

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт машиностроения
(наименование института полностью)

Кафедра «Промышленная электроника»
(наименование)

11.04.04 «Электроника и нанoeлектроника»
(код и наименование направления подготовки)

Электронные приборы и устройства
(направленность (профиль))

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)

на тему Система позиционирования объектов в рабочей области

Обучающийся

И.В. Борисов

(Инициалы Фамилия)

(личная подпись)

Научный

Д.т.н, доцент, В.П. Певчев

руководитель

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

Содержание

Введение.....	5
1 Состояние вопроса	6
1.1 Виды 3D-печати и ее применение.....	8
1.2 Пять критериев 3D-печати.....	10
1.3 3D-принтеры: технологии и применение	11
1.4 Причины остановки печати на FDM 3D-принтере и пути их решения.....	18
2 Датчики и их принципы работы	21
2.1 Инфракрасные датчики препятствий.....	21
2.2 Ультразвуковые датчики.....	27
2.3 Лидары	32
3 Разработка системы позиционирования	43
3.1 Сборка на макетной плате, подключение и исследование первого датчика.....	43
3.2 Подключение и исследование второго датчика	52
4 Создание устройства сканирования рабочей поверхности 3D-принтера.....	59
4.1 Подключение компонентов и написание программы.....	59
Заключение	72
Список используемых источников.....	74
Приложение А	78

Введение

3D-печать, или аддитивное производство – это технология, которая произвела революцию в производстве прототипов и конечных продуктов. С момента своего создания развитие материалов и технологий, а также снижение стоимости оборудования значительно расширили возможности 3D-печати. Эта технология активно используется в различных отраслях, включая авиацию, автомобилестроение, медицину и архитектуру. В то же время, настольные 3D-принтеры стали широко использоваться в повседневной жизни, они позволили обычным пользователям создавать уникальные продукты на основе индивидуальных проектов.

Актуальность 3D-печати в промышленности обусловлена несколькими ключевыми факторами. Во-первых, это позволяет значительно сократить время, необходимое для создания прототипов и внедрения новых продуктов. Компании могут производить детали за часы или дни вместо нескольких недель ожидания. Во-вторых, 3D-печать позволяет создавать геометрические формы, которые сложно или даже невозможно изготовить традиционными методами. Это открывает новые горизонты в проектировании и позволяет разрабатывать более эффективные и легкие конструкции. Кроме того, технология аддитивного производства способствует значительной экономии материалов. Классическое производство часто использует метод литья, при котором детали вырезаются из одного материала, что в свою очередь приводит к образованию большого количества отходов. При 3D-печати материал добавляется слой за слоем, что снижает количество отходов и делает процесс экологически чистым. Это важный фактор в контексте борьбы с загрязнением окружающей среды и необходимости оптимизации использования ресурсов.

3D-печать также широко используется в повседневной жизни. Обычные пользователи могут создавать индивидуальные продукты, такие как декоративные элементы, игрушки, аксессуары для дома и даже запчасти

для различных устройств. Это позволяет не только удовлетворить свои творческие потребности, но и сэкономить на покупке новых вещей. С развитием онлайн-платформ, открывающих доступ к миллионам моделей для 3D-печати, процесс создания собственных изделий стал более удобным и интересным.

Однако, несмотря на множество преимуществ, 3D-печать сталкивается с рядом технических проблем, одной из которых является потеря позиционирования места печати после отключения электроэнергии. В случае неожиданного отключения электроэнергии или системного сбоя печатающая головка может разрушиться, что приведет к дефектам печатного продукта и необходимости начинать процесс заново. Особенно это важно в массовом производстве, где ценна каждая минута, затраченная на производство продукта.

В данной работе будут рассмотрены и предложены решения проблемы позиционирования печатающей головки 3D-принтера после сбоя печати. Основное внимание уделяется разработке методов автоматического определения точного положения печатающей головки для продолжения печати с минимальными потерями качества и времени. Для этого проанализированы существующие технологии и алгоритмы, а также произведены разработка и тестирование нового способа для повышения надежности и точности восстановления печати.

1 Состояние вопроса

Как уже было упомянуто выше, 3D-печать, или аддитивное производство, представляет собой технологию, которая коренным образом изменила подход к созданию прототипов и производству конечных продуктов. С момента своего появления 3D-печать значительно расширила свои возможности, что обусловлено развитием материалов и технологий, а также снижением стоимости оборудования. Эта технология активно используется в различных отраслях промышленности, включая авиацию, автомобилестроение, медицину и архитектуру. В то же время, благодаря доступности настольных 3D-принтеров, она нашла широкое применение и в быту, позволяя обычным пользователям создавать уникальные изделия по индивидуальным проектам.

Актуальность применения 3D-печати в промышленности обусловлена несколькими ключевыми факторами. Во-первых, она позволяет значительно сократить время на создание прототипов и внедрение новых продуктов. Вместо того чтобы заказывать изготовление деталей на сторонних предприятиях и ждать несколько недель, компании могут производить их самостоятельно в считанные часы или дни. Во-вторых, 3D-печать предоставляет возможность создавать сложные геометрические формы, которые трудно или даже невозможно произвести традиционными методами. Это открывает новые горизонты в дизайне и инженерии, позволяя разрабатывать более эффективные и легкие конструкции.

Кроме того, технология аддитивного производства способствует значительной экономии материалов. В традиционном производстве часто используется метод вычитания, когда из цельного куска материала вырезаются детали, что приводит к значительным отходам. В 3D-печати же материал добавляется послойно, что минимизирует количество отходов и делает процесс более экологичным. В условиях глобальной борьбы с загрязнением окружающей среды и необходимости оптимизации

использования ресурсов это является важным преимуществом.

В повседневной жизни 3D-печать также находит широкое применение. Обычные пользователи могут создавать индивидуальные изделия, такие как декоративные предметы, игрушки, бытовые аксессуары и даже запчасти для ремонта различных устройств. Это позволяет не только удовлетворять творческие потребности, но и экономить средства на покупке новых вещей. С развитием онлайн-платформ, предоставляющих доступ к миллионам моделей для 3D-печати, процесс создания собственных изделий стал еще более доступным и увлекательным.

Однако, несмотря на многочисленные преимущества, 3D-печать сталкивается с рядом технических проблем, одной из которых является потеря точности позиционирования печатающей головки после обрыва печати. В случае внезапного отключения электричества или сбоя в системе, печатающая головка может сместиться, что приводит к дефектам в печатаемом изделии и необходимости начинать процесс заново. Это особенно критично в промышленном производстве, где каждая минута простоя обходится дорого.

В данной работе исследовано и предложено решение проблемы позиционирования печатающей головки 3D-принтера после обрыва печати. Основное внимание уделено разработке методов автоматического восстановления точного положения печатающей головки для продолжения печати с минимальными потерями качества и времени. Для этого проведен анализ существующих технологий и алгоритмов, а также разработка и тестирование нового подхода, позволяющего улучшить надежность и точность восстановления печати.

Исследования в данной области имеют высокую практическую значимость, так как они направлены на повышение эффективности и надежности 3D-принтеров как в промышленном, так и в бытовом использовании. Улучшение точности позиционирования после обрыва печати позволит снизить количество брака и сократить время на

восстановление процесса печати, что, в свою очередь, повысит общую производительность и удовлетворенность пользователей.

Как уже было упомянуто выше проблема обрыва печати актуальна по нескольким важным причинам, затрагивающим как промышленное производство, так и повседневное использование 3D-принтеров.

В промышленном производстве важна непрерывность рабочих процессов. Обрывы печати могут нарушить производственный цикл, вызывая задержки и отставания в графике поставок. Непредвиденные простои негативно влияют на ритм работы предприятия и снижают его конкурентоспособность. Особенно это критично в отраслях, где требуется срочное изготовление деталей, таких как медицина или аэрокосмическая промышленность. В таких случаях обрыв печати может поставить под угрозу выполнение срочных заказов, что недопустимо, когда своевременность имеет решающее значение.

Экономическая эффективность также страдает из-за обрывов печати. Частые перезапуски процесса приводят к значительным финансовым потерям, связанным с дополнительным потреблением материалов и времени на переналадку оборудования. Это увеличивает себестоимость продукции и снижает общую экономическую эффективность производства [23]. Кроме того, повторные запуски печати ускоряют износ оборудования, что приводит к сокращению срока службы компонентов и увеличению затрат на их обслуживание и замену.

Технические и инженерные аспекты также играют важную роль. В 3D-печати часто создаются изделия с высокой степенью сложности, требующие максимальной точности. Обрыв печати может повредить структуру таких изделий, делая невозможным продолжение работы без дефектов. Это особенно важно для инженерных деталей, где точность имеет решающее значение. В сфере индивидуального производства, где каждое изделие уникально и создается под заказ, обрыв печати приводит к значительным потерям, так как повторное изготовление уникальных заказов увеличивает

затраты и нарушает сроки исполнения, что негативно сказывается на репутации производителя.

В целом, проблема обрыва печати влияет на надежность и стабильность производственных процессов, экономическую эффективность, техническую точность и пользовательский опыт, делая ее одной из ключевых задач, требующих решения для дальнейшего развития и распространения технологий 3D-печати.

1.1 Виды 3D-печати и ее применение

Термин «3D-печать» охватывает несколько производственных технологий, при которых деталь создается последовательно путем наслоения нового слоя на предыдущий. Эти способы наслоения отличаются друг от друга по таким параметрам как: способ наслоения (технология запекания слоев), материал из которого выполнена будущая модель, качество итоговой поверхности, износостойкость, скорость печати и ее стоимости.

3D-печать применяется в том числе для быстрого создания прототипов моделей.

Можно сказать, что 3D-печать чаще всего используется для прототипирования. Ее способность быстро производить одну деталь позволяет разработчикам продукции проверять и обмениваться идеями экономически эффективным способом. Определение цели прототипа позволит определить, какая технология 3D-печати будет наиболее выгодной. Аддитивное производство может подойти для создания широкого спектра прототипов: от простых физических моделей до деталей, используемых для функциональных испытаний.

Существует несколько видов 3D-печати, к которым относятся:

- стереолитография (SLA),
- селективное лазерное спекание (SLS),
- моделирование наплавленным осаждением (FDM),

- цифровой световой процесс (DLP),
- многоструйный синтез (MJF),
- полиджет,
- прямое лазерное спекание металлов (DMLS),
- электронно-лучевая плавка (ЭЛП).

Выбор правильного способа 3D-печати для требуемых задач требует понимания сильных и слабых сторон каждого процесса и сопоставления этих атрибутов с потребностями разработки продукта. В первую очередь рассмотрено, как 3D-печать вписывается в цикл разработки продукта, а затем распространены типы технологий 3D-печати и преимущества каждой из них. Рассмотрены некоторые распространенные процессы 3D-печати пластиком, а также области, когда каждый из них представляет наибольшую ценность для разработчиков продукции, инженеров и дизайнеров.

1.2 Пять критериев 3D-печати

Как и большинство вещей в жизни, при выборе процесса 3D-печати редко бывает простой ответ. Когда выбираются варианты 3D-печати, обычно рассматриваются пять ключевых критериев, позволяющих определить, какая технология будет отвечать потребностям:

- бюджет,
- механические требования,
- косметический внешний вид,
- выбор материала,
- геометрия.

Несмотря на то, что 3D-печать является почти синонимом быстрого прототипирования, существуют сценарии, когда этот производственный процесс имеет место быть. Обычно эти детали связаны с небольшими объемами и сложной геометрией. Зачастую компоненты для аэрокосмической и медицинской промышленности являются идеальными

кандидатами для производственной 3D-печати, поскольку они часто соответствуют ранее описанным критериям.

Таким образом, данная работа направлена на понимание текущего состояния развития этой технологии, а также перспектив развития 3D-печати, а также предоставить рекомендации для практического применения этой технологии в различных областях.

1.3 3D-принтеры: технологии и применение

Стереолитографические (SLA) 3D-принтеры

Стереолитография (SLA) – это процесс промышленной 3D-печати. Принтер SLA показан на рисунке 1. Они подходят для печати моделей с высоким уровнем детализации, а также где требуется гладкая поверхность модели. Гладкая поверхность модели, которая напечатана с использованием SLA-технологии, выглядит эстетично, а также позволяет использовать модель при прототипировании за счет малой погрешности поверхности. Это способ зачастую используется в медицинской промышленности, а именно для создания протезов и имплантов. Для деталей SLA используются 3D-принтеры Vipers, ProJets и iPros производства 3D Systems. Технология SLA формирует пластиковые детали путем отверждения жидкой термореактивной смолы УФ-лазером. По мере сборки деталей им требуются опорные конструкции, которые удаляются после завершения сборки. После завершения сборки SLA техник извлекает деталь из слоя порошка, счищает лишний материал, а затем подвергает деталь очистке.

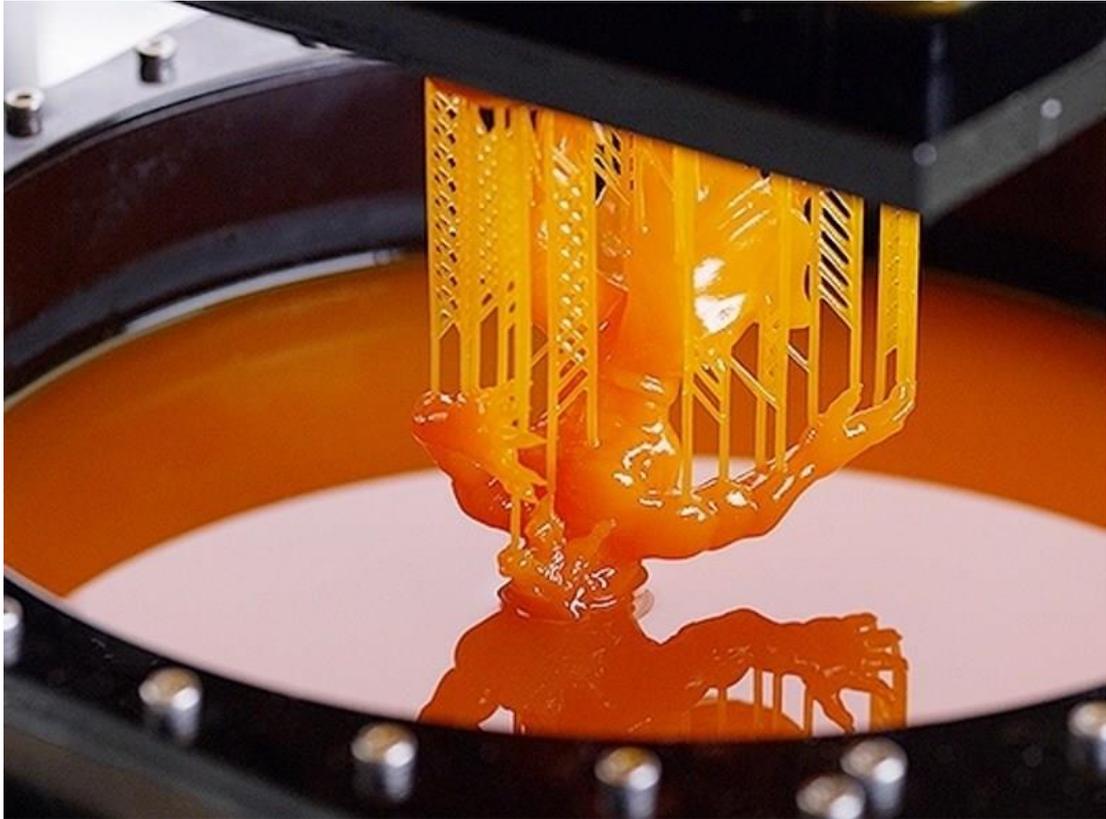


Рисунок 1 – SLA-принтер

Моделирование наплавленным осаждением (FDM)

Каждый 3D-принтер FDM загружается нитью (провод, обычно изготовленный из пластика), который плавится с помощью принтерной головки и выдавливается на рабочую платформу под ней. Процесс показан на рисунке 2.

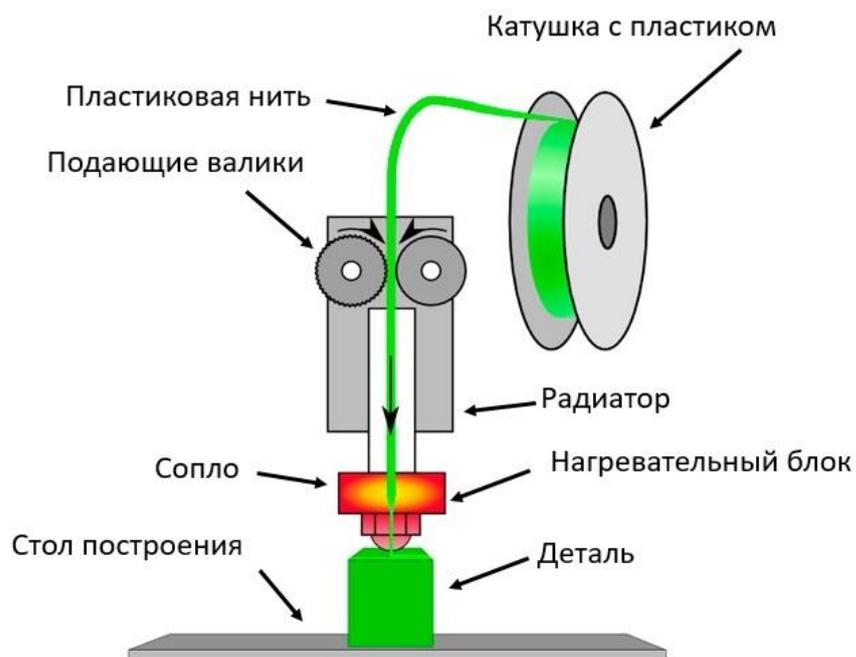


Рисунок 2 -Принцип работы FDM-принтера

Движение печатающей головки относительно платформы печати автоматизировано с помощью электронных двигателей. Печатная головка выдавливает расплавленный материал, чтобы рисовать модель по одному слою за раз. Каждый раз при печати высота печатающей головки по отношению к рабочей платформе увеличивается на фиксированную величину. Выдавливаемый горячий материал слоев хорошо скреплены, причем первый слой прилипает к рабочей платформе.

Принтеры на основе селективного лазерного спекания (SLS)

Селективное лазерное спекание (SLS) плавит порошки на основе нейлона в твердый пластик. Каждая аббревиатура (SLS, SLM, DMLS и DMP) относится к одной технологии 3D-печати и приблизительно схожи по принципу работы: 3D-принтер наносит тонкий слой порошка на рабочую платформу, используя автоматический валик и скребок для его разглаживания. Порошок распределяется при подаче в камеру, расположенную сбоку или над камерой сборки. Затем лазерный луч

перемещается на порошок, рисуя форму первого слоя печатного объекта и сплавляет эту область порошка. Платформа сборки перемещается вниз на шаг, и этот процесс повторяется, чтобы сформировать следующий слой поверх предыдущего. Процесс представлен на рисунке 3.

