

Власник документу:
Бевза Олег Миколайович

ID перевірки:
1004063768

Дата перевірки:
16.06.2020 03:15:37 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
16.06.2020 03:18:24 EEST

ID користувача:
90740

Назва документу: 2020-bachelor-EDD_Osokin_Manipulyator_fch

ID файлу: 1004076725 Кількість сторінок: 24 Кількість слів: 8482 Кількість символів: 62739 Розмір файлу: 82.46 KB

13.1% Схожість

Найбільша схожість: 2.26% з джерело бібліотеки. ID файлу: 1003485409

5.79% Схожість з Інтернет джерелами

60

Page 26

11.4% Текстові збіги по Бібліотеці акаунту

144

Page 26

0.68% Цитат

Цитати

1

Page 27

Вилучення переліку посилань вимкнено

0% Вилучень

Вилучений текст відсутній

Підміна символів

Не знайдено замієених символів

Осокіну Влдіславу

Система керування та позиціонування мобільного робота-маніпулятора
АНОТАЦІЯ

Метою виконання дипломного проекту є розробка системи керування для мобільного будівельного 3D-принтера на мікроконтролері STM32F4 Discovery для автономної роботи промислового робота; побудова алгоритму роботи та створення програмного забезпечення на мові C.

За основу системи позиціонування було взято стандартну прошивку 3D-принтера та прикладна програма для обробки 3D-моделей. Система позиціонування базується на алгоритмі SLAM (Simultaneous Localization And Mapping – одночасна локалізація та будування карти місцевості), завдяки якому є можливість одночасно реалізовувати як локалізацію робота (позиціонування у визначних межах), так і створення карти місцевості виходячи з отриманих вимірювань одометрії та датчиків, що дозволяє передбачити наявність перешкод на шляху робота.

Розроблено електричну принципову схему робота, до складу якої входять крокові двигуни, мікроконтролер STM32F4, стабілізатор напруги, ультразвуковий далекомір, драйвери крокових двигунів та блок живлення.

Ключові слова: система позиціонування, Graph-Based SLAM, 3D-принтер, кроковий двигун, мікроконтролер, датчики, побудова карти, джерело живлення, стабілізатор напруги.

ANNOTATION

The purpose of the diploma project is to develop a control system for a mobile construction 3D-printer on a microcontroller STM32F4 Discovery for autonomous operation of an industrial robot; construction of algorithm of work and creation of the software in language C.

The positioning system was based on the standard firmware of the 3D printer and the application program for processing 3D models. The positioning system is based on the SLAM algorithm (Simultaneous Localization And Mapping), thanks to which it is possible to simultaneously implement both robot localization (positioning within significant limits) and creating a terrain map based on the obtained measurements of odometry and sensors that allows you to predict the presence of obstacles in the way of the robot.

An electrical circuit diagram of the robot has been developed, which includes stepper motors, STM32 Discovery microcontroller, voltage stabilizer, ultrasonic rangefinder, stepper motor drivers and power supply.

Keywords: positioning system, Graph – Based SLAM, 3D printer, stepper motor, microcontroller, sensors, map construction, power supply, voltage stabilizer.

ВСТУП

Проблеми довгого будівельного процесу існують стільки, скільки існує цивілізація. Ця проблема гостра сьогодні, як ніколи раніше, і означає більше залучення машин і будівельників, більше забруднення атмосфери і більше експлуатації земельних ресурсів. Більше проблем. Це сьогоднішні реалії будівельного процесу.

Об'єктом дослідження є гусенична платформа, відтворена за допомогою тривимірного друку для малогабаритних робототехнічних комплексів. Мета бакалаврської роботи - створення системи керування для мобільної платформи малогабаритних роботів з використанням технологій 3D-друку, реалізація одного зі шляхів до автоматизації процесу побудов будівель, а також рішення, яке має такі переваги як швидкість побудови та мобільність у порівнянні зі своїми попередниками.

В процесі роботи була реалізована електрична і програмна частини. В результаті дослідження було створено малогабаритний робототехнічний комплекс. Створення та застосування подібного мобільного будівельного 3D-принтеру є актуальним з урахуванням вимог досить простої та швидкої відтворюваності деталей для ремонту або модифікації в важкодоступній місцевості. Область застосування – побудова будівель для автоматизації будівельного процесу. Ефективність гусеничної платформи, виготовленої виключно за допомогою 3D-друку, обумовлена невисокою собівартістю при виконанні серії друківаних об'єктів, порівняно високою швидкістю відтворення і зручністю реалізації [1].

