name}_».

По нажатию кнопки «Apply and Process» выполняется сшивание облаков, отмеченных флажком «Merge» в соответствии с заданными параметрами, назначение которых следует из их наименований, а подробное описание приведено в таблице 6.7. Ход выполнения отображается в текстовом поле логов, расположенном в правом нижнем углу интерфейса.

После завершения сшивания флажки «Merge» автоматически снимаются, а в таблицу добавляется новое облако – результат объединения.

Пример результата сшивания облаков представлен на рисунке 8.14.

Помимо сшивания облаков способом по умолчанию, предусмотрена возможность использовать пользовательскую реализацию. Для этого необходимо реализовать функцию с именем «custom_clouds_process», которая должна принимать список объектов pcd_list, содержащий облака точек типа o3d.geometry.PointCloud и возвращать облако-результат.

Имя файла с пользовательской функцией указывается в поле «Custom Processing». При этом сам файл должен быть размещен в корневом каталоге программы. Для активации пользовательской обработки необходимо установить флажок «Use Custom Processing».

Рисунок 8.14 – Результат сшивания облаков

Для сохранения облака точек необходимо выбрать его в таблице и нажать кнопку «Save». Результаты могут быть сохранены в одном из поддерживаемых форматов: «PLY», «STL» или «OBJ». Сохранение выполняется в соответствии с заданными параметрами, назначение которых следует из их наименований, а подробное описание приведено в таблице 6.8.

Полученные файлы пригодны для последующей обработки или использования в сторонних программах. В качестве примера можно назвать MeshLab, вывод которой представлен на рисунке 8.15.

Рисунок 8.15 – Полученное облако точек в программе MeshLab

Заключение

В ходе работы над дипломным проектом была выполнена работа по проектированию и разработке программно-аппаратного комплекса для 3D сканирования объектов.

Результатом работы является комбинированная система, включающая в себя физическую установку с вращающейся платформой и прикладное программное обеспечение.

Физическая установка предназначена для дистанционного и автономного управления поворотом объекта на заданный угол в процессе сканирования. С помощью шагового электродвигателя и особенностей конструкции обеспечивается высокая точность позиционирования на любой угол. В ходе работы были вытравлены платы, изготовлен и собран корпус устройства.

Прикладная программа реализует широкий функционал, включающий в себя:

1 Подключение к установке и обмен данными с целью выполнения заданного числа поворотов на определенный угол и управления конфигурации сетевых параметров.

2 Настройку и управление камерой Orbbec Astra, включая работу как с RGB-потоком, так и потоком глубины.

3 Инструменты для обработки RGBD-данных и получения облаков точек.

4 Инструменты для фильтрации, сшивания и другой обработки облаков точек.

Из основных достоинств комплекса можно выделить:

– автономное питание установки;

– модульная архитектура;

– открытое программное обеспечение;

– широкий спектр пользовательских настроек для обработки данных.

К недостаткам относятся:

– отсутствие поддержки других моделей камер;

– наличие большого числа параметров, неочевидных для неподготовленного пользователя, с отсутствием режима автоматической обработки «по умолчанию»;

– функциональность широкая, но не уровня профессиональных решений.

В качестве возможных вариантов развития системы могут быть рассмотрены:

– расширение протокола взаимодействия между установкой и ПО для увеличения дистанционного функционала;

– полная автоматизация построения 3D-модели с интеллектуальным подбором параметров;

– использование AI для генерации CAD-файлов.