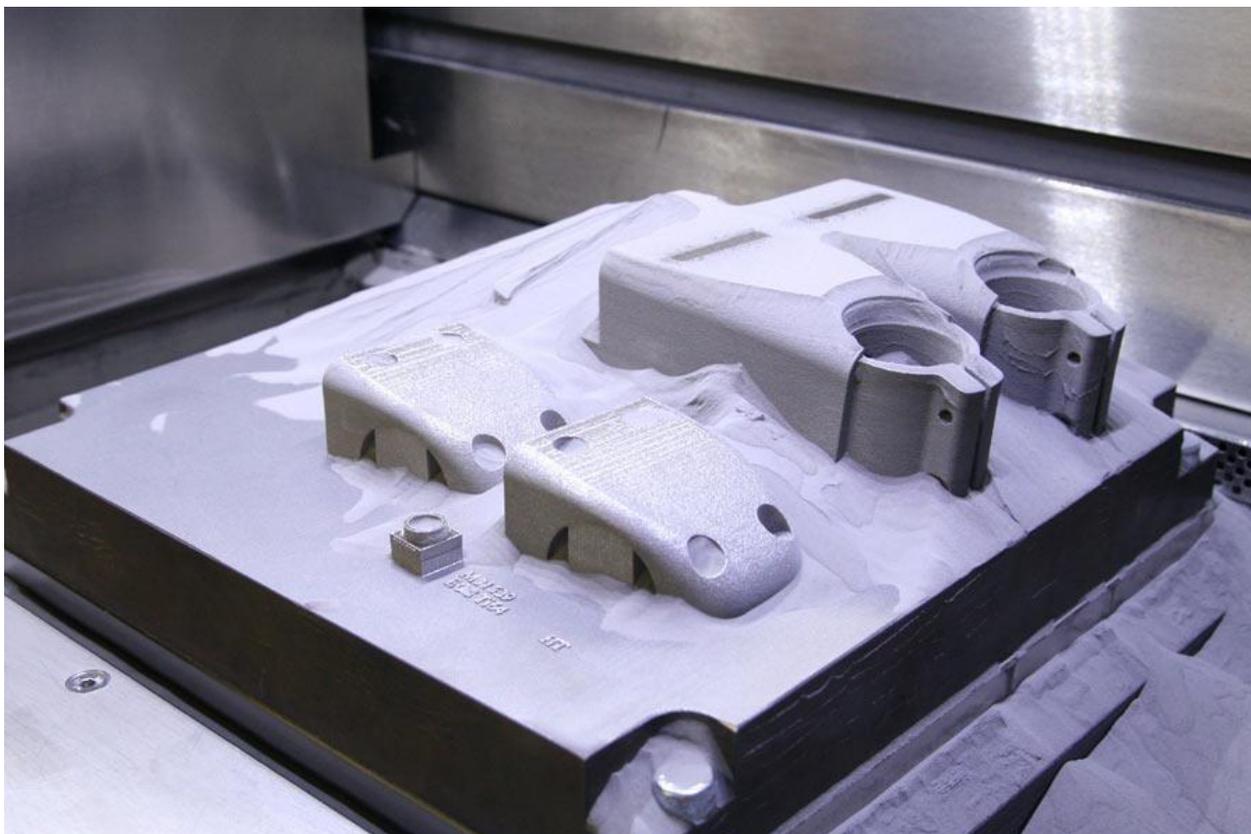


Рисунок 3- SLS принтер

Между слоями образуется крепкая связь. Основные отличия между каждой из технологий заключаются в методах отверждения. Также можно разделить машины на печатаемые только из пластика и оптимизированные для 3D-печати металлом. Эти принтеры часто называют технологиями плавления в порошковом слое.

Принтеры на основе технологии DLP (Digital Light Processing)

Цифровая обработка светом аналогична SLA в том, что она отверждает жидкую смолу с помощью света. Основное различие между этими двумя технологиями заключается в том, что DLP использует экран цифрового

светового проектора, тогда как SLA использует УФ-лазер. Это означает, что 3D-принтеры DLP могут отображать весь слой сборки одновременно, что приводит к более высокой скорости сборки. Несмотря на то, что DLP-печать часто используется для быстрого прототипирования, более высокая производительность делает ее подходящей. Принцип работы FDM принтера показан на рисунке 4.

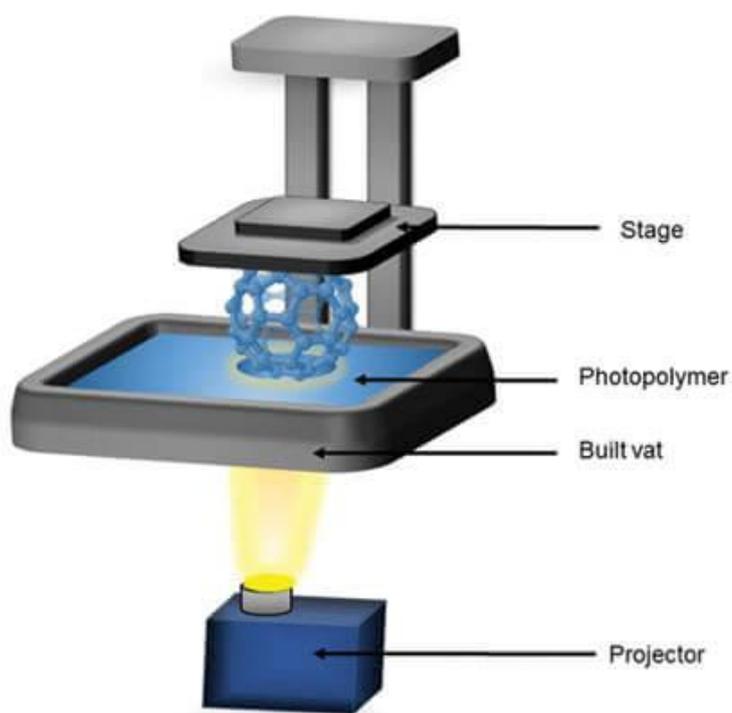


Рисунок 4 – Принцип работы FDM принтера

Принтеры на основе технологии PolyJet

PolyJet – это еще один процесс 3D-печати пластиком, но есть одна особенность. Процесс изображен на рисунке 5.

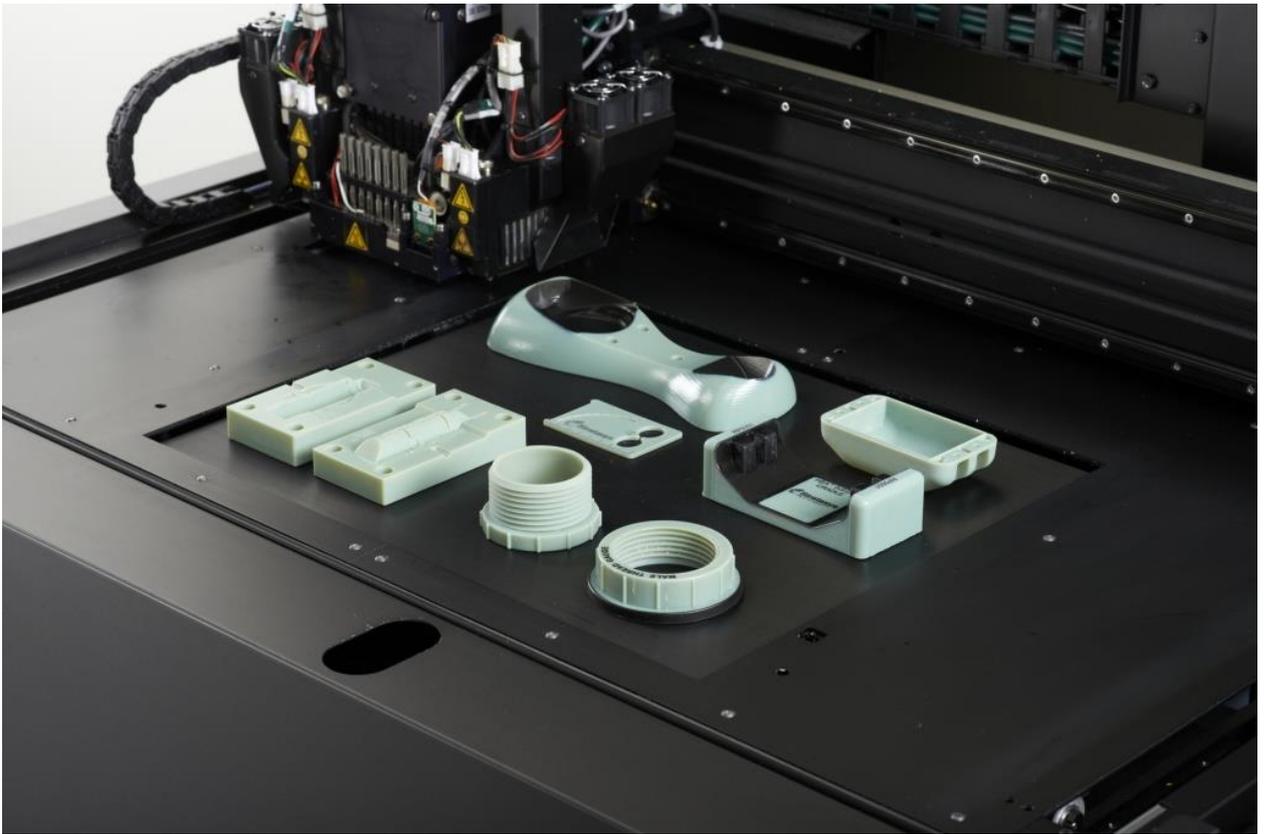


Рисунок 5 – PolyJet принтер

Этот вид 3D принтеров использует различные материалы для печати, начиная от резины и заканчивая металлами.

Принтеры на основе технологии MJF

Вся технология основана на очень точном и одновременно быстром нанесении материала постройки из специальных реагентов и дальнейшем взаимодействии с теплом, излучаемым тепловой лампой. В процессе работы многоструйным способом, который показан на рисунке 6, на тонкий слой порошка наносится плавильный реагент. По внешнему контуру будущей детали подается разделяющий реагент. Принтер работает послойно в соответствии с набором цифровых данных, загруженных из специального программного обеспечения. После нанесения порошка инфракрасная лампа используется для подачи тепловой энергии после каждого напечатанного слоя. Черные части порошкового слоя поглощают больше энергии и расплавляются, в то время как внешний контур остается не расплавленным

благодаря разделяющему реагенту.



Рисунок 6 – MJF принтер

Подведем следующие выводы рассмотренным вариантам:

Для задач, где не требуется массовое производство- 3D печать подходит лучше всего (оптимальное количество моделей от 1 до 70 штук). Также если модель имеет сложную форму, и если этот параметр является важным при проектировании и создании детали, то 3D-печать также оптимально подойдет для этой задачи.

3D-печать использует различные материалы и технологии для печати. Это связано с широким использованием данной технологии в различных областях.

1.4 Причины остановки печати на FDM 3D-принтере и пути их решения

Поскольку наиболее распространенный способ 3D-печати – это FDM-печать, то решено проанализировать причины ее прекращения.

Остановка экструзии «во время печати – в произвольный момент в процессе печати принтер прекращает экструзию пластика произвольным образом в процессе распечатывания изделия» [2], последствия которого показаны на рисунке 7.

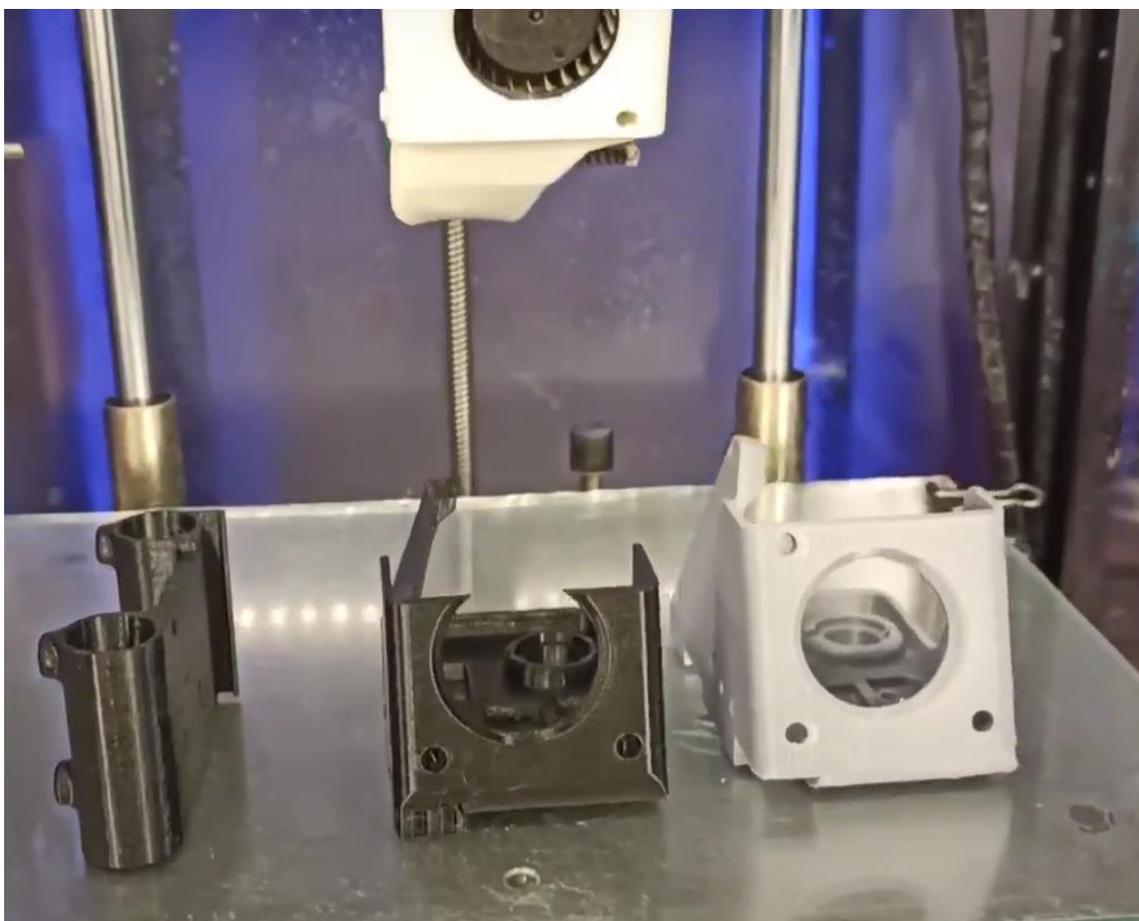


Рисунок 7 – обрыв печать FDM принтера

Если принтер исправно забирал материал и подавал экструзию в начале печати, но вдруг перестал нормально печатать деталь, то есть несколько

возможных объяснений причин [21], которые могли вызвать это явление.

Закончился филамент

Это довольно очевидная проблема, если филамент закончился во время печати, следует остановить печать и загрузить в экструдер новую катушку нити филамента пластика того же типа.

Двигатель экструдера – вращается, а нить – не движется.

Во время печати, экструдер двигателя крутится постоянно, пытаюсь подтолкнуть нить в сопло, чтобы принтер продолжал экструзию пластика. Если попытаться напечатать слишком быстро, мотор может в конечном итоге грызть нить до тех пор, пока ведущая шестерня подачи катушки не повредится.

Филамент зацепился за приводную шестеренку

Так же можно сказать, что филамент остановился против ведущей шестерни. В этом случае нить не намоталась на ведущую шестерню, а застопорилась в сопле, и шестерня ее проела поскольку прижим не отключался. «Во время печати подающая шестерня экструдера крутится постоянно, пытаюсь подать нить в сопло, чтобы принтер мог печатать. Если пытаться печатать слишком быстро, или попытаться выдавить слишком много пластика, эта шестерня мотора экструдера может в конечном итоге начать наматывать нить на себя если это мягкие материалы, а если твёрдые то он начнёт его перемалывать» [2].

Пластик запутался на катушке

Иногда, из-за некачественной намоточной работы поставщиков филамента, или из-за оператора принтера [26], струна пластика запутывается за другие петли пластика на подающей катушке. Чтобы решить эту проблему, «нужно остановить печать, затем снять катушку с держателя, посмотреть, можно развязать ли узел, в другом случае – отрезать ту часть пластика, которая является проблемной, потом вытащить остаток пластика, который остался в экструдере» [5], зарядить вместо него новый пластик.

Экструдер засорился

Если это произошло посреди распечатки, следует лишний раз убедиться, что нить филамента чиста и нет пыли и мусора на бобине катушки филамента. Пыль способна накапливаться и образовывать пробки в расплаве пластика перед соплом. «Для того, чтобы очистить экструдер, следует остановить печать. Далее разобрать экструдер и посмотреть – если экструдер засорился, то аккуратно убрать то, что мешает вращению шестерни и собрать всю конструкцию,» [5] соблюдая все допуски соосностей привода и усилий прижимных механизмов.

Перегрелся драйвер одного из моторов

«Драйвер экструдерного мотора (моторы могут перегреться после длительной печати, или в замкнутом неохлаждаемом объеме, либо в отсутствии радиатора). Мотор экструдера должен работать всегда во время печати. Он постоянно вращается поступательно вперед-назад, толкая и оттягивая пластик. Эти быстрые движения требуют небольшую силу тока, и, если электроника принтера не имеет достаточного охлаждения, это может привести к перегреву двигателя. Драйвера двигателей обычно имеют тепловой порог, при превышении которого, драйвер и мотор перестает работать. Если это произойдет на экструдерном драйвере, то X и Y двигатели оси будут продолжать работать, а экструдер перестанет подавать пластик. Единственный способ быстро решить эту проблему, выключить принтер и подождать пока электроника остынет.» [2] Также можно добавить дополнительный радиатор охлаждения [17].

Термистор начал некорректно работать, сработала защита нагревателя

Еще одной значимой проблемой в 3D печати является необходимость восстановления процесса после обрыва питания [15]. В таких случаях важно точно определить, на каком слое произошел обрыв, чтобы продолжить печать с минимальными потерями качества. «Следует проверить исправность ИБП 220V 500W в цепи питания 3D-принтера. Если ИБП по какой-либо причине не подключен, следует рассмотреть вариант свинцово-кислотной

батареи 14V параллельно блоку питания (напряжение нужно повысить с 12 до 14V). В этом варианте подключения возможна необходимость наличия коммутационных выключателей.» [2]

Исходя из вышесказанного, можно сделать вывод, что при большинстве причин остановки печати оператор 3D-принтера не может наблюдать сбой в его работе, так как находится на удалении от принтера и не имеет возможности постоянно наблюдать за его работоспособностью. В следствии этого возникает необходимость создать универсальный способ отслеживания области, в которой произошел сбой печати, а также информирования оператора о прекращении работы экструдера путем использования как визуальных, так и аудиальных средств вывода информации.

2 Датчики и их принципы работы

Как было упомянуто выше, в нынешнее время широко применяются технологии трехмерного проектирования и трехмерной печати. В частности, популярность набрала технология FDM производства деталей у мелких предприятий (в том числе у частных («домашних») операторов 3D-принтеров). А поскольку в большинстве случаев такие предприятия имеют ограниченный бюджет, то они приобретают наиболее простые и как следствие наименее технологичные FDM-принтеры. Но так как эти устройства также подвержены проблемам внезапной остановки печати, описанных в предыдущей главе, то эта проблема также требует решения. Главными пунктами в решении данной проблемы являются:

- простота установки будущего устройства,
- низкая стоимость будущего устройства.

В связи с этим принято решение использовать датчики расстояния для определения слоя, на котором произошел обрыв печати.

Произведен анализ и сравнение таких датчиков, отличающихся по конструкции и своему принципу действия. А именно выбраны датчики: ультразвуковые, инфракрасные и лидар. Результаты данных анализов и сравнения датчиков приведены далее.