Звичайний принтер для 3D-друку здатний виводити тривимірну інформацію, тобто створювати тривимірні об'єкти згідно з даними, які поступають з керуючого персонального комп'ютера. Принцип дії 3D-принтера полягає у послідовному накладанні тонких шарів матеріалу (пластику, металу та ін.) [2].

Основна проблема існуючих систем полягає у тому, що подібні технології не відрізняються високою швидкістю, але автономні технології позбавлені людського фактору і тому дозволяють вирішити задачу точності друку та ідентичності відтворюваних деталей оригіналу [1].

Будівельний 3D-принтер працює так само, як і звичайний, тобто використовує технологію екструджування [3]. Принтер використовує спеціальні будівельні матеріали. Вони видавлюються шар за шаром, поступово створюючи тривимірну модель конструкції, яка була задана програмою.

3D-друк знаходить широке застосування у виготовленні архітектурних макетів будівель, споруд, цілих мікрорайонів з прилеглою інфраструктурою (дорогами, деревами, вуличним освітленням). Як правило, для друку тривимірних архітектурних макетів використовують принтери, засновані на методі ZDP і в якості матеріалу гіпсовий композит, який забезпечує низьку собівартість готових моделей. Принтери використовують кольорові або рідше чорно-білі. Так, наприклад, «3D ZPrinter 250» або «3D ZPrinter 850» здатний відтворити тисячі відтінків кольорів, шляхом змішування. Але цим, звичайно, не обмежується застосування 3D-друку в будівництві, зараз на неї покладають великі надії оскільки 3D-друк будівель допоможе вирішити житлові проблеми багатьох людей. Розробка принтерів ведеться по всьому світу [5], так американські військові планують використовувати 3D-друк бетоном при розгортанні баз, китайські фахівці налагоджують будівництво бетонних блоків, іспанські розробники намагаються йти в напрямку

мініатюризації будівельних 3D-принтерів, створюючи роботів, здатних використовувати вже побудовані елементи будівель в якості робочої опори. Але крім розробок на даний момент є і готові прототипи.

Наприклад, компанія «BetAbram» випустила в продаж принтери для друку будівель площею до 150 м² у найбільшій моделі. Інший приклад - інженери з Массачусетського університету створили систему 3D-друку для роботи з великогабаритними об'єктами.

Система працює за принципом будівельного крана, який зводить стіни з шарів бетону. Для перекриття поверхів будуть потрібні бетонні плити. Робочим залишиться встановити вікна, двері і провести внутрішню обробку приміщення.

Який метод стане найбільш практичним, покаже час. Але в разі успіху будь-якого з них, будівельна галузь може зробити якісний ривок, виражений в підвищеній економії, безпеки і швидкості зведення будівель. Цілком можливо, що через кілька десятків років виростуть цілі міста з чудовими комфортними будинками, побудованими за технологією 3D-друку.

Застосування в робототехніці

Мабуть, ця одна з найактивніших областей застосування 3D-друку, де можна знайти приклади створення конструкцій, повністю виготовлених за допомогою 3D-друку. Інший, більш інноваційний приклад - м'який робот з вбудованими сенсорами, розроблений дослідниками з гарвардського університету. Тут подібні технології застосовані не тільки для друку конструкцій, а також сенсорів з органічних чорнил, вбудованих безпосередньо всередині основного матеріалу корпусу. Рідкий матеріал можна надрукувати на 3D-принтері в складі м'якого еластомеру, з якого складається корпус робота. Таким чином, вчені зуміли виготовити м'які сенсори замість твердих. Вони дозволяють вловлювати рух, температуру та інші типи впливу.

Бетонна конструкція у майбутньому піддається обробці, у тому числі декоративній. Перевагами даного методу є:

- Скорочення часу на будівництво об'єкту;
- Мінімізація ризику виробничих травм;
- Висока точність конструкції.

Будівельний 3D-принтер використовує технологію багатошарового друку.

Створені таким об'єктом конструкції є міцними та гладкими [2].