2.1 Инфракрасные датчики препятствий

«Инфракрасный датчик препятствия YL-63 широко применяется в робототехнике, когда нужно определить препятствие. Также данный датчик можно использовать в проектах, в которых необходимо определить препятствия» [8] и расстояние до них. Ниже приведены основные характеристики данного датчика:

- модель: YL-63;
- тип датчика: диффузионный;

- расстояние обнаружения объект: 2-30 см;
- угол обнаружения: 35°;
- напряжение питания: 3.3 В - 5.5 В;
- потребляемый ток: 10 мА;
- цифровой выход: TTL (лог 1 или лог 0);
- диаметр монтажного отверстия: 2.5 мм;
- выходной ток: 15 мА;
- светодиодная индикация срабатывания;
- габариты: 42мм. x 15мм. x 8мм.

Описание датчика препятствия YL-63 (FC-51)

Модуль содержит инфракрасный передатчик (ИК диод) изображенный на рисунке 8, «излучающий свет в прямом направлении (~ 700 нм, этот свет не виден невооруженным глазом, его можно увидеть только камерой), и приемник (фотодиод)»[8], который измеряет отраженное ИК-излучение. Модуль показан на рисунке 8.

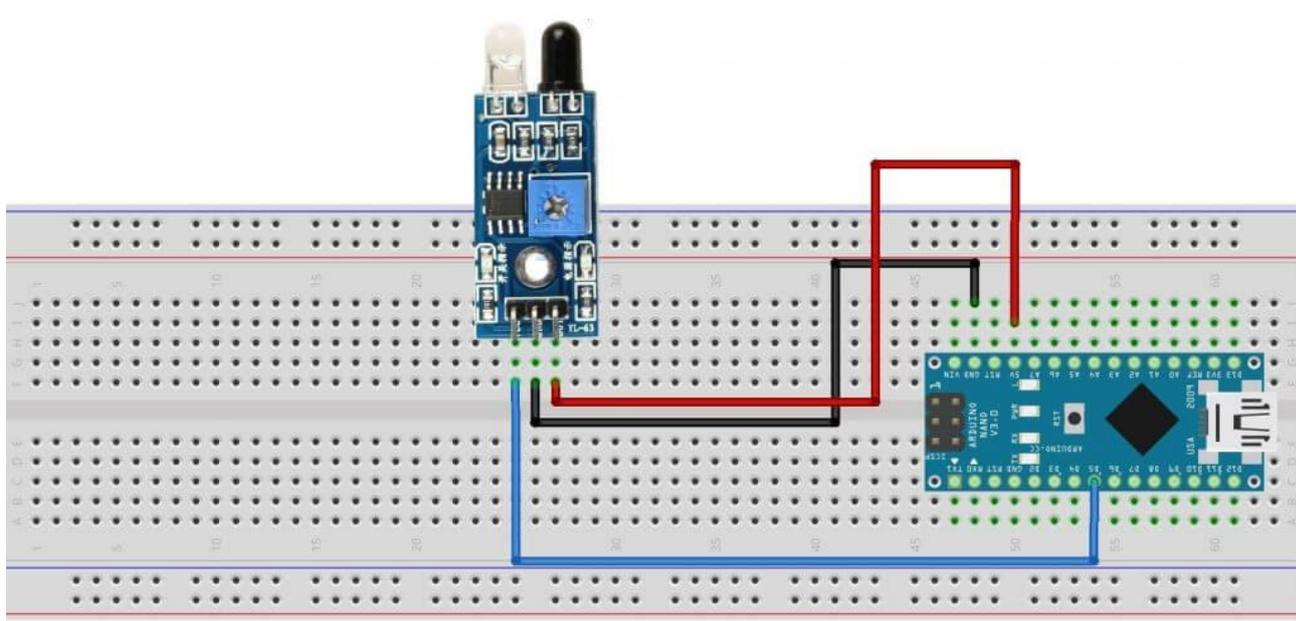


Рисунок 8 – ИК модуль YL-63

«Если отраженный свет достигает определенного порога, на выходе появляется положительный импульс. Так же, количество отраженного излучения зависит от цвета поверхности, от которой оно отражается. Если поверхность белая, то модуль сработает на максимальном расстоянии, если темная, или матовая, излучение не отразится, и модуль не сработает.» [3]
Принцип работы датчика изображен на рисунке 9.

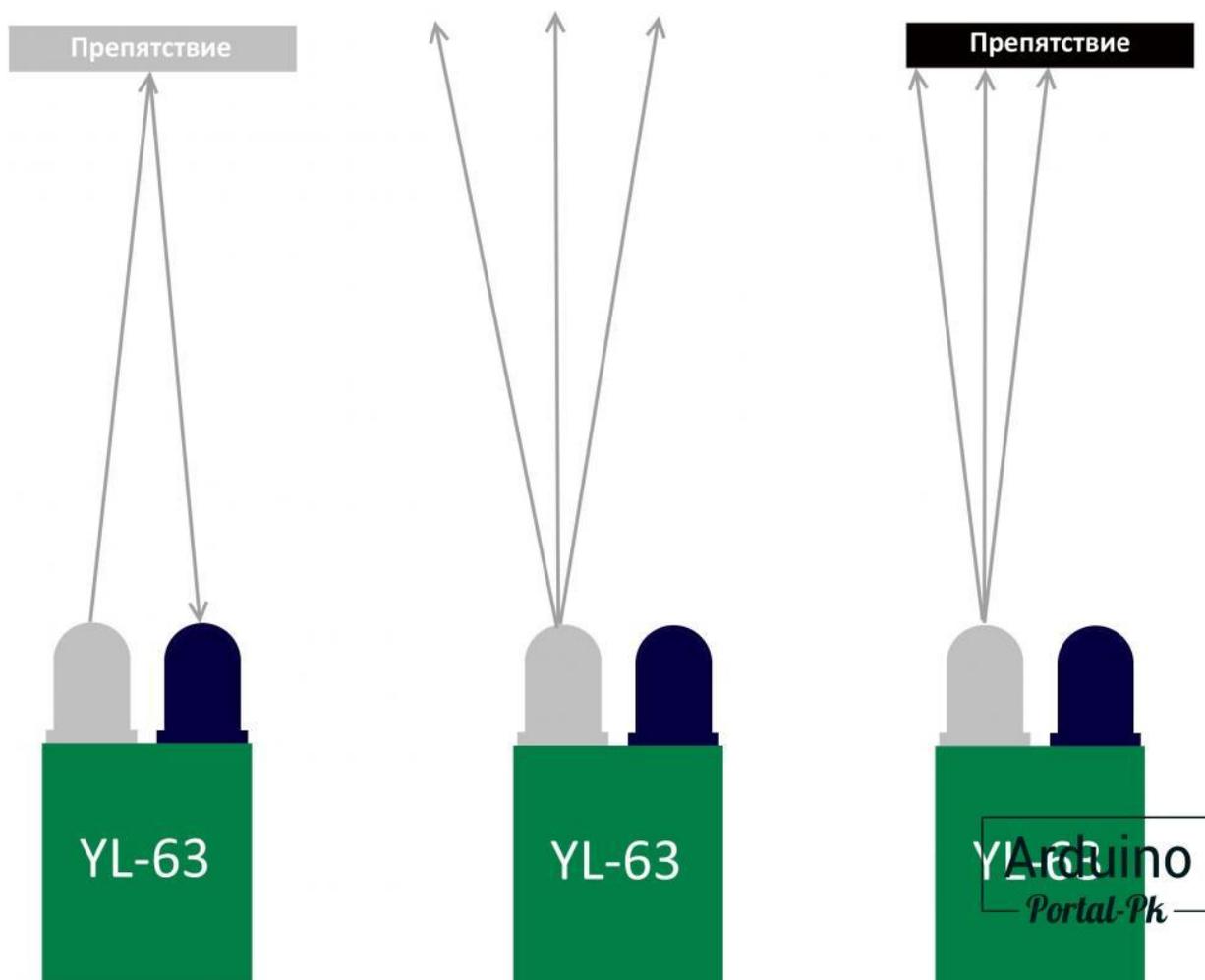


Рисунок 9 – Принцип работы инфракрасного датчика расстояния

«Основная микросхема ИК датчика препятствия – это компаратор LM393, который производит сравнение уровней напряжений на входах INB- и INB+. Чувствительность порога срабатывания задается с помощью потенциометра и в результате сравнений, на выходе OUTB микросхемой U1

формируется «LOW» или «HIGH» [4]. Принципиальная схема ИК модуля препятствия показана на рисунке 10.

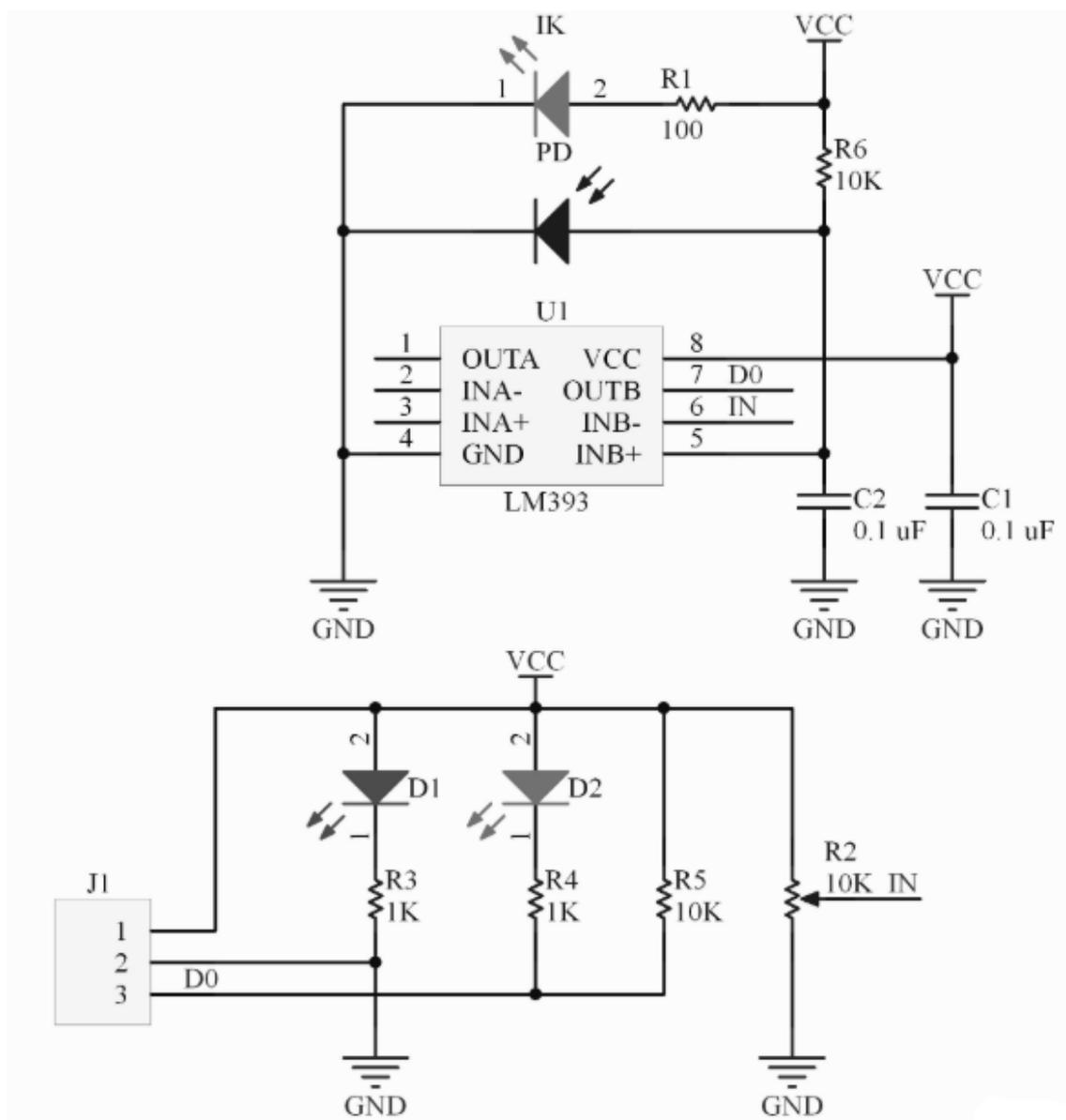


Рисунок 10 – Принципиальная схема ИК модуля

«Оптический датчик YL-63 относится к классу диффузионных. Название группы датчиков возникло из-за, лежащего в основе работы, датчика отражения излучения по множествам направлений – диффузии излучения отражающей поверхностью. Работа устройства заключается определении освещенности фотоприемника. Поскольку YL-63 фиксирует отраженное излучение, то возникает погрешность измерения расстояния,

вызванная различной отражающей способностью поверхностей объектов, изготовленных из разнообразных материалов» [3]. Схема работы диффузии излучения отражающей поверхности изображена на рисунке 11.

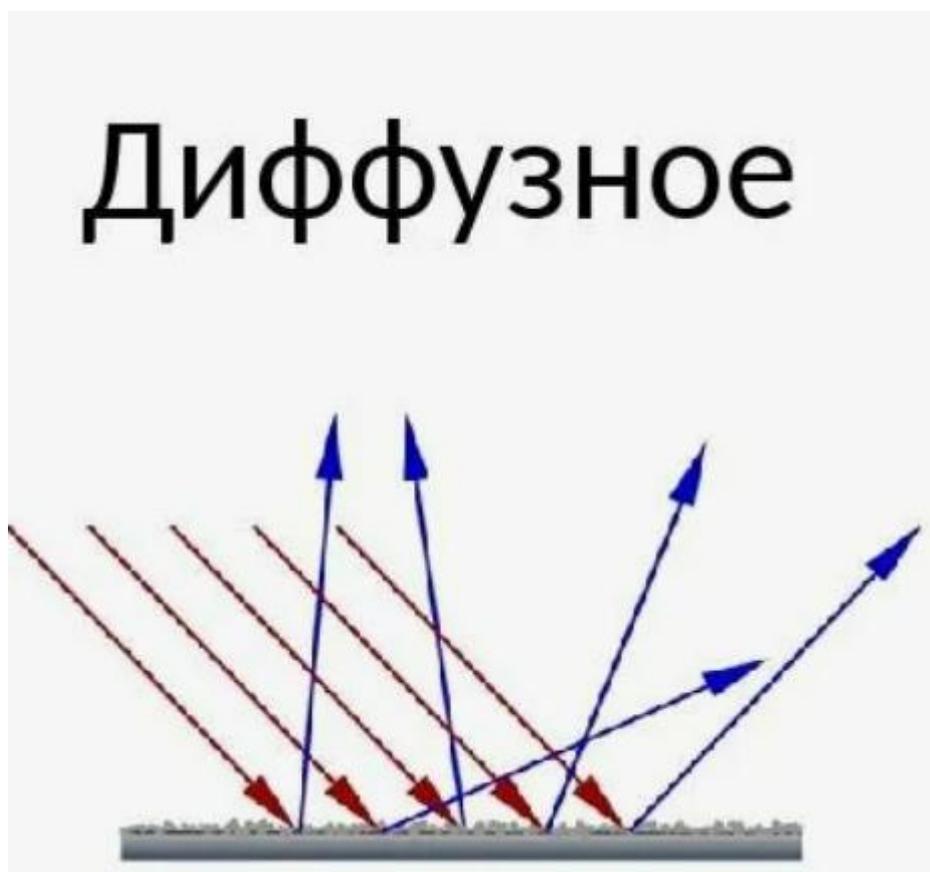


Рисунок 11 – Диффузия излучающего отражения

Коэффициенты расстояния для отражения [22] от различных материалов приведены в таблице 1.

Таблица 1 – Коэффициент расстояния

Материал	Значение коэффициента
1	2
Белая матовая бумага	1
Хлопчатобумажная ткань	0,6
Серый поливинилхлорид	0,57
Белый	0,7

Продолжение таблицы 1

1	2
Черный	0,22
Черная резина	0,28
Матовый алюминий	1,2
Нержавеющая полированная сталь	2,3
Черное глянцевое стекло	1,9
Глянцевый пластик	0,87

«Различное отражение и поглощение излучения разных материалов используются для работы воспринимающего узла тахометра. Предположим, у нас есть двигатель Стирлинга. Требуется узнать количество оборотов в минуту вала двигателя. Нас выручит YL-63. Достаточно приклеить на маховик фрагмент белой бумаги, направить луч датчика на маховик и получим воспринимающий узел тахометра. Для снижения последствий различных помех, обрабатывающим микроконтроллером накапливаются данные, полученные от датчика за короткий промежуток времени, и производится усреднение. Датчик YL-63 может работать в приборах, не имеющих» [4] микроконтроллера.

«Индикаторы на плате YL-63. На плате модуля расположены два индикатора. Свечение зеленого сообщает о включении питания. Красный светодиод светится, если в зоне обнаружения находится объект» [6].

Датчик препятствия YL-63 (FC-51) имеет вилку разъема [14] из трех контактов:

- VCC – питание,
- GND – общий провод,
- OUT – выход.

На рисунке 12 изображена вилка разъема YL-63.

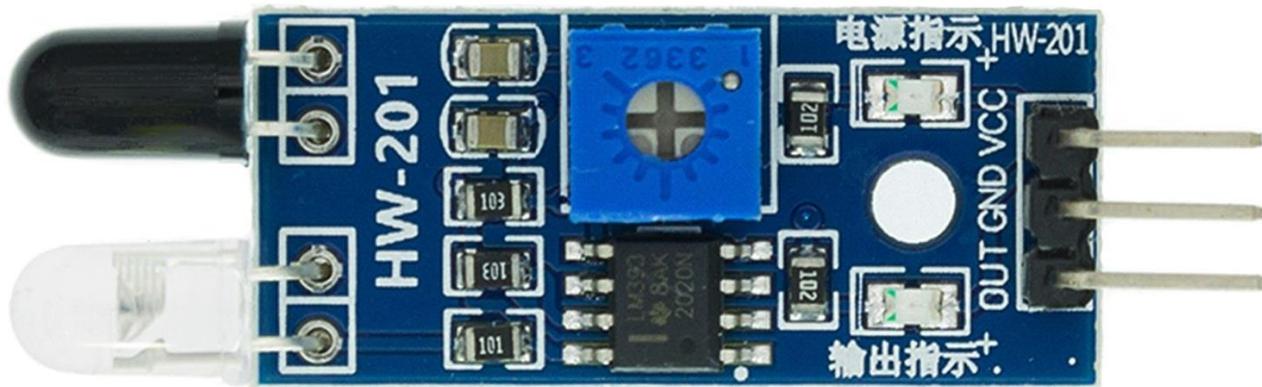


Рисунок 12– Вилка разъема YL-63

2.2 Ультразвуковые датчики

Ультразвуковые датчики широко используются в качестве датчиков приближения, для дистанционного обнаружении различных объектов и измерения расстояний. Как правило, датчики работают путем послыки короткого пакета ультразвуковых волн в направлении объекта обнаружения, который, отразившись от поверхности объекта, возвращается обратно. Затем, электронная схема производит расчет времени между моментом послыки сигнала и моментом приема отраженного эха. При этом расстояние является величиной, зависящей от скорости звука в окружающей среде. В настоящее время на рынке представлен широкий выбор ультразвуковых датчиков в различных конструктивных исполнениях, действующих в различных акустических частотах.

В данной главе описан измеритель расстояния на основе ультразвукового датчика типа HC-SR04, который показан на рисунке 13.



Рисунок 13 – Ультразвуковой датчик расстояния HC-SR04

Кроме излучателя и приемника, современный ультразвуковой датчик, используемый для измерения расстояния до различных объектов [24], должен иметь в своём составе электронную схему обработки сигнала и управления распределением временных интервалов и функционированием выходного каскада датчика. Блок-схема ультразвукового датчика показана на рисунке 14.

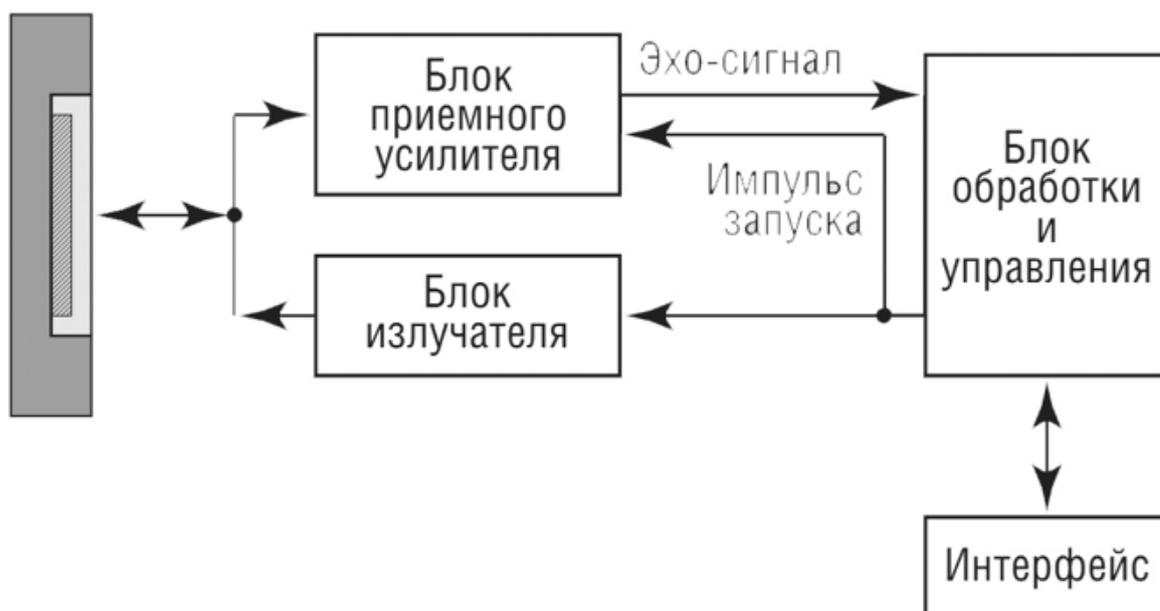


Рисунок 14 – Блок-схема ультразвукового датчика

Ультразвуковой дальномер HC-SR04 представляет «собой простое и дешевое средство для измерения расстояния, не превышающее 4 м [25]. Благодаря встроенной электронике дальномер позволяет легко реализовать программный алгоритм измерения расстояний.

Для подключения к схеме измерения датчик оснащен 4 выводами. Два из них служат для подключения питания. На вход Trig подается запускающий импульс, а с выхода Echo снимается сигнал, длительность которого пропорциональна измеренному расстоянию» [1]. Схема датчика показана на рисунке 15.

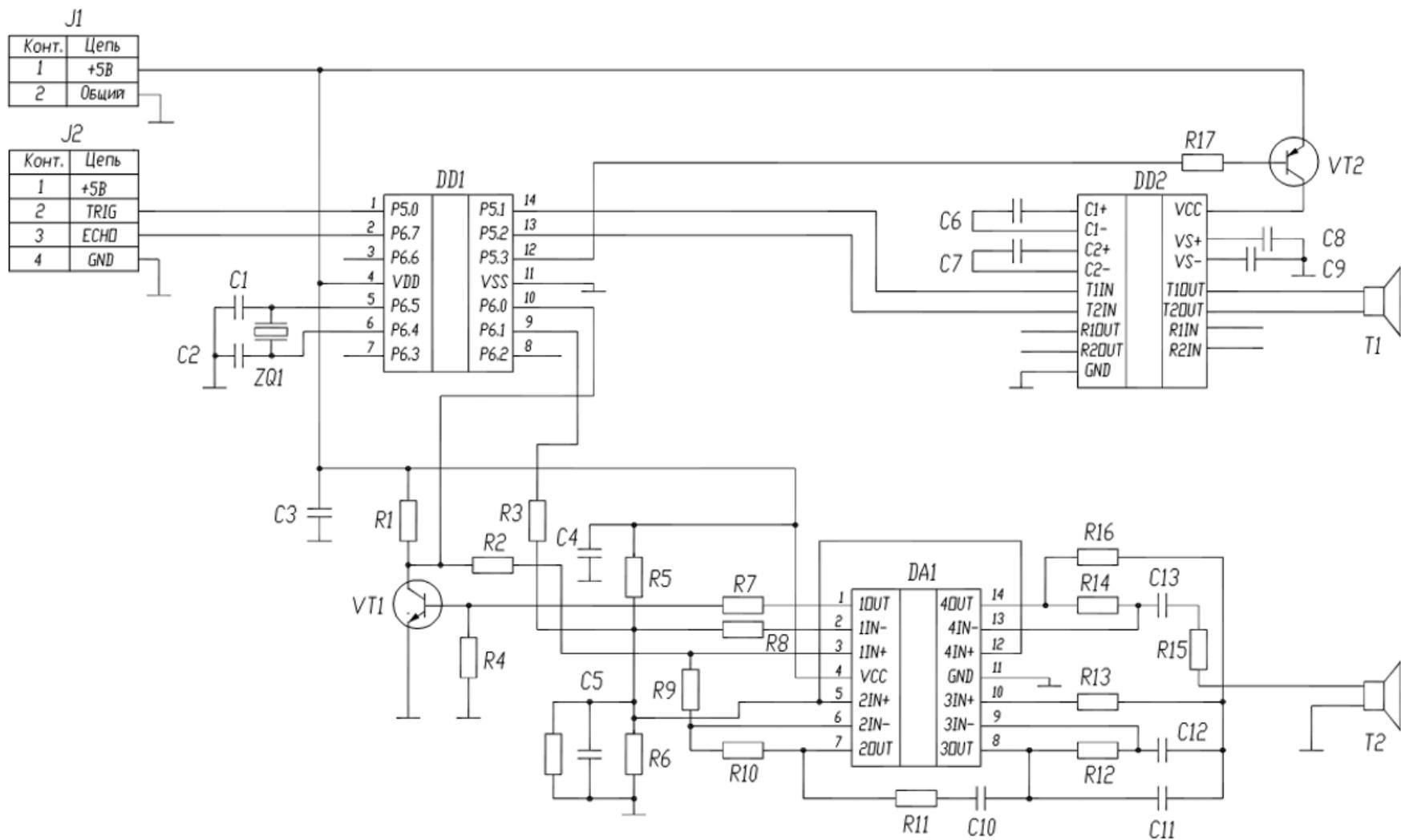


Рисунок 15 – Схема ультразвукового датчика HC-SR04

Основой его служит однократно программируемый микроконтроллер на микросхеме DD1. Программа, записанная в память микроконтроллера, не только формирует пачки ультразвуковых импульсов, но и реализует общий алгоритм работы всей схемы датчика.

«На микросхеме DD2 собран усилитель, к выходу которого подключен излучатель T1. Работой усилителя управляет транзистор VT2, который либо включает микросхему, либо выключает ее в зависимости от управляющего сигнала, поступающего с микроконтроллера.

На микросхеме DA1, в составе которой находятся 4 операционных усилителя, собран приемный тракт отраженного ультразвукового сигнала.»[1]

Алгоритм работы датчика показан на рисунке 16.: «для старта измерения, на вход Trig подается запускающий импульс длительностью 10 микросекунд; после обнаружения запускающего импульса, датчик излучает пачку из 8 импульсов с частотой 40 кГц; обнаружив отраженный сигнал, HC-SR04 устанавливает высокий уровень на выходе Echo.»[7] Длительность данного состояния в микросекундах будет пропорциональна измеренному расстоянию в метрах.

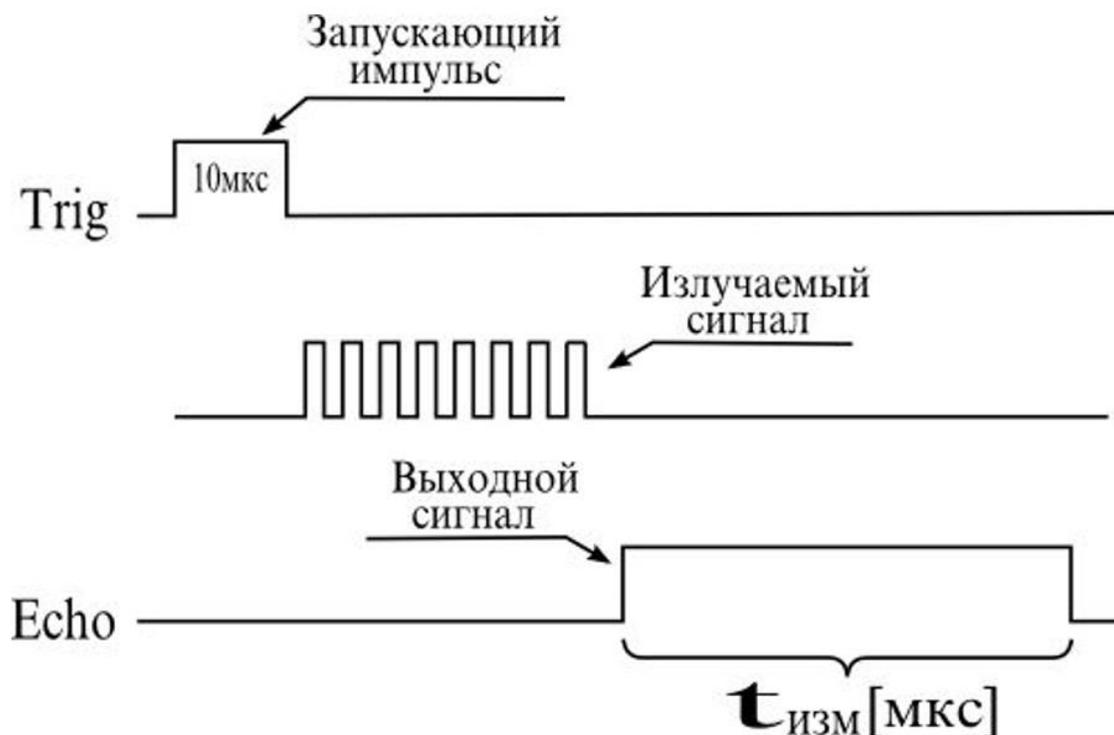


Рисунок 16 – Диаграмма работы HC-SR04

Управляющий микроконтроллер должен сформировать запускающий импульс и измерить значение эхо-сигнала. «Для этого используется таймер, встроенный в микроконтроллер и инкриминирующийся с частотой 1 МГц, (1 микросекунда). При обнаружении эхо-сигнала таймер запускается, а по отрицательному фронту останавливается. После этого читается значение счетчика» [7]. Таким образом, на выходе счетчика получается число пропорциональное расстоянию. Учитывая значения величин, используется 16 разрядный таймер.

2.3 Лидары

Лидар (Light Detection and Ranging) является технологией дистанционного зондирования, использующей свет в форме лазеров для измерения расстояний

до объектов. Благодаря своей способности создавать точные трехмерные модели окружающей среды, лидар нашел широкое применение в различных отраслях промышленности и робототехники. Преимущества лидара включают высокую точность, возможность создания трехмерных моделей окружающей среды и относительно быструю обработку данных. Эти характеристики делают лидар перспективным инструментом для применения в 3D печати, особенно для задач, требующих высокой точности и надежности.

Лидар может помочь в решении этой задачи, предоставляя точные данные о высоте объекта и позволяя быстро и точно определить истинный слой, на котором необходимо возобновить печать.

Лидар («обнаружение и определение дальности с помощью света») – «технология измерения расстояний путем излучения света (лазер) и замера времени возвращения этого отражённого света на ресивер» [1].

«Сканирующие лидары в системах машинного зрения формируют двумерную или трёхмерную картину окружающего пространства. Лидары особенно ценны в развивающихся сегодня стремительными темпами технологиях благодаря таким своим возможностям как построение карты высот местности, определение расстояний до объектов с высокой точностью, определение скорости движущегося объекта» [1].

Лидар (LIDAR) состоит из передатчика светового потока (обычно это лазер) и приемника [18]отраженного от препятствия света. Зная скорость распространения света и время распространения луча света до препятствия и обратно, можно определить расстояние до препятствия.

Одним из способов измерения времени пролета (time of flight, ToF) светового луча является использование импульсного лазера и непосредственное измерение затраченного времени. Но в данном случае электроника, измеряющее это время, должна уметь работать с пикосекундными интервалами времени, что делает эти устройства весьма дорогими.

Однако есть и другой, более дешевый и простой способ измерения времени пролета светового луча, который основан на измерении фазового сдвига отраженного света. Для подобных измерений используется коллимированный инфракрасный лазер [27].

Для поверхностей, «шероховатость (неравномерность) которых больше длины волны падающего света, возникает рассеянное (диффузное) отражение светового потока. В этом случае компоненты инфракрасного света возвращаются обратно практически параллельно излученному световому потоку.»[1] После этого датчик измеряет фазовый сдвиг [19] между излученным и принятыми сигналами. При этом справедливо выражение 1:

$$c = f \cdot \tau, \quad (1)$$

где c – скорость света,

f – модулирующая частота,

τ – длина волны.

Расстояние D' , пройденное излученным светом, определяется выражением 2:

$$D' = B + 2A = B + (\theta * \tau) / 2\pi, \quad (2)$$

где A – измеряемое расстояние,

B – расстояние от устройства измерения фазы.

θ – измеренный электроникой фазовый сдвиг между переданным и принятым лучами света

В этом случае необходимое расстояние между излучателем света и препятствием рассчитывается по формуле 3:

$$D = \tau \cdot \frac{\theta}{4\pi}. \quad (3)$$

Можно сказать, что расстояние обратно пропорционально квадрату амплитуды принятого сигнала. [20] То есть амплитуда принятого сигнала непосредственно влияет на точность датчика.

«VL53L0X – это миниатюрный высокоточный лазерный дальномер, обеспечивающий измерение расстояния до 2 м. Для измерения расстояния используется метод измерения времени (TOF, Time-of-Flight). TOF-дальномеры ещё называют время пролётными дальномерами. В датчике установлен миниатюрный лазер и фоточувствительная матрица диодов. Лазер короткими импульсами испускает свет, который отражаясь от препятствий попадает обратно и регистрируется на фоточувствительной матрице.» [28] Датчик измеряет время, за которое свет прошёл путь от излучателя до препятствия и обратно до датчика и на основе этого вычисляет расстояние до препятствия [29].

Датчик дальности представляет собой модуль лазерной дальности, разработанный и изготовленный на базе VL53L0X, который обеспечивает точные и воспроизводимые функции измерения на больших расстояниях. Благодаря встроенному ведущему массиву SPAD (однофотонный лавинный диод) и технологии FlightSense второго поколения он может достигать более высоких расстояний [30], более точных результатов измерений и более высокой устойчивости к окружающему свету.

TOF200C одновременно поддерживает режим последовательного порта, режим моделирования последовательного порта Modbus и режим I2C, который хорошо адаптирован к различным сценариям применения.

TOF200C имеет радиус действия до 2 м и может выбирать режимы высокой точности или измерения большого расстояния в соответствии с требованиями, что делает его более гибким.

VL6180X – это «времяпролётный» датчик расстояния, он показан на рисунке 17.



Рисунок 17- датчик расстояния VL6180X

Датчик содержит лазерный источник и соответствующий датчик. VL53L0X может определять «время полета» или время, которое потребовалось свету, чтобы вернуться обратно к датчику (рисунок 18). Поскольку он использует очень узкий источник света, он подходит для определения расстояния только до поверхности, находящейся непосредственно перед ним. У данного датчика «конус» зондирования очень узкий. В отличие от ИК-датчиков расстояния, которые пытаются измерить количество отраженного света, VL53L0X гораздо точнее и не имеет проблем с линейностью или «двойного изображения», когда невозможно определить, находится ли объект очень далеко или очень близко.

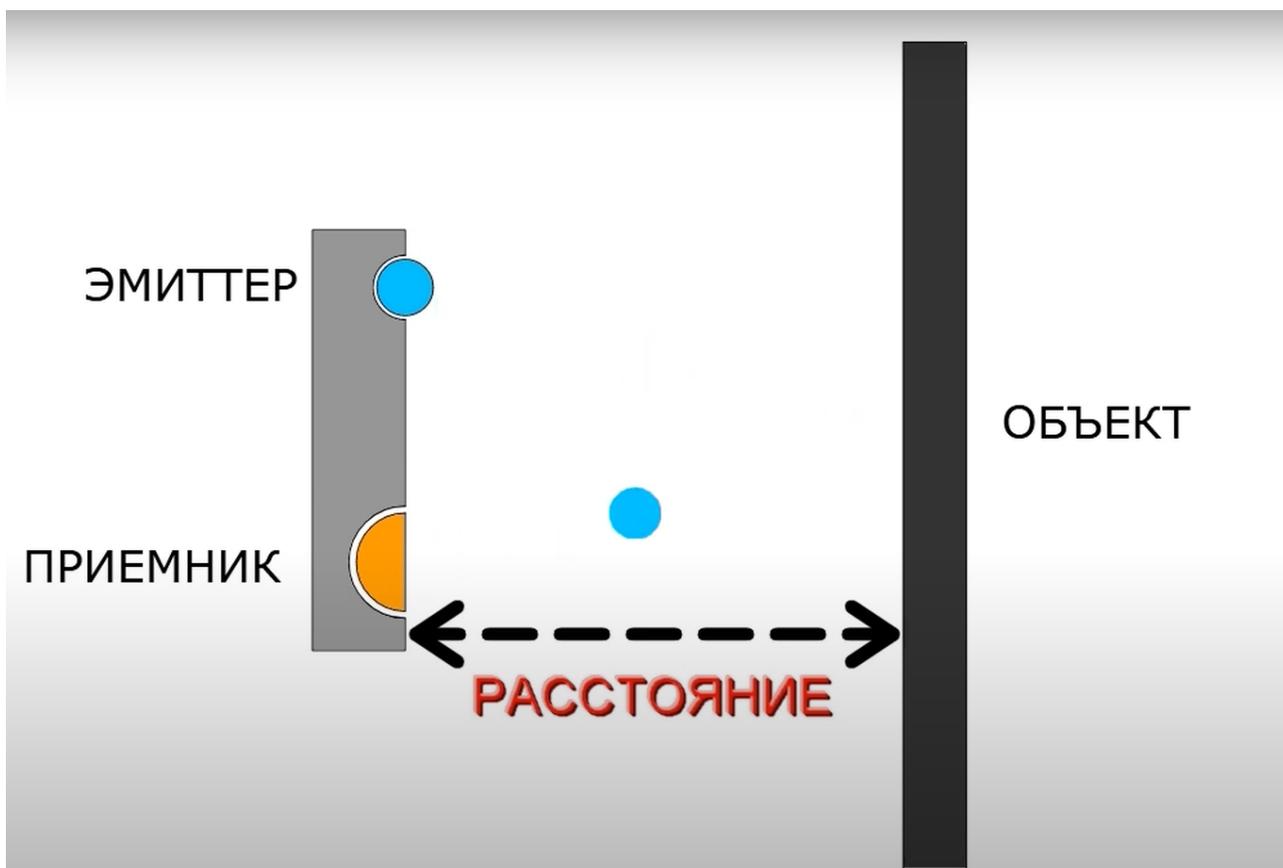


Рисунок 18 – принцип работы Лидара

Датчик небольшой и простой в использовании в любом робототехническом или интерактивном проекте. Поскольку ему требуется питание 2,8 В и логический выход, его можно без проблем использовать с любым силовым или логическим микроконтроллером 3-5 В.