1. ОПИС АЛГОРИТМУ СИСТЕМИ ПОЗИЦІОНУВАННЯ

1.1 Огляд існуючих рішень робототехнічних систем

Однією з найбільш активно розвинених областей робототехніки є автономні мобільні робототехнічні системи. Важливою рисою таких платформ є можливість самостійно здійснювати навігацію в просторі. Найчастіше точна і детальна карта місцевості відсутня, тому активно розвивається область побудови роботою карти невідомій місцевості самостійно. При цьому роботу необхідно зберегти всю необхідну інформацію про об'єкти місцевості. Згодом ця інформація може бути використана для вирішення роботою різноманітних завдань у досліджуваній області. Другим

актуальним завданням є ідентифікація положення робота в обмеженому просторі, для якого вже побудована карта, при цьому необхідно наносити на карту знову розпізнані об'єкти під час руху робота [2].

Позиціонування всередині приміщення, або процес визначення роботом його положення в просторі, є однією з основних невирішених проблем в області навігації робототехнічних систем [3].

Системи позиціонування поділяють на три типи:

- глобальна - визначення абсолютних координат в просторі;
- локальна - визначення координат щодо заданих точок;
- персональна - визначення положення одних частин пристрою

відносно інших;

На рис. 1.1 зображено мобільний 3D-принтер.

Рис. 1.1 Зображення будівельного 3D-принтеру

В процесі роботи була спроектована принципова схеми системи керування і підібрані необхідні елементи, потім реалізована, включаючи електричну і програмну частини. Основні конструктивні і техніко-експлуатаційні показники: швидка відтворюваність деталей для ремонту або модифікації в важкодоступній місцевості.

Набір компонентів, які входять до принципової схеми: 6 крокових двигунів Nema17, які обрані для керування маніпулятором, та 2 крокових двигуна Nema34 для керування гусеничною частиною. Для перетворення вхідної напруги 24 В у 3В використовується імпульсний стабілізатор «LM2596» на 3А.

1.1 Структурна схема робототехнічної системи

Проектування здійснювалось таким чином, щоб алгоритм системи можна було застосувати в інших роботизованих пристроях, де використовується кутове керування кроковими двигунами або сервоприводами. Проектована система призначена як для використання в лабораторних умовах, так і в умовах виробництва виробів на 3D-принтері. Вимоги до системи: вона повинна бути: компактною, досить легкою, з високим ступенем інформативності. Перші дві умови реалізуються з використанням композитних матеріалів і мінімізації розмірів електронних компонентів і механічних пристроїв [6].

Окремі вузли проектуються спеціально для завдань принтера, зі зміною стандартних габаритних розмірів. Для реалізації третьої умови систему необхідно або забезпечити власним інформаційним табло, або забезпечити її взаємодію з персональним комп'ютером. Використання зробить систему більш гнучкою і спростить процес підготовки до друкування нового виробу. Контролер вирішує завдання формування траєкторії руху робочого органу, технологічних команд управління пристроями автоматики. 3D-принтер для своєї роботи вимагає відповідного пакету прикладних програм і досить швидкодіючого комп'ютера. Пакет прикладних програм являє собою стандартний набір для реалізації інтерфейсу користувача [3].

Загальна структурна схема та блоки робототехнічної системи представлені на рис. 1.2.

Рис. 1.2 Структурна схема мобільного роботу

З дисплею користувач через прикладну програму завантажує 3D-модель будівлі, після чого команди поступають на мікроконтролер, де виконується алгоритм керування кроковими двигунами. Для керування кроковими двигунами були застосовані драйвери крокових двигунів. З драйвера сигнал поступає на редуктор, на якому відбувається зниження зусилля до необхідного для обертання крокових двигунів. Редуктор був введений для мінімізації втрат потужності на керування кроковими двигунами [4].

Для блокування руху, забезпечуючи надійність роботи конструкції, були використані кінцеві вимикачі, які забезпечують необхідне положення мобільного робота.

1.2 Програмна частина

Програмна частина реалізується за допомогою програмного коду для роботи 3D-принтера та системи позиціонування, яка являє собою керування кроковими двигунами та датчиками за певним алгоритмом.

Реалізація програмної частини передбачає написання програми для знаходження початкового положення перед початком друку, з використанням прикладної програми для обробки 3D-моделей. При обробці програма автоматично переводить завантажену 3D-модель у послідовність координат, які у подальшому використовується для надання інформації о знаходженні 3D-моделі. При цьому треба врахувати, що робот для пересування згідно цим координатам, повинен визначати своє положення у просторі, яке обмежується ультразвуковими маяками. Для визначення положення у даний момент часу, був реалізований спеціальний алгоритм, який дозволяє не тільки обчислювати координати, але й за отриманими даними будувати карту місцевості. Також при реалізації було враховано керування кроковими двигунами, які складають основну компонентну базу, для них також був прописаний окремий алгоритм.