Связь с датчиком осуществляется через I2C с помощью API, написанного ST, поэтому его легко и быстро можно подключить к микроконтроллеру.

Датчик может измерять расстояние примерно от 0,2 мм до 50 мм.

В режиме «дальнего радиуса действия» он может обнаруживать объекты на расстоянии от 1,5 до 2 метров на отражающей поверхности.

В зависимости от окружающего освещения и расстояния возможно получить точность определения дальности от 3 до 12 % – лучшее освещение и

блестящие поверхности дают лучшие результаты. Потребуется некоторые эксперименты, поскольку, если объект поглотит лазерный свет, будет невозможно получить хорошие показания.

VL53L0X – это датчик I2C[9]. Он использует два провода данных I2C, доступных на большинстве микроконтроллеров, и может использовать эти контакты для других датчиков, если у них нет конфликта адресов. Схема данного датчика изображена на рисунке 19.

Адрес I2C по умолчанию – 0x29. Изменить его можно, но только программно. Это означает, что необходимо подключить контакт SHUTDOWN и удерживать все датчики, кроме одного, в состоянии сброса, пока переконфигурируется один датчик за раз.

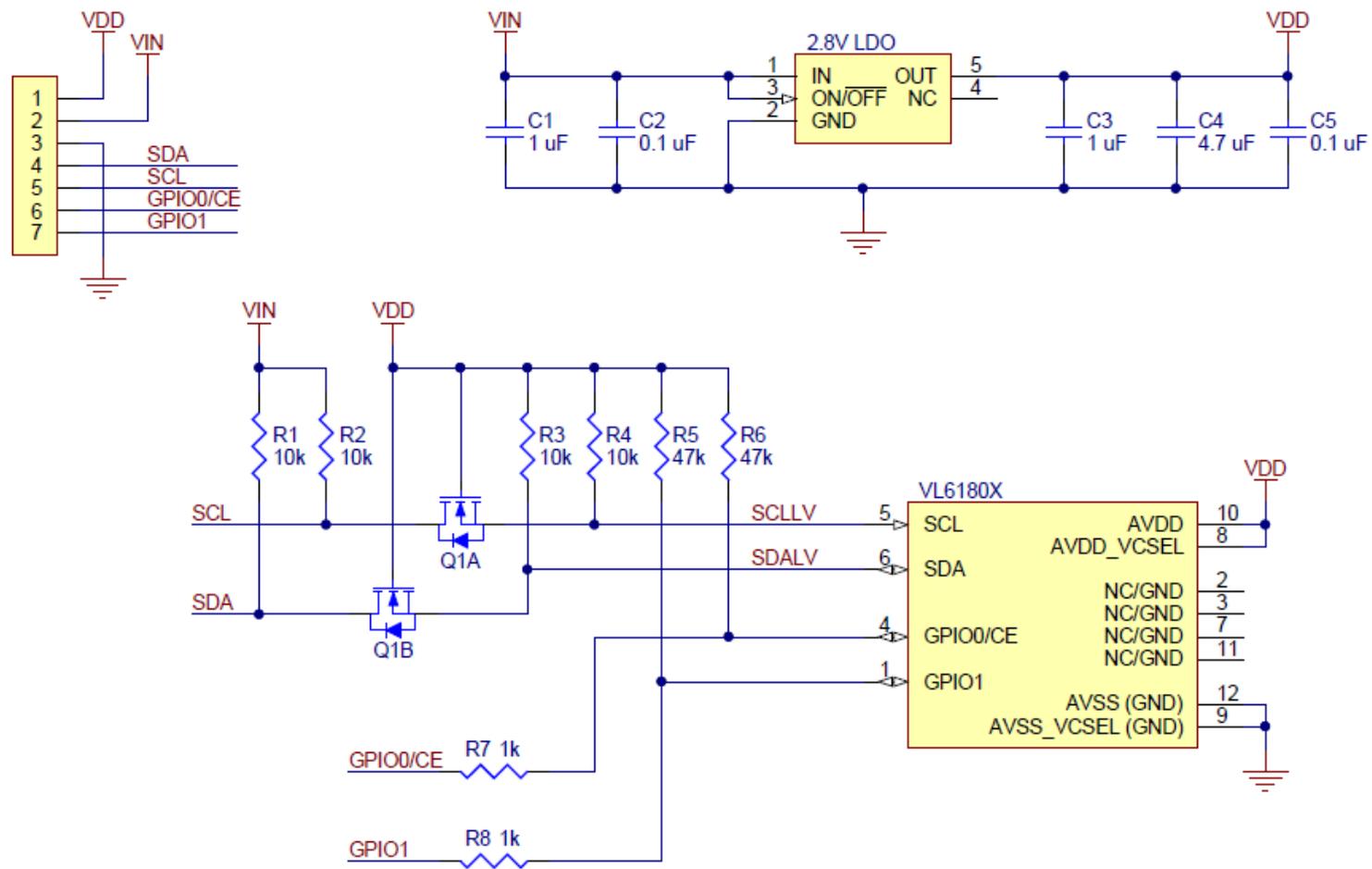


Рисунок 19 – Схема VL53L0X

Силовые контакты:

- Vin – это контакт питания. Поскольку чип использует напряжение 2,8 В [10] постоянного тока, на плату включен стабилизатор напряжения, который принимает напряжение 3–5 В постоянного тока и безопасно преобразует его. Для питания платы подается на нее то же напряжение, что и логический уровень микроконтроллера – например, для микросхемы с напряжением 5 В, такой как Arduino, используется 5 В;
- 2,8 – это выход 2,8 В от стабилизатора напряжения, при желании от него можно получить до 100мА;
- GND – общее заземление для питания и логики.

Логические контакты I2C:

- SCL – тактовый контакт I2C, подключается к тактовой линии I2C микроконтроллера;
- SDA – вывод данных I2C, подключается к линии передачи данных I2C микроконтроллера;
- STEMMA QT – эти разъемы позволяют подключаться к платам разработки с разъемами STEMMA QT.

Управляющие контакты:

- GPIO – это контакт, который используется датчиком для обозначения готовности данных. Это полезно при непрерывном зондировании. На этом выводе нет смещения уровня, что может не позволять прочитать напряжение логического уровня 2,8 В на микроконтроллере 5 В;
- SHDN – вывод отключения датчика. На этом выводе имеется диод, сдвигающий уровень, поэтому возможно использовать логику 3–5 В. Когда штифт опущен вниз, датчик переходит в режим отключения.

Преимущества лидара:

- легкий вес,
- однозначный прямой расчет дальности,

- без искажения движения
- возможность представлять объекты, которые наклонены к камере,
- нет необходимости в точных сканирующих механизмах,
- объединение 3D Flash LIDAR с 2D камерами (ЕО и IR) для создания 2D текстуры,
- возможность объединения нескольких камер 3D Flash LIDAR для создания полноценной трехмерной картины,
- меньше по размеру и легче, чем точечные системы сканирования[16],
- нет движущихся частей,
- низкое энергопотребление,
- способность «видеть» в затруднённых условиях (туман, дым, пар, дымка, дождь).

Подведем следующие выводы:

В ходе анализа датчиков расстояния VL53L0X, YL-63 и HC-SR04 выявлены их основные характеристики, преимущества и недостатки, что позволяет сделать выводы о наиболее подходящих областях их применения и эффективности.

VL53L0X от STMicroelectronics представляет собой лазерный датчик [13] расстояния, основанный на технологии Time-of-Flight (ToF). Его основные преимущества заключаются в высокой точности измерений, малом размере и низком энергопотреблении. Этот датчик способен измерять расстояние до 2 метров с точностью до миллиметров, что делает его идеальным для использования в мобильных устройствах, робототехнике и других областях, требующих высокоточных измерений на коротких дистанциях. Однако его стоимость выше по сравнению с другими датчиками, что следует учитывать при массовом производстве.

YL-63 – это инфракрасный датчик расстояния[12], который предлагает среднюю точность и надежность измерений. Этот датчик хорошо подходит для

простых приложений, таких как обнаружение препятствий и измерение расстояния в домашних условиях. Его преимущества: низкая стоимость и простота интеграции. Однако, по сравнению с лазерными и ультразвуковыми датчиками, YL-63 имеет ограниченную дальность измерения и может быть подвержен влиянию окружающего освещения, что может снизить его точность.

HC-SR04 – это ультразвуковой датчик расстояния, который широко используется благодаря своей доступности и надежности. Он способен измерять расстояние до 4 метров с достаточно хорошей точностью, что делает его подходящим для задач, таких как робототехника, системы безопасности и автоматизация. Преимуществами HC-SR04 являются его низкая стоимость и широкая распространенность. Однако ультразвуковые датчики могут сталкиваться с проблемами при работе с поверхностями, которые плохо отражают звук, а также в условиях высоких шумов.

Исходя из вышесказанного можно сделать вывод, что каждый из проанализированных датчиков имеет свои уникальные особенности и области применения. VL53L0X обладает высокой точностью и компактностью, что делает его подходящим датчиком для поставленной задачи, которая требует точные измерения, так как высота слоя при 3D-печати может достигать 0,1 мм. В свою очередь YL-63 является бюджетным решением для простых задач измерения расстояния, где не требуются высокая точность. HC-SR04 является универсальным и доступным датчиком для широкого круга применений, хотя его точность также может варьироваться в зависимости от условий окружающей среды и типов поверхностей, следовательно, этот датчик не подходит для решения задачи.

3 Разработка системы позиционирования

3.1 Сборка на макетной плате, подключение и исследование первого датчика

После анализа датчиков и их характеристик собрана экспериментальная установка на макетной плате для исследования датчиков расстояния. В первую очередь собрана схема для экспериментов с ультразвуковым датчиком расстояния. Схема подключения изображена на рисунке 20. В качестве процессора, выполняющего вычислительные действия, а выбрана плата NodeMCU, так как имеет возможность дистанционного подключения, что пригодится в дальнейших разработках устройства. А также подключен OLED дисплей для вывода информации о статусе сканирования поверхности рабочей плоскости.

Так же добавлена возможность срабатывания звукового сигнала посредством звукового излучения зуммера в случае, если обнаружена ошибка при печати. Данный сигнал дублируется срабатывание светодиода, который так же подключен к NodeMCU.

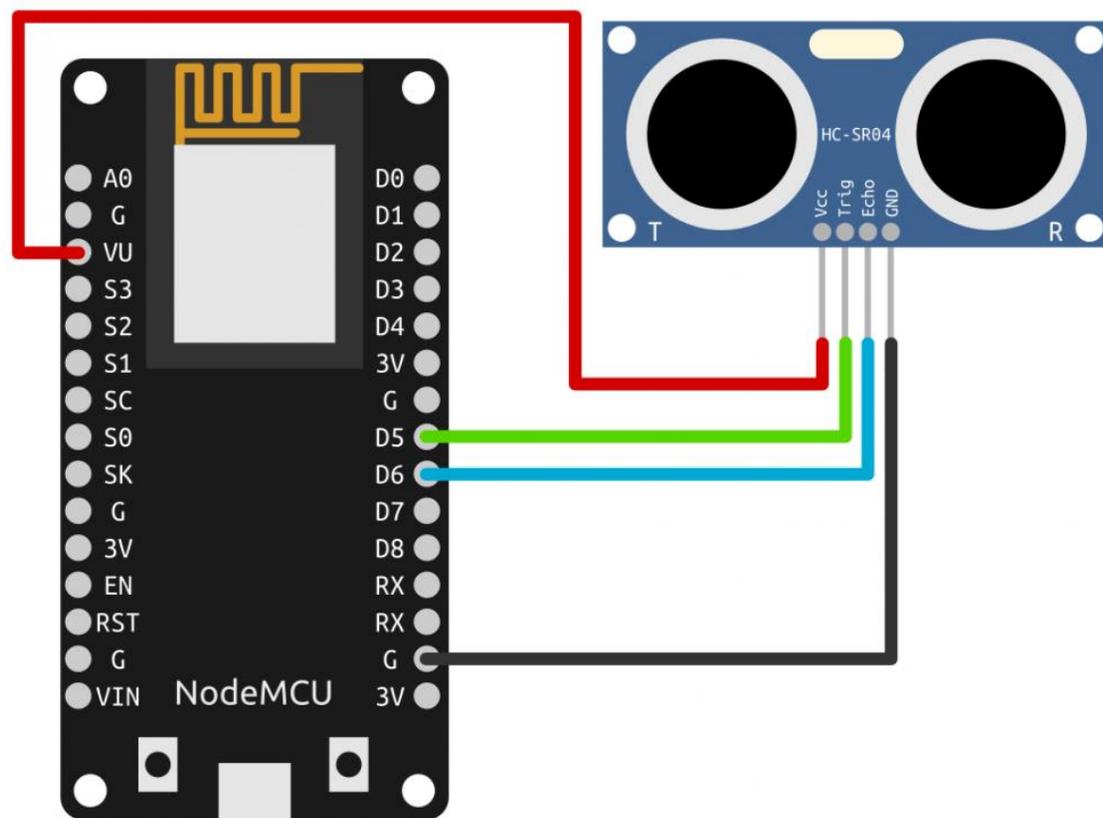


Рисунок 20 - Подключение датчика HC-SR04 к NodeMCU

Далее написана программа для считывания значения расстояния до объекта с датчика расстояния и вывода этой информации «монитор порта» программы. Для написания программы использована среда разработки Arduino IDE. Текст программы приложен ниже:

```
// Подключение библиотек
#include <ESP8266WiFi.h>

// Определение пинов для HC-SR04 и встроенного светодиода
const int trigPin = D1; // Пин для отправки сигнала (TRIG)
const int echoPin = D2; // Пин для приема сигнала (ECHO)
const int ledPin = D4; // Пин для встроенного светодиода

// Переменные для измерения расстояния
long duration;
float distance_cm; // Расстояние в сантиметрах с десятичными долями

void setup() {
    // Настройка пинов для HC-SR04 и светодиода
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    pinMode(ledPin, OUTPUT);

    // Настройка скорости последовательного порта
    Serial.begin(9600);

    // Настройка подключения к Wi-Fi (если необходимо)
    // WiFi.begin("SSID", "password");

    // while (WiFi.status() != WL_CONNECTED) {
    //     delay(1000);
    //     Serial.print(".");
    // }

    // Serial.println("WiFi connected");
}
```

```

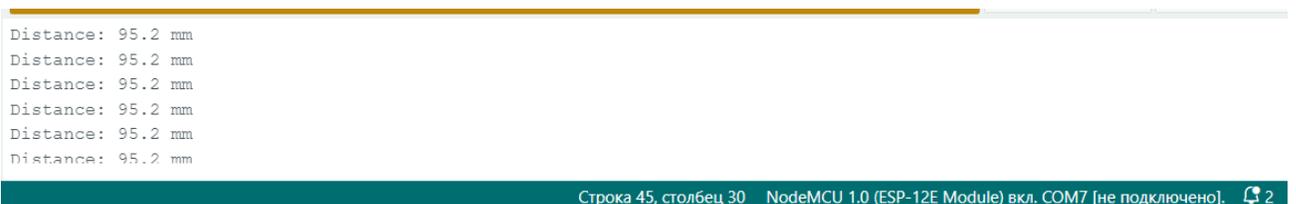
// Serial.println("IP address: ");
// Serial.println(WiFi.localIP());
}

void loop() {
// Отправляем импульс для запуска измерения
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
// Считываем время задержки до возврата эхо-сигнала
duration = pulseIn(echoPin, HIGH);
// Вычисляем расстояние в сантиметрах с десятичными долями
distance_cm = duration * 0.0343 / 2.0;
// Выводим расстояние в Serial Monitor
Serial.print("Distance: ");
Serial.print(distance_cm, 1); // Один знак после запятой для десятых долей
Serial.println(" cm");
// Моргание светодиодом в зависимости от расстояния
if (distance_cm < 10.0) {
    blinkLed(200); // Моргаем светодиодом с интервалом 200 мс
} else if (distance_cm < 20.0) {
    blinkLed(500); // Моргаем светодиодом с интервалом 500 мс
} else {
    digitalWrite(ledPin, LOW); // Выключаем светодиод
}
// Задержка между измерениями
delay(1000);

```

```
}  
// Функция для моргания светодиодом с указанным интервалом  
void blinkLed(int interval) {  
    digitalWrite(ledPin, HIGH); // Включаем светодиод  
    delay(interval);           // Ждем указанный интервал  
    digitalWrite(ledPin, LOW); // Выключаем светодиод  
    delay(interval);           // Ждем еще указанный интервал  
}
```

После подключения проведен эксперимент, который заключался в создании препятствия перед датчиком на расстоянии 95 мм путем установки напротив датчика белого листа бумаги. В ходе эксперимента получили данные, изображенные на рисунке 21, что подтвердило правильность написания кода и сборки схемы на макетной плате.

A screenshot of a serial monitor window. The main area shows six lines of text, each reading "Distance: 95.2 mm". At the bottom of the window, there is a status bar with the text "Строка 45, столбец 30 NodeMCU 1.0 (ESP-12E Module) вкл. COM7 [не подключено]. 2".

```
Distance: 95.2 mm  
Строка 45, столбец 30 NodeMCU 1.0 (ESP-12E Module) вкл. COM7 [не подключено]. 2
```

Рисунок 21 – Полученные данные при первом эксперименте

Далее был проведен эксперимент с измерением расстояния до объектов из различных материалов. Для усреднения значений было произведено 5 замеров. В качестве материалов были выбраны стекло и термоковрик для 3D-принтера с черной матовой поверхностью. Таки материалы были выбраны, так как это самые распространенные материалы, которыми накрывают рабочую область принтера.

В результате измерений пяти различных удалений от датчика получили данные, приведенные в таблице 2 и таблице 3 соответственно.

Таблица 2 – Эксперимент с термоковриком и датчиком HC-SR04 на столе

Истинное значение, см	1	3	5	7
Значение первого измерения, см	0,98	22,9	5,2	6,7
Значение второго измерения, см	10,16	3	4,9	7,2
Значение третьего измерения, см	10,19	32,3	5,2	6,6
Значение четвертого измерения, см	11,09	3,2	5,1	6,5
Значение пятого измерения, см	00,98	3,2	4,8	7,1

Таблица 3 – Эксперимент со стеклом и датчиком HC-SR04 на столе

Истинное значение, см	1	3	5	7
Значение первого измерения, см	1,39	2,93	4,84	6,58
Значение второго измерения, см	1,11	3,64	4,63	6,21
Значение третьего измерения, см	0,94	3,24	4,20	6,86
Значение четвертого измерения, см	0,82	2,56	5,04	7,10
Значение пятого измерения, см	1,26	3,03	5,37	7,71

Проанализировав полученные данные, получим среднее значение измерений для каждой дистанции и каждого материала, которые сведем в таблицу 4.