Для налаштування базових конфігурацій використовується стандартна прошивка 3D принтера, за допомогою якої виконується такі налаштування як температура екструдера, в якому розігрівається матеріал перед початком друку, швидкість друку та ін.

Структурна схема системи позиціонування представлена рис 1.3, де використовуються наступні позначення: ДПП–датчики початкового положення, ДКП–датчики кінцевого положення; КД–кроковий двигун, МК – мікроконтролер, М–маяк.

Рис. 1.3. Структурна схема системи позиціонування

Дана схема побудована за модульним принципом, причому кожний наступний модуль доповнює наступний блок і його основні функціональні можливості в рамках всієї системи керування принтером, окремих вузлів.

Для виготовлення 3D-принтера необхідно визначити елементну базу системи керування принтером. Всі основні компоненти системи керування можна розділити на 3 частини:

- крокові двигуни;

- драйвери крокових двигунів;
- контролер.

1.3 Алгоритм пошуку початкового положення

Забезпечення навігації мобільної робототехнічної платформи в просторі вимагає одночасного вирішення проблеми локалізації та побудови карти або її доповнення з новими виявленими об'єктами в умовах недосконалих інформаційно-вимірювальних засобів. Більшість сучасних реалізацій ділить поставлену задачу на дві основні підзадачі: завдання послідовного уточнення карти і завдання перерахунку свого положення в просторі.

Ці завдання повинна вирішувати мобільна робототехнічна платформа, яка отримує інформацію про зовнішній світ за допомогою датчиків і будує гіпотезу про своє місцезнаходження на основі обробки отриманих даних. В результаті розв'язання обох поставлених підзадач помітно ускладнюється через те, що датчики мають деяку модель помилки, точно обчислити і скорегувати яку в загальному випадку не представляється можливим. Тому роботу необхідно постійно коригувати дані про своє положення. Уточнення власних координат здійснюється за рахунок обчислення положення щодо деяких виділених орієнтирів [7].

У відповідність кожному виділеному орієнтиру ставиться опис, що дозволяє ідентифікувати його в кожен момент часу. Методики і технології для вирішення описаних задач отримали загальну назву SLAM (Simultaneous Localization And Mapping) – одночасна локалізація та будівництва карти місцевості [8].

Фільтр Калмана - це, напевно, найпопулярніший алгоритм фільтрації, який використовується в багатьох областях науки і техніки. Завдяки своїй простоті й ефективності його можна зустріти в обробниках показань датчиків, GPS-приймачах, при реалізації систем управління і т.д.

Фільтри Калмана базуються на дискретних за часом лінійних динамічних системах.

Так як рух автономних роботів погано апроксимується лінійним рухом, то фільтри Калмана не набули широкого поширення в реалізаціях завдання SLAM. У той же час, можливо застосування фільтрів Калмана для обчислення положення нерухомих просторових орієнтирів, так як вони задовольняють властивості лінійності. При використанні фільтра Калмана для отримання оцінок вектора стану процесу по серії вимірювань необхідно представити модель даного процесу відповідно до структури фільтра у вигляді матричного рівняння.

Розширений фільтр Калмана є не лінійною версією фільтра Калмана. У розширеному фільтрі Калмана моделі системи і спостереження не зобов'язані бути лінійними функціями, а можуть бути диференційованими.

Алгоритми SLAM можна розділити за методом, що використовується для розрахунків на три групи: а) На основі фільтра Калмана та його модифікацій, б) На основі фільтра частинок, в) Методи, засновані на графах. Кожен з цих підходів можна застосовувати для вирішення завдання SLAM,

але у кожного з них є свої переваги і недоліки, розглянемо кожен з підходів окремо [8].

1.4 Різновиди алгоритмів SLAM

а) Алгоритм FastSLAM.

FastSLAM-один з підходів до вирішення завдання SLAM. В основі алгоритму лежить фільтр частинок. Fast SLAM розділяє завдання локалізації та картографії на безліч підзадач, використовуючи не залежність стану окремих елементів моделі SLAM.

Метод полягає у безперервному вимірюванні показників з ультразвукових далекомірів, на основі нових та попередніх вимірювань робиться оцінка положення робота у даний момент часу. При вимірюванні виконується прогнозування подальшого шляху, які називаються гіпотезами або частинками. Кожна гіпотеза може бути правдивою або хибною, ця інформація отримується при подальших вимірюваннях. Система відсіює хибні гіпотези залишаючи тільки правдиві. Така фільтрація отриманих даних називали фільтром частинок.