Таблица 4 – Усредненные значения измерений датчика HC-SR04 на столе

Истинное значение, см	1	3	5	7
Среднее показание измерения расстояния до термоковрика, см	1,08	3,12	5,24	6,28
Среднее показание измерения расстояния до стекла, см	1,10	3,15	4,72	5,61

Исходя из этих данных, построим график погрешности измерений, изображенный на рисунке 22.

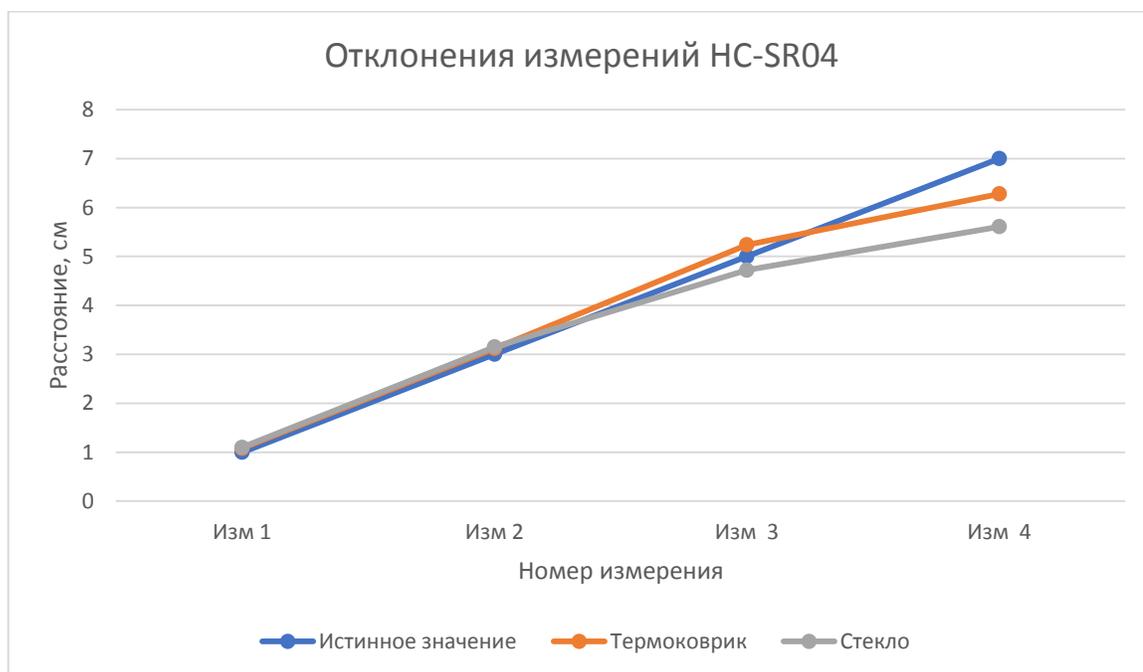


Рисунок 22 – Отклонения измерений HC-SR04 на столе

Далее проведен эксперимент с теми же материалами, но уже расположенными на рабочей поверхности 3D-принтера. Датчики также были установлены на 3D-принтер вблизи с печатающей головкой. Также были включены мотор экструдера, обдув сопла и модели. Это было сделано для достижения условий максимально приближенным к рабочим для проверки датчика в этом режиме. В ходе эксперимента были получены данные, отображенные в таблице 5 и таблице 6 соответственно.

Таблица 5 – Эксперимент с термоковриком и датчиком HC-SR04 в составе принтера

Истинное значение, см	1	3	5	7
Значение первого измерения, см	0,86	3,56	5,65	7,92
Значение второго измерения, см	1,48	2,80	4,40	6,14
Значение третьего измерения, см	0,95	3,68	4,27	6,86
Значение четвертого измерения, см	0,74	2,96	5,51	7,6
Значение пятого измерения, см	1,52	2,55	5,79	6,44

Таблица 6 - Эксперимент со стеклом и датчиком HC-SR04 в составе принтера

Истинное значение, см	1	3	5	7
Значение первого измерения, см	0,93	3,27	4,19	9,92
Значение второго измерения, см	1,08	2,55	5,31	4,02
Значение третьего измерения, см	0,69	3,29	3,08	4,14
Значение четвертого измерения, см	1,25	3,4	5,79	5,84
Значение пятого измерения, см	0,94	3,29	3,55	5,58

Также, проанализировав данные, получены и сведены в таблицу 7 следующие усредненные значения замеров.

Таблица 7 – Усредненные значения измерений датчика HC-SR04 на столе

Истинное значение, см	1	3	5	7
Среднее показание измерения расстояния до термоковрика, см	1,11	3,22	5,32	5,99
Среднее показание измерения расстояния до стекла, см	0,78	3,26	4,38	5,9

Ниже, на рисунке 23 приведен график, полученный из данных таблицы 7.

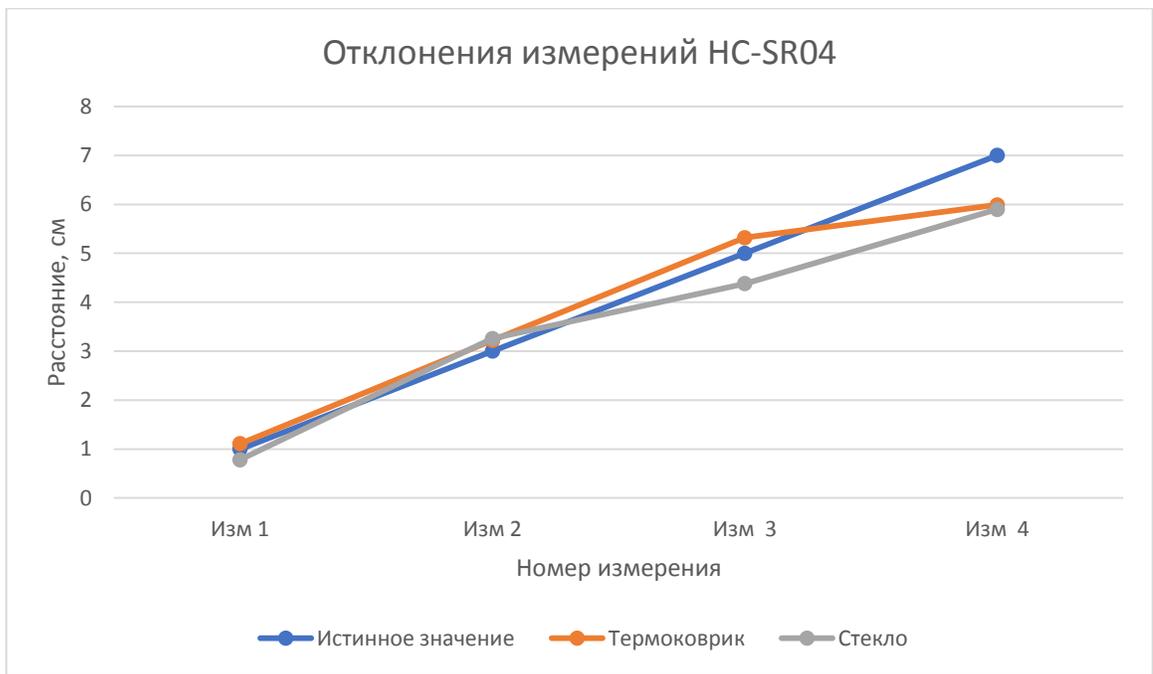


Рисунок 23 – Отклонения измерений HC-SR04 на принтере

В ходе данного эксперимента видна прямая зависимость наличия посторонних шумов и точности измерения расстояния датчиком HC-SR04.

3.2 Подключение и исследование второго датчика

На рисунке 24 показано, как подключен лазерный датчик расстояния TOF200к NodeMCU.

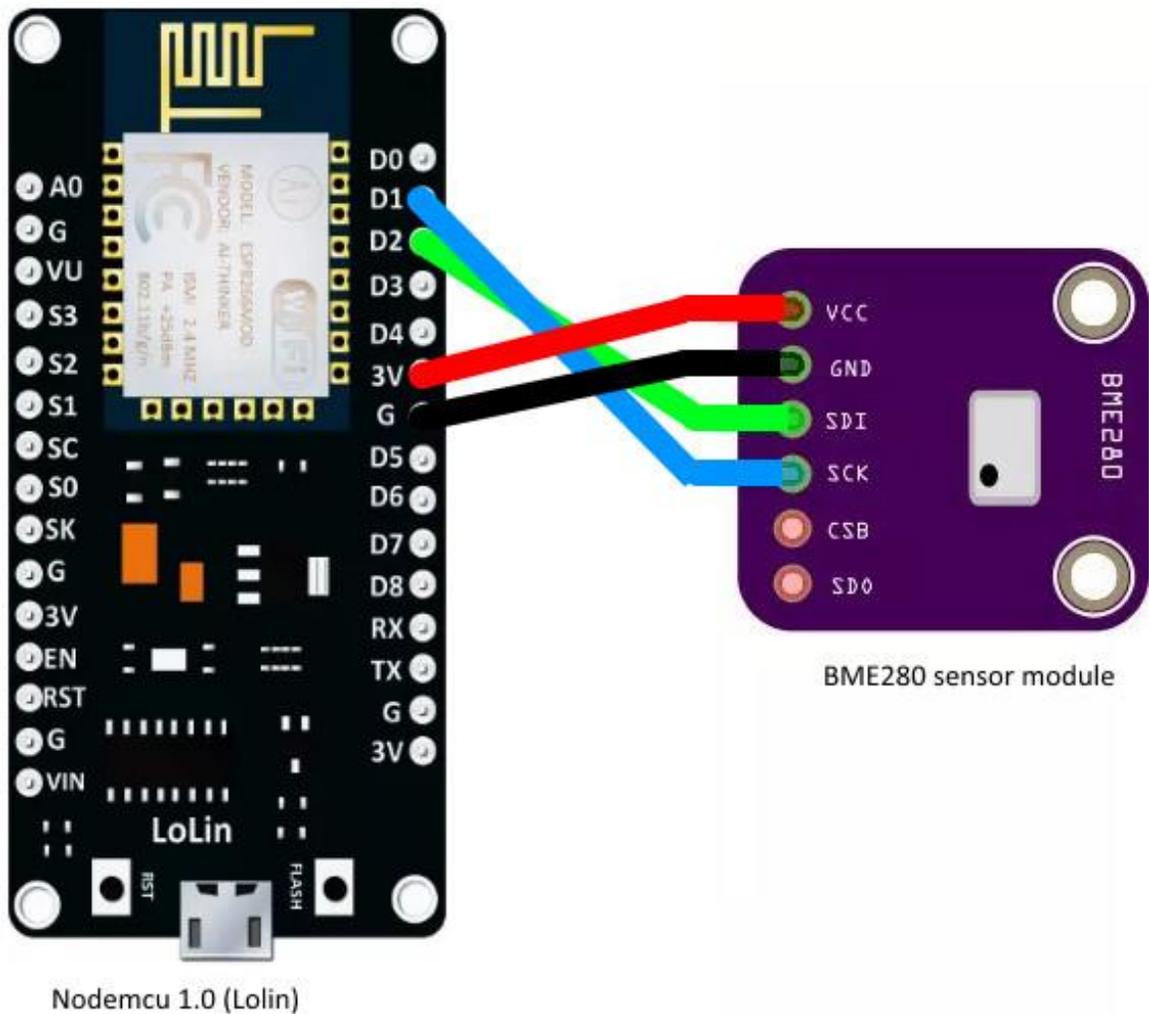


Рисунок 24 – Подключение датчика TOF200 к Arduino Nano

Написана подобная программа для измерения расстояния до объекта, но использовались уже другие библиотеки. Код программы для экспериментов с датчиком TOF200, листинг которой приведен ниже.

```

#include <Wire.h>
#include <Adafruit_VL53L0X.h>
Adafruit_VL53L0X distanceSensor = Adafruit_VL53L0X();
void setup() {
  Serial.begin(115200);
  // Инициализация I2C
  Wire.begin();
  if (!distanceSensor.begin()) {
    Serial.println(F("Не удалось найти датчик VL53L0X"));
    while (1);
  }
  Serial.println(F("Датчик VL53L0X инициализирован"));
}
void loop() {
  VL53L0X_RangingMeasurementData_t measure;
  distanceSensor.rangingTest(&measure, false);
  if (measure.RangeStatus != 4) { // Если не ошибка
    Serial.print(F("Расстояние: "));
    Serial.print(measure.RangeMilliMeter);
    Serial.println(F(" мм"));
  } else {
    Serial.println(F("Ошибка измерения"));
  }
  delay(1000); }

```

Убедившись в правильности подключения и написания программы путем выставления препятствия в виде белого листа бумаги на расстоянии 95 мм. произведены идентичные тесты, как и с ультразвуковым датчиком расстояния, что показано на рисунке 25.

```
Distance: 95.2 mm
Distance: 98.6 mm
Distance: 112.5 mm
Distance: 98.6 mm
Distance: 95.4 mm
Distance: 95.2 mm
```

Строка 45, столбец 30 NodeMCU 1.0

Рисунок 25 – Проверка корректности сборки схемы с датчиком лидаром

В первую очередь были поставлены эксперименты с измерением расстояния до объектов на столе. Результаты измерений продемонстрированы в таблице 8 и таблице 9 соответственно.

Таблица 8 – Эксперимент с термоковриком и датчиком TOF200 на столе

Истинное значение, см	1	3	5	7
Значение первого измерения, см	1,01	2,86	5,04	7,1
Значение второго измерения, см	1,08	2,85	4,95	6,97
Значение третьего измерения, см	0,99	3,09	5	7,06
Значение четвертого измерения, см	0,96	2,89	4,92	6,95
Значение пятого измерения, см	1,1	3,14	4,87	7,04

Таблица 9 – Эксперимент со стеклом и датчиком TOF200 на столе

1	2	3	4	5
Истинное значение, см	1	3	5	7

Продолжение таблицы 9

1	2	3	4	5
Значение первого измерения, см	1,09	2,83	5,17	6,98
Значение второго измерения, см	0,98	3,11	5,04	6,84
Значение третьего измерения, см	0,93	3,02	5,13	7,07
Значение четвертого измерения, см	0,91	3,05	5,15	7,00
Значение пятого измерения, см	1,08	3,02	4,86	6,97

Проведя анализ, основываясь на данных полученных ранее, получены следующие средние значения измерений, представленные в таблице 10 ниже.

Таблица 10 – Усредненные значения измерений датчика TOF200 на столе

Истинное значение, см	1	3	5	7
Среднее показание измерения расстояния до термоковрика, см	1,03	2,97	4,96	7,02
Среднее показание измерения расстояния до стекла, см	1	3,01	5,07	6,97

Исходя их данных таблицы 10, построен график, представленный на рисунке 26.



Рисунок 26 – Отклонения измерений TOF200 на столе

На данном графике видно, что у лидара отклонения от истинных значений минимальное.

Далее проведены эксперименты с установкой лазерного датчика на каретку непосредственно рядом с печатающей головкой. Измерения, которые были получены в ходе данных экспериментов приведены ниже в таблице 11 и таблице 12.

Таблица 11 – Эксперимент с термоковриком и датчиком TOF200 в составе принтера

Истинное значение, см	1	3	5	7
Значение первого измерения, см	0,9	2,86	4,81	7,2
Значение второго измерения, см	1,03	2,91	5,02	6,85
Значение третьего измерения, см	0,93	2,91	4,80	7,14
Значение четвертого измерения, см	0,95	2,84	5,08	7,1
Значение пятого измерения, см	1,06	3	5,09	6,97

Таблица 12 – Эксперимент со стеклом и датчиком TOF200 в составе принтера

Истинное значение, см	1	3	5	7
Значение первого измерения, см	0,97	3,06	5,11	6,95
Значение второго измерения, см	1,09	2,89	5,06	6,91
Значение третьего измерения, см	0,87	2,98	4,99	6,83
Значение четвертого измерения, см	0,93	3,1	5,06	6,97
Значение пятого измерения, см	1	2,94	4,84	7,06

В таблице 13 рассчитаны средние значения измерений с каждой поверхностью

Таблица 13 – Усредненные значения измерений датчика HC-SR04 на столе

Истинное значение, см	1	3	5	7
Среднее показание измерения расстояния до термоковрика, см	0,97	2,9	4,96	7,05
Среднее показание измерения расстояния до стекла, см	0,97	2,99	5,01	6,94

Ниже, на рисунке 27 приведен график, полученный из данных таблицы 13.

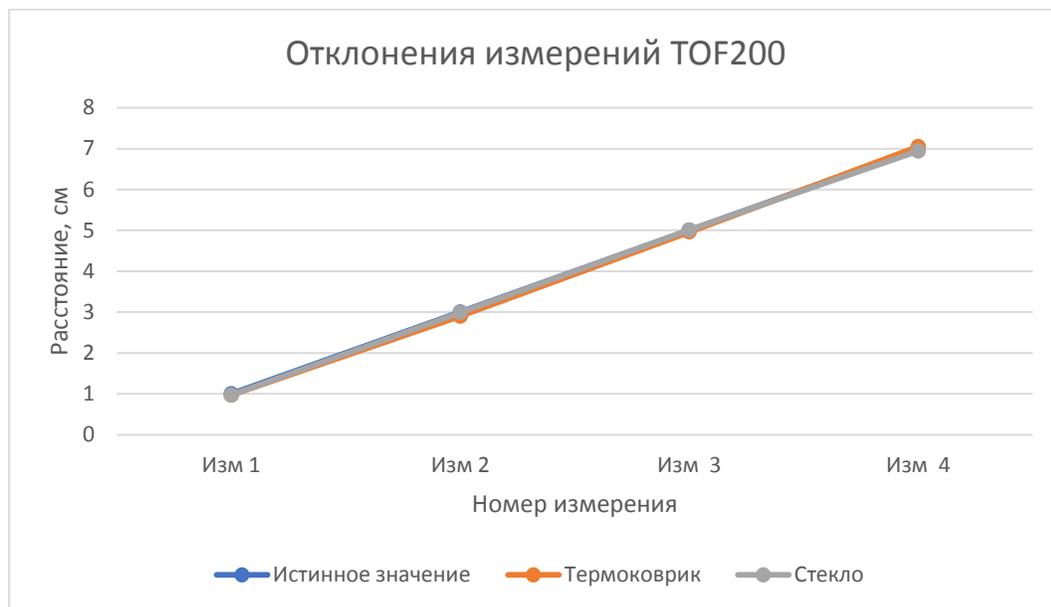


Рисунок 27 – Отклонения измерений TOF200 в составе принтера

Исходя из данных, полученных в ходе всех экспериментов, можно сделать вывод, что при неподвижном состоянии рабочей области измерения датчиком TOF200 и HC-SR04 достаточно близки по своему значению. Однако при условиях максимально похожих на рабочих, лидар показывает значения намного более приближенные, в отличии от HC-SR04. Это связано с тем, что принцип работы HC-SR04 основан на ультразвуковых волнах. А так как при работе 3D-принтер излучает множество посторонних шумов, то ультразвуковой датчик ловит помехи в сигнале, вследствие чего выдает некорректный результат.

Таким образом для будущего устройства был выбран лазерный датчик расстояния TOF200, так как его погрешность в измерениях в 1,24% позволяет производить достаточно точные измерения (критической погрешностью является ошибка в измерениях на более чем 0,15 мм, так как высота слоя печати составляет 0,3 мм).

4 Создание устройства сканирования рабочей поверхности 3D-принтера

Далее произведена разработка устройства для установки на 3D-принтер для определения положения печатающей головки после внезапной остановки печати. В данной работе для создания такого устройства выбран лидар для определения расстояния до объекта, а в качестве управляющего устройства выбран микроконтроллер NodeMCU. В ходе анализа лидара, описанного в предыдущей главе, выбрана модель TOF200C.

4.1 Подключение компонентов и написание программы

Для создания схемы подключения устройства использованна программа TinkerCAD. Схема подключения всех компонентов в данной программе приведена на рисунке 28.

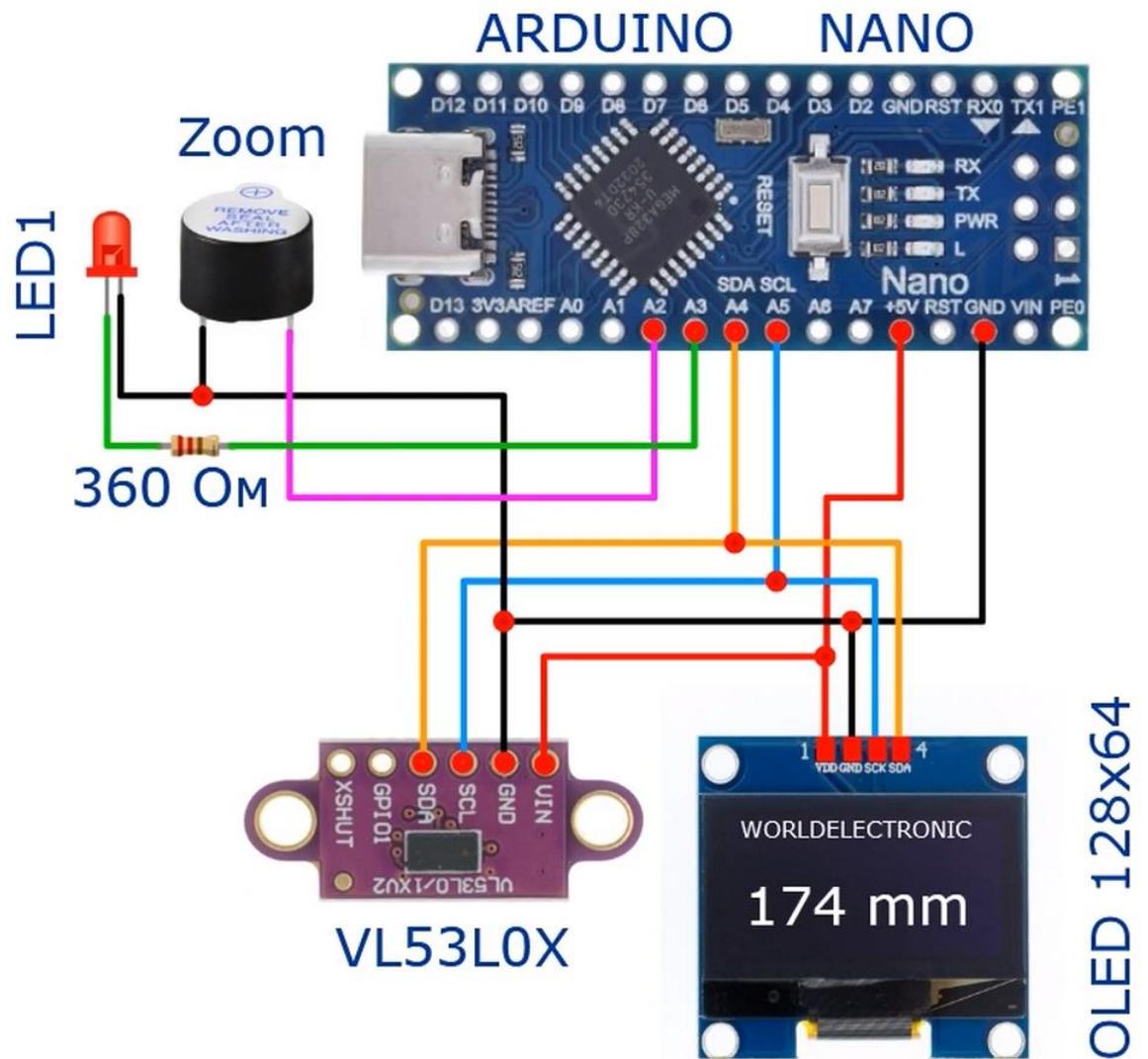


Рисунок 28 – Схема подключения устройства

Составление блок-схемы программы

Для составления программы в текстовом виде потребовалось ее визуализировать в графическом варианте.

Для создания блок-схемы программы использован сайт programforyou.ru [11]. Данный ресурс позволяет составить в кратчайшие сроки блок-схему, после чего эту схему можно экспортировать в нескольких форматах: json, png и zip. Для экспорта схемы выбран формат json, так как с его помощью возможно

дальнейшее редактирование файла.

После определения среды разработки графической части программы начато ее построение.

Блок-схема программы, которая продемонстрирована на рисунке 29, описывает следующий алгоритм работы: после включения 3D-принтера пользователь параллельно задает в настройках программы записанной на NodeMCU высоту каждого печатаемого слоя (значения предоставленные для выбора 0,2-0,5 мм с шагом в 0,05 мм). Далее устройство 3D-принтера начинает печать. Посредством установленного лидара в непосредственной близости к соплу подачи филамента программа производит замер расстояния от наименьшей до наивысшей точки. Наивысшей точке соответствует положение напечатанной модели по оси Z, а наименьшей расстояние до стола принтера соответственно. Данный замер производится каждые 5 напечатанных слоев. Для произведения замера каждые 5 слоев принтер останавливает печати и за счет работы шаговых двигателей, а также смещающейся рамки производит сканирование всей рабочей поверхности. В случае обнаружения что минимальное расстояние превышает заданное ранее значение высоты слоя программа сигнализирует об ошибке звуковым и световым сигналом. Далее программа отправляет пользователю на телефон сообщение об ошибке с номером слоя, на котором остановлена печать. Также номер этого слоя выводится на ЖК-дисплей. Расчет этого слоя производится следующим образом: из результата измерения максимального значения (расстояние до стола) вычитается минимальное значение (расстояние до модели), тем самым получая фактическую высоту напечатанной модели. Далее полученное значение делится на ранее заданную высоту слоя и таким образом выводится номер слоя остановки печати. Также программа анализирует ранее загруженный в память микроконтроллера документ, содержащий информацию с G-кодом, после чего производит его редактирование путем удаления строчек отвечающих за печать

уже произведенных слоев.

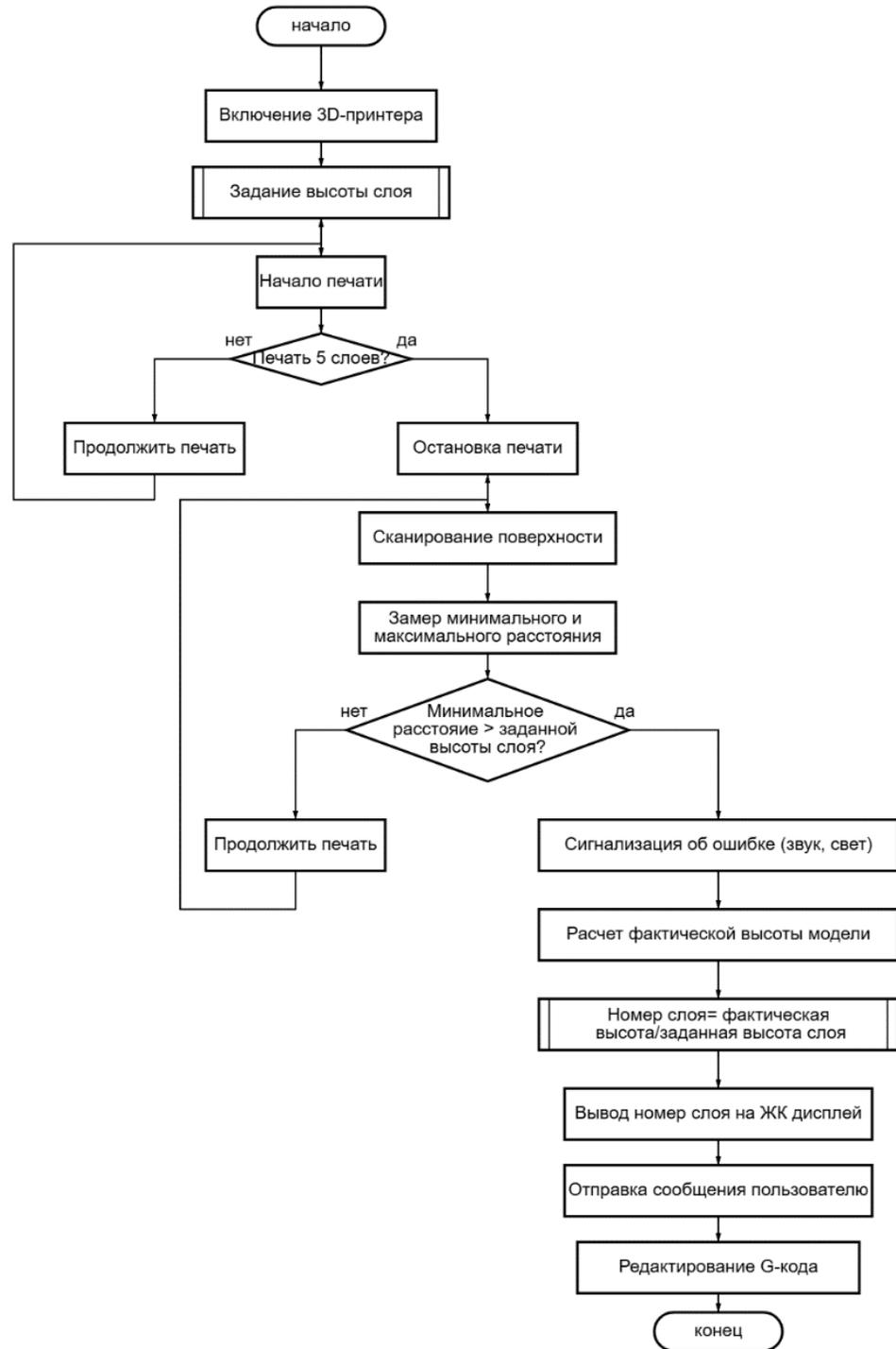


Рисунок 29 – Блок-схема программы

G-код и его редактирование

В 3D-принтере для управления процессом печати используется особое программное обеспечение. Процесс управления моторами, экструдером, хотэндом и так далее происходит посредством G-кода.

G-код – это тип языка программирования, используемый в области числового программного управления (ЧПУ) и 3D-печати для управления движением станков. Он написан на программном обеспечении автоматизированного производства (САМ) и предоставляет инструкции по автоматизации различных функций и инструментов станка.

G-код означает геометрический код. G-коды также известны как подготовительные коды для станков с ЧПУ.

Инструкции, предоставляемые G-кодами, сообщают станку, как двигаться в декартовой системе координат (X, Y, Z).

В дополнение к инструкциям по местоположению G-код также предоставляет множество других входных данных, таких как скорость и угол оси вращения, смещение длины инструмента, начальная точка, точка остановки, скорость подачи, время ожидания и т. д.

G-коды работают в тандеме с M-кодами. M-коды обозначают «машинные коды» или разные коды. M-коды содержат инструкции по различным функциям машины, не связанным с движениями.

Примером M-кода является M0, что означает завершение программы.

G-код представляет собой комбинацию алфавита и цифры. Номер может состоять из нескольких цифр. Расположение пробела между алфавитом и цифрой зависит от конкретного станка с ЧПУ.

Некоторые примеры распространенных команд G-кода: «G00», «F10», «M03» и т. д. Команды не обязательно начинаются с буквы «G». Однако «G» – наиболее распространенная буква, встречающаяся в инструкциях G-кода.

Каждая строка языка программирования G-кода может содержать несколько наборов инструкций. Это также известно как блок G-кода. Станок с ЧПУ считывает и выполняет инструкции в определенной последовательности слева направо и сверху вниз. Примеры распространенных алфавитов, используемых в G-коде:

- G: Общие движения,
- F: Скорость подачи,
- T: Смена филамента
- S: Скорость подачи филамента
- X, Y и Z: три линейные оси в декартовой системе координат,
- A, B и C: оси углового вращения вокруг X, Y и Z.

Для редактирования G-кода с помощью микроконтроллера NodeMCU следует использовать определенные библиотеки и методы.

Для начала требуется предварительно загрузить в память микроконтроллера файл с G-кодом. Это можно сделать с использованием библиотеки 'Serial'. Также возможна загрузка информации на SD-карту используя плату расширения «MH-SD Card Module». На рисунке 30 приведен фрагмент кода для инициализации SD-карты.

```
if (!SD.begin(chipSelect)) {  
  Serial.println("SD card initialization failed!");  
  return;  
}  
Serial.println("SD card initialized.");
```

Рисунок 30 – Инициализация SD-карты

Далее следует открыть файл с помощью команды ‘SD.open’, а также произвести чтение и обработку строк с использованием команды ‘readStringUntil’. Фрагменты кода, отвечающие за эти действия изображены на рисунках 31 и 32 соответственно.

```
File originalFile = SD.open("/gcode.txt", FILE_READ);  
File newFile = SD.open("/new_gcode.txt", FILE_WRITE);
```

Рисунок 31 – Открытие файла

```
while (originalFile.available()) {  
    String line = originalFile.readStringUntil('\n');  
    if (!shouldDeleteLine(line)) {  
        newFile.println(line);  
    }  
}
```

Рисунок 32 – Чтение и обработка строк файла

Выполнив команды описанные выше, приступили к написанию последней строчки кода, выполняющую редактирование файла. С помощью функции ‘shouldDeleteLine’ и функции ‘indexOf’ произвели удаление выбранных строчек G-кода, что показано на рисунке 33.

```

bool shouldDeleteLine(String line) {
    if (line.indexOf("LAYER:0") >= 0) {
        return true;
    }
    return false;
}

```

Рисунок 33 – Удаление выбранной строки кода

Таким образом получили следующий код для микроконтроллера:

```

#include <SPI.h>
#include <SD.h>
const int chipSelect = 4; // Пин для CS на SD-карте
void setup() {
    Serial.begin(9600);
    if (!SD.begin(chipSelect)) {
        Serial.println("SD card initialization failed!");
        return;}
    Serial.println("SD card initialized.");
    String gcodeFile = "/gcode.txt"; // Имя файла G-кода
    String newGcodeFile = "/new_gcode.txt"; // Имя нового файла G-кода
    if (!SD.exists(gcodeFile)) {
        Serial.println("G-code file not found!");
        return;}
    File originalFile = SD.open(gcodeFile, FILE_READ);
    File newFile = SD.open(newGcodeFile, FILE_WRITE);

```

```

if (originalFile && newFile) {
    while (originalFile.available()) {
        String line = originalFile.readStringUntil('\n');
        // Пропустить строки, соответствующие напечатанным слоям
        if (!shouldDeleteLine(line)) {
            newFile.println(line);} }
    originalFile.close();
    newFile.close();
    Serial.println("G-code file has been edited.");
} else {
    Serial.println("Error opening files!");} }
bool shouldDeleteLine(String line) {
    // Пример условия удаления строки, например, если строка содержит
"LAYER:0"
    if (line.indexOf("LAYER:0") >= 0) {
        return true;}
    return false;}
void loop() {
    // Пусто}

```

Отправка сообщений с помощью IFTTT

Для своевременного информирования оператора об ошибке во время печати предложена система удаленного оповещения о неполадках с помощью IFTTT (рисунок 34).

Чтобы создать данную систему оповещения в первую очередь потребовалось создать учетную запись на IFTTT [30].

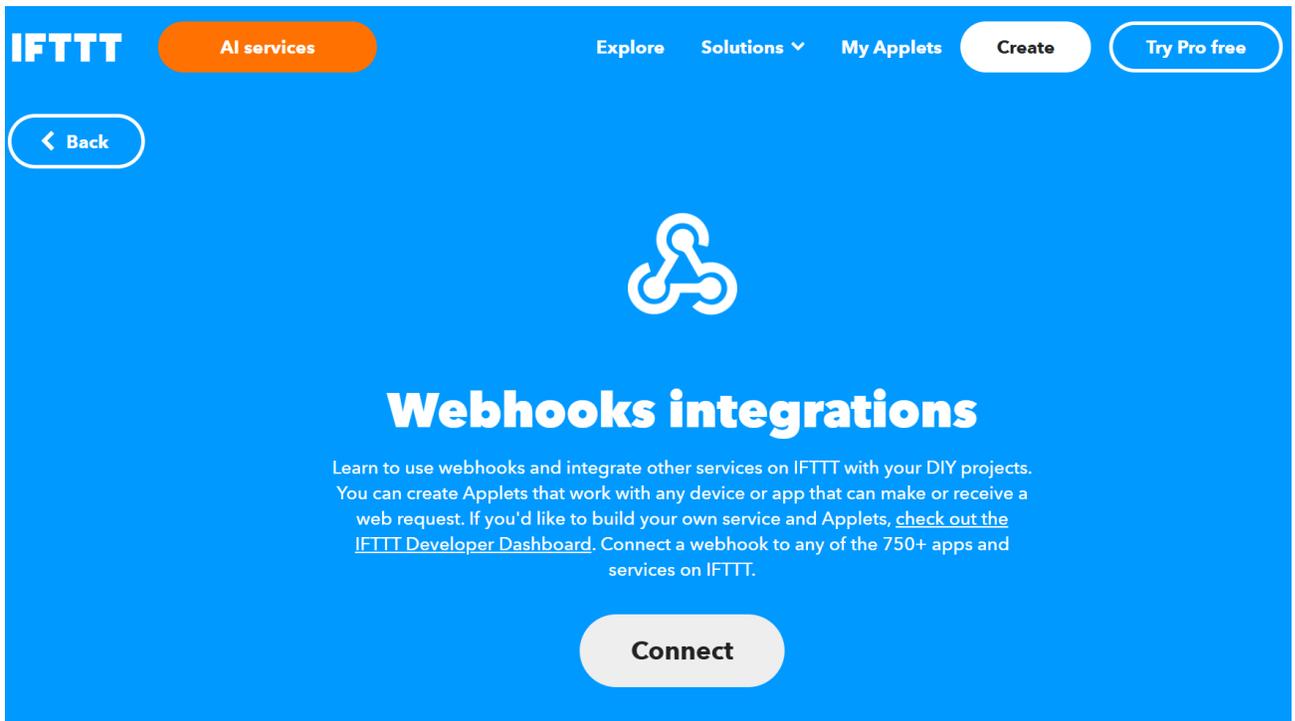


Рисунок 34 – Сервис IFTTT

Далее создано условие и действие для отправки уведомления, это показывает рисунок 35.