б) Алгоритм DP-SLAM.

DP-SLAM-метод з принципово іншим підходом побудови карти місцевості. У цій постановці робот знаходиться в середовищі з перешкодами складної форми, якими є, наприклад, приміщення житлових будинків.

В таких умовах практично неможливо виділити окремі орієнтири. замість положення орієнтирів будується повна деталізована карта, для отримання достовірних результатів одночасно підтримка декількох гіпотетичних карт, за якими і відбувається фільтрація із застосуванням послідовних методів Монте-Карло. Для зберігання і структурування великого числа проміжних карт була реалізована деревоподібна структура.

Варто звернути увагу, що, незважаючи на велику спільність щодо середовища, цей метод розрахований на певний тип сенсорів-точні лазерні далекоміри і може потребувати окремих удосконалень для інших типів.

с) Алгоритм Gmapping. Даний алгоритм є одним з основних алгоритмів SLAM в операційній системі ROS (Robot Operating System), що обумовлено його відносною ефективністю і простотою підходу. В основі даного алгоритму лежить метод рекурсивної фільтрації частинок Рао-Блеквелла на сітчастій карті.

д) Кожна частинка є потенційною траєкторією робота і містить в собі, як потенційну траєкторію робота, так і інформацію про стан карти робочого середовища в поточний момент часу [10].

При такому підході на кожній ітерації алгоритму генеруються кілька нових частинок з кожної попередньої, при цьому їх число зростає по експоненті, що призводить до неможливості реалізації даного алгоритму.

Було запропоновано два підходи, які дозволяють збільшити продуктивність фільтра для можливості вирішення завдання SLAM в реальному часі, це: а) допоміжна функція розподілу, яка враховує точність сенсорів, таким чином, дозволяючи створювати карту місцевості точним способом; б) техніка адаптивного

Ресемплінгу, яка підтримує розумне число частинок для достатньої точності локалізації, в той же час, зниження ризику спустошення безлічі частинок. Допоміжна функція розподілу обчислюється шляхом оцінки ступеня схожості скану і карти за допомогою процедури реєстрації та одометричної інформації [9].

Для того, щоб принтер почав друкувати модель, йому потрібно задати початок координат, після чого за допомогою спеціалізованого програмного забезпечення виконується розділення 3D-моделі на відповідні шари, за якими буде друкуватись модель.

Пошук початкового положення виконується за алгоритмом Graph Based SLAM. Цей метод є найбільш доцільним для системи позиціонування, оскільки дозволяє здійснити постійне корегування при отриманні нових даних у режимі реального часу, а також є найбільш точним серед інших алгоритмів [5].

Розглянемо метод Graph Based SLAM більш детально.

I.5 Опис методу Graph-Based SLAM

Основу даного методу складає розширений фільтр Калмана для оцінки стану динамічної системи. Метод передбачає побудову графу, вузли якого являють собою позицію робота

і орієнтири

, а ребра - це зв'язки між відповідними вузлами, вони представляють інформацію про помилку позиції, що вимірюється датчиками руху або енкодерами.

Інформація про позицію робота буде отримана при вирішенні графу.

Приклад графу наведено на рис. 1.4.

Крім цього, метод Graph Based SLAM використовує розширений вектор стану

, який включає положення робота

і положення всіх елементів карти

Рис. 1.4 Граф переміщення робота відносно орієнтирів

Суттєва перевага Graph-Based SLAM полягає в тому, що кількість обчислень і пам'яті, які потрібні для оновлення та збереження коваріаційної матриці, не росте квадратично з числом елементів на відміну від Fast SLAM [6].

На самому початку, коли робот робить перші вимірювання, заповнюється коваріаційна матриця. Робиться припущення, що ці елементи не пов'язані між собою. Це означає, що недіагональні елементи нульові.

(1.1)

де

-покази ультразвукових датчиків,

- коваріаційна матриця

Проте, коли робот починає рух і робити нові виміри, місце розташування робота і орієнтири починають корелювати. Відповідно, коваріаційна матриця (1.1) перестає бути розрідженою (1.2).

Матриця кореляції потрібна для побудови карти місцевості, в якій додаються нові змінні по мірі руху робота. При пересуванні робота матриця кореляції заповнюється новими змінними (перестає бути розрядженою), ці