You're using 0 of 2 Applets

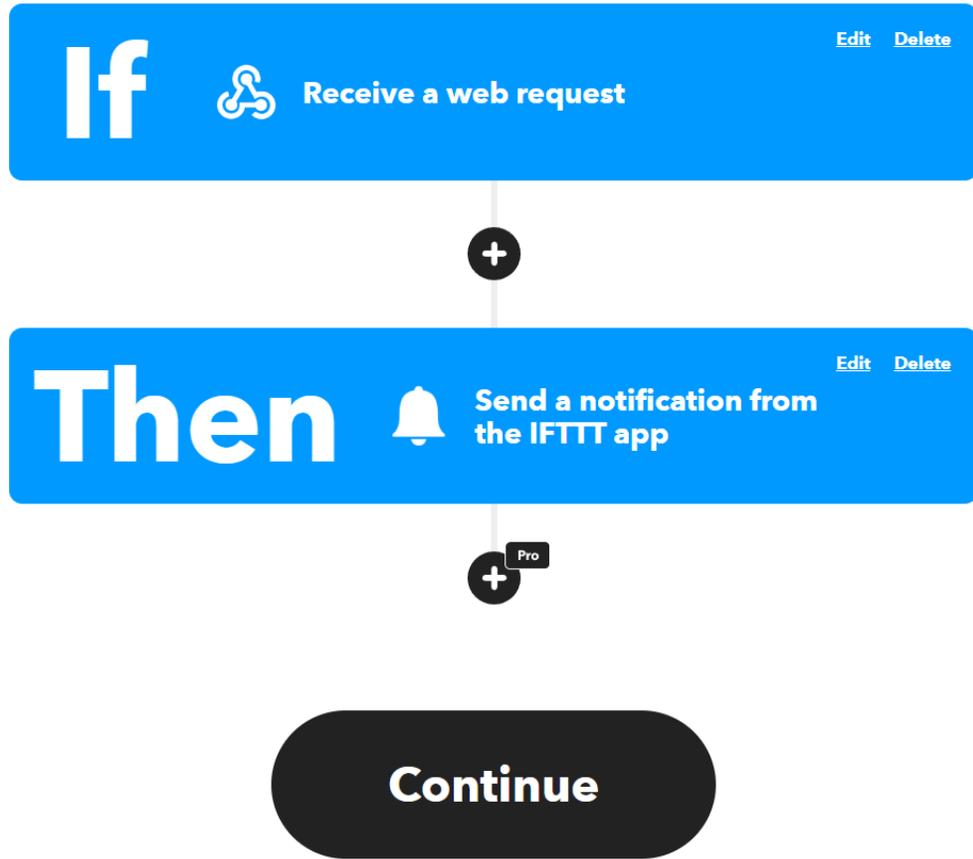


Рисунок 35 – Создание условия

Далее, перешли в раздел Webhooks и скопировали ключ API. После чего произведено программирование NodeMCU. В первую очередь требуется подключить микроконтроллер к сети Wi-Fi. Это показано на рисунке 36.

```
WiFi.begin(ssid, password);  
while (WiFi.status() != WL_CONNECTED) {  
    delay(1000);  
    Serial.print(".");  
}  
Serial.println("Connected to WiFi");
```

Рисунок 36 – Подключение NodeMCU к сети Wi-Fi

Второй шаг заключался отправке HTTP-запроса на Webhooks IFTTT с помощью следующих команд:

- sendIFTTTNotification,
- WiFi.Status.

Это показано на рисунке 37.

```

void sendIFTTTNotification(int layerNumber) {
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    String serverPath = String(serverName) + "?value1=" + String(layerNumber);
    http.begin(serverPath.c_str());
    int httpResponseCode = http.GET();

    if (httpResponseCode > 0) {
      Serial.print("HTTP Response code: ");
      Serial.println(httpResponseCode);
    } else {
      Serial.print("Error code: ");
      Serial.println(httpResponseCode);
    }
    http.end();
  } else {
    Serial.println("WiFi Disconnected");
  }
}

```

Рисунок 37 – Отправка запроса на IFTTT

Разработав программный код NodeMCU для каждой отдельной команды и произведя их тестирование собран и скомпилирован общий код, отвечающий всем вышеупомянутым требованиям. Текст программного кода приведен в приложении А.

Подводя итог по данной главе, можно сделать вывод, что успешно спроектирована и запрограммирована система об оповещении ошибки во время печати принтера. Также изучив функционал веб-сервиса для работы с умным домом IFTTT [30] разработана система удаленного оповещения пользователя о приостановке печати посредством отправки уведомления на мобильный телефон сообщения. Также в устройство встроена функция автоматического редактирования кода программы для 3D-принтера в целях ускорения восстановления печати.

Заключение

В рамках данной работы исследована проблема обрыва печати 3D-принтеров и предложены решения для восстановления точного позиционирования печатающей головки после внезапного отключения питания. 3D-печать, ставшая неотъемлемой частью современных производственных и бытовых процессов, предъявляет высокие требования к точности и надежности. Внезапные обрывы печати могут существенно нарушить производственные циклы, привести к финансовым потерям и увеличить износ оборудования. Это особенно критично в отраслях, где важна каждая минута, и в производстве сложных, уникальных изделий.

Решение проблемы обрыва печати важно как для промышленных предприятий, так и для обычных пользователей. В промышленности оно позволяет поддерживать непрерывность и стабильность производственных процессов, снижая издержки и повышая экономическую эффективность. Для домашних пользователей и малых предприятий это означает повышение надежности и удобства использования 3D-принтеров, что делает технологию более доступной и привлекательной.

В ходе работы проанализированы различные методы решения проблемы. В ходе анализа и экспериментов предложено решение в виде создания отдельного устройства для модификации 3D-принтеров.

Предложенные методы автоматического восстановления точного положения печатающей головки после обрыва питания продемонстрировали свою эффективность в ходе экспериментальных исследований. Они позволяют минимизировать количество брака и время на восстановление процесса печати, что повышает общую производительность и удовлетворенность пользователей.

Благодаря возможности удаленно получать информировании о прерывании печати оператор 3D-принтера имеет возможность ускорить процесс

создания моделей за счет дополнительно появившемуся времени, которое в обычное время тратится на наблюдение за процессом печати. Тем самым повышается эффективность использования 3D-принтера, что в свою очередь позволит приобрести дополнительную экономическую выгоду.

Таким образом, исследование проблемы обрыва печати и разработка эффективных решений для её преодоления имеет высокую практическую значимость. Эти решения могут быть интегрированы как в новые модели 3D-принтеров, так и в существующие устройства, что способствует повышению качества и надежности 3D-печати. Внедрение предложенных подходов позволит значительно сократить количество брака и потерь, связанных с обрывами печати, что в конечном итоге приведет к более широкому применению и развитию технологий 3D-печати как в промышленности, так и в быту.

Список используемых источников

1. 23 Шага к Победе – Шаг № 12 – И, Внезапно, Останов !? / [Электронный ресурс] // 3dtoday : [сайт]. – URL: <https://3dtoday.ru/blogs/ski/23-steps-to-victory-step-12-and-suddenly-stop-> (дата обращения: 24.03.2024).
2. 3D принтер остановился. Как продолжить 3D печать. Универсальный и бесплатный способ - 3DRadar / [Электронный ресурс] // 3dradar : [сайт]. — URL: <https://3dradar.ru/post/609/> (дата обращения: 28.11.2023).
3. G-код для 3D печати. Что это такое?: описание, подключение, схема, характеристики | ВИКИ / [Электронный ресурс] // 3d-DIY : [сайт]. — URL: <https://3d-diy.ru/wiki/3d-printery/g-kod-dlya3d-pechati-chto-eto-takoe/> (дата обращения: 14.07.2023).
4. Анализ стандартного кода для датчиков YL-63, HC-SR04 / [Электронный ресурс] // elibrary : [сайт]. — URL: https://elibrary.ru/download/elibrary_41518974_75681435.pdf (дата обращения: 17.03.2024).
5. Ардуино уно робот обход препятствий ик датчик / [Электронный ресурс] // shamrin : [сайт]. – URL: <https://shamrin.ru/arduino-uno-robot-obhod-prepyatstviy-ik-datchik/> (дата обращения: 16.03.2024).
6. Все о технологии MJP (Multi Jet Printing) в 3D-печати | 3Dcast / [Электронный ресурс] // 3dcast : [сайт]. — URL: <https://3dcast.ru/vse-o-tehnologii-mjp-multi-jet-printing-v-3d-pechati/> (дата обращения: 23.03.2024).
7. Купить недорого ИК-датчик препятствий для роботов-машин YL-63 FC-51 в интернет магазине Arduino-kit.ru / [Электронный ресурс] // Arduino-KIT : [сайт]. – URL: https://arduino-kit.ru/product/ik-datchik-prepyatstviy-dlya-robotov-mashin-yl-63-_fc-51_ (дата обращения: 21.03.2024).
8. Лидар (дальномер) на Arduino и датчике VL6180X: схема и программа / [Электронный ресурс] // Мир контроллеров : [сайт]. – URL:

<https://microkontroller.ru/arduino-projects/lidar-dalnomer-na-arduino-i-datchike-vl6180x/?ysclid=lxoklycw1y556004327> (дата обращения: 21.04.2024).

9. Лидарные датчики ToF и 10 их потрясающих применений / [Электронный ресурс] // Российские беспилотники : [сайт]. — URL: <https://russiandrone.ru/publications/lidarnye-datchiki-tof-i-10-ikh-potryasayushchikh-primeneniy/> (дата обращения: 15.01.2024).

10. Обзор инфракрасного модуля препятствия, LM393 – RobotChip / [Электронный ресурс] // RobotChip : [сайт]. — URL: <https://robotchip.ru/obzor-infrakrasnogo-modulya-prepyatstviya-lm393/>. (дата обращения: 07.03.2024).

11. Онлайн редактор блок-схем | Создать блок-схему онлайн | Programforyou / [Электронный ресурс] // Programforyou : [сайт]. — URL: <https://programforyou.ru/block-diagram-redactor> (дата обращения: 03.06.2024).

12. Печи для бани и сауны - Оборудование для бани, сауны и СПА - IdealSauna / [Электронный ресурс] // idealsauna : [сайт]. – URL: https://idealsauna.ru/catalog/oborudovanie/pechi_dlya_bani/ (дата обращения: 21.06.2024).

13. ПОВЫШЕНИЕ ТОЧНОСТИ ТВЕРДОТЕЛЬНЫХ ЛАЗЕРНЫХ ДАЛЬНОМЕРОВ ПУТЕМ ПЕРВИЧНОЙ ОБРАБОТКИ ДАННЫХ / [Электронный ресурс] // elibrary : [сайт]. — URL: <https://elibrary.ru/item.asp?id=48704315> (дата обращения: 03.06.2024).

14. Подключение инфракрасного датчика препятствия YL-63 к Arduino. / [Электронный ресурс] // portal-pk : [сайт]. – URL: <https://portal-pk.ru/news/272-26--podklyuchenie-infrakrasnogo-datchika-prepyatstviya-yl-63.html> (дата обращения: 01.06.2024).

15. Помощь начинающим в 3D Печати. Часть 2 / [Электронный ресурс] // 3D today : [сайт]. – URL: <https://3dtoday.ru/blogs/kirill35/help-beginners-in-3d-printing-part-2> (дата обращения: 21.05.2024).

16. ПРИМЕНЕНИЕ ЛАЗЕРНЫХ ДАТЧИКОВ РАССТОЯНИЯ VL53L0X

В СИСТЕМАХ ОБНАРУЖЕНИЯ ОБЪЕКТОВ / [Электронный ресурс] // elibrary : [сайт]. — URL: <https://elibrary.ru/item.asp?id=36392848> (дата обращения: 23.05.2024).

17. Проблемы и дефекты 3D-печати — методы их устранения / [Электронный ресурс] // Техно3Д : [сайт]. — URL: <https://3dpt.ru/page/faq> (дата обращения: 23.05.2024).

18. Разработка измерительных систем с применением контроллеров Arduino / [Электронный ресурс] // StudentFiles : [сайт]. — URL: <https://studfile.net/preview/6388754/> (дата обращения: 23.02.2024).

19. Разработка системы управления автоматизированным транспортным средством / [Электронный ресурс] // Репозиторий Dspace : [сайт]. — URL: <https://dspace.susu.ru/xmlui/handle/0001.74/11677> (дата обращения: 21.11.2023).

20. Ремонт лазерного дальномера своими руками / [Электронный ресурс] // Техника и Инструмент : [сайт]. — URL: <https://separett.su/instrumenty/remont-lazernogo-dalnomera-svoimi-rukami.html> (дата обращения: 02.06.2023).

21. РОБОТОТЕХНИКА - статьи / [Электронный ресурс] // РоботоТехника : [сайт]. — URL: <https://robototehnika.ru/content/article/samodelnyy-lidar-na-arduino-i-vl53l0x-tof-dalnomere> (дата обращения: 13.06.2023).

22. Самодельный лидар на Arduino и VL53L0X TOF-дальномере / [Электронный ресурс] // robototehnika : [сайт]. — URL: <https://robototehnika.ru/content/article/samodelnyy-lidar-na-arduino-i-vl53l0x-tof-dalnomere/> (дата обращения: 23.06.2023).

23. ТЕХПРОЦЕСС 3D-ПЕЧАТИ / [Электронный ресурс] // CyberLeninka : [сайт]. — URL: <https://cyberleninka.ru/article/n/tehprotsess-3d-pechati/viewer> (дата обращения: 11.04.2023).

24. УЛЬТРАЗВУКОВОЙ ДАТЧИК ИЗМЕРЕНИЯ РАССТОЯНИЯ HC-SR04 / [Электронный ресурс] // Cyberleninka : [сайт]. — URL: <https://cyberleninka.ru/article/n/ultrazvukovoy-datchik-izmereniya-rasstoyaniya-hc-sr04/viewer> (дата обращения: 14.04.2023).

25. Ультразвуковые датчики / [Электронный ресурс] // Mega-sensor : [сайт]. — URL: <http://mega-sensor.ru/PDF/uschoise.pdf> (дата обращения: 13.06.2024).

26. An Overview on 3D Printing Technology: Technological, Materials, and Applications - ScienceDirect / [Электронный ресурс] // sciencedirect : [сайт]. — URL: <https://www.sciencedirect.com/science/article/pii/S2351978919308169> (дата обращения: 16.06.2024).

27. Assistive System and GPS Tracker for the Visually Impaired | IEEE Conference Publication | IEEE Xplore / [Электронный ресурс] // IEEE Xplore : [сайт]. — URL: <https://ieeexplore.ieee.org/document/10441487> (дата обращения: 07.06.2024).

28. Evaluation of surface smoothness by a laser displacement sensor II: comparison of lateral effect photodiode and multielement array | Journal of Wood Science | Full Text / [Электронный ресурс] // jwoodscience : [сайт]. — URL: <https://jwoodscience.springeropen.com/articles/10.1007/s10086-003-0523-0> (дата обращения: 02.06.2024).

29. Laser Distance VL53L0X / [Электронный ресурс] // armonitor : [сайт]. — URL: <https://armonitor.com/dde/index.php/Main/LaserDistanceSensor> (дата обращения: 21.09.2023).

30. Comparative Study Method - ScienceDirect / [Электронный ресурс] // sciencedirect : [сайт]. — URL: <https://www.sciencedirect.com/science/article/abs/pii/S0143816622003207> (дата обращения: 23.09.2023).

Приложение А
Программа работы устройства

```
#include <Wire.h>
#include <VL53L0X.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClientSecureBearSSL.h>
#include <FS.h>
#include <SPIFFS.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);

// Объект для датчика
VL53L0X sensor;

// WiFi настройки
const char* ssid = "YOUR_SSID";
const char* password = "YOUR_PASSWORD";

// Twilio настройки
const char* account_sid = "YOUR_ACCOUNT_SID";
```

```

const char* auth_token = "YOUR_AUTH_TOKEN";
const char* from_number = "YOUR_TWILIO_NUMBER";
const char* to_number = "YOUR_PHONE_NUMBER";

uint16_t maxDistance = 0;

void setup() {
  // Инициализация последовательного соединения для вывода данных
  Serial.begin(115200);

  // Инициализация I2C
  Wire.begin(D2, D1);

  // Инициализация дисплея
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("Ошибка инициализации дисплея OLED"));
    while(1);
  }
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);

  // Инициализация датчика
  sensor.setTimeout(500);
  if (!sensor.init()) {
    Serial.println("Ошибка инициализации датчика!");
    while (1) {}
  }
}

```

```

// Опционально: настройка точности
sensor.setMeasurementTimingBudget(20000); // 20ms

// Подключение к WiFi
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(1000);
  Serial.println("Подключение к WiFi...");
}
Serial.println("WiFi подключен");

// Инициализация SPIFFS
if (!SPIFFS.begin()) {
  Serial.println("Ошибка монтирования SPIFFS");
  return;
}

// Чтение файла G-кода
readGCodeFile("/gcode.txt");
}

void loop() {
  // Получение значения расстояния
  uint16_t distance = sensor.readRangeSingleMillimeters();

  // Проверка на ошибки
  if (sensor.timeoutOccurred()) {

```

```
Serial.print("Ошибка: ");
} else {
Serial.print("Расстояние: ");
Serial.print(distance);
Serial.println(" мм");

// Обновление максимального расстояния
if (distance > maxDistance) {
    maxDistance = distance;
}

// Проверка условия и вывод на дисплей
if (distance > 0.2) {
    float result = distance / 0.2;
    display.clearDisplay();
    display.setCursor(0, 0);
    display.print("Расстояние: ");
    display.print(distance);
    display.println(" мм");
    display.print("Результат: ");
    display.print(result);
    display.display();

// Отправка SMS уведомления
    sendSMS(distance, result);
}
}
```

```

// Задержка перед следующим измерением
delay(500);
}

void readGCodeFile(const char* path) {
    File file = SPIFFS.open(path, "r");
    if (!file) {
        Serial.println("Ошибка открытия файла G-кода");
        return;
    }

    while (file.available()) {
        String line = file.readStringUntil('\n');
        parseGCodeLine(line);
    }

    file.close();
}

void parseGCodeLine(String line) {
    // Простой пример анализа строки G-кода
    // Пример строки G-кода: "G1 X50.0 Y50.0 Z10.0 F1500.0"
    if (line.startsWith("G1") || line.startsWith("G0")) {
        int index = line.indexOf('Z');
        if (index != -1) {
            float zValue = line.substring(index + 1).toFloat();
            if (zValue > maxDistance) {
                Serial.print("Найдено значение Z: ");
            }
        }
    }
}

```

```

        Serial.println(zValue);
    }
}
}
}

void sendSMS(uint16_t distance, float result) {
    if (WiFi.status() == WL_CONNECTED) {
        std::unique_ptr<BearSSL::WiFiClientSecure> client(new
BearSSL::WiFiClientSecure);
        client->setInsecure();
        HTTPClient https;
        https.begin(*client, "https://api.twilio.com/2010-04-01/Accounts/" +
String(account_sid) + "/Messages.json");
        https.addHeader("Content-Type", "application/x-www-form-urlencoded");
        String postData = "From=" + String(from_number) + "&To=" +
String(to_number) + "&Body=Расстояние: " + String(distance) + " мм. Результат: "
+ String(result);
        String auth = String(account_sid) + ":" + String(auth_token);
        https.addHeader("Authorization", "Basic " + base64::encode(auth));
        int httpResponseCode = https.POST(postData);

        if (httpResponseCode > 0) {
            String response = https.getString();
            Serial.println(httpResponseCode);
            Serial.println(response);
        } else {
            Serial.print("Ошибка отправки SMS: ");

```

```
Serial.println(httpResponseCode);  
}  
  
https.end();  
} else {  
Serial.println("WiFi не подключен");  
}  
}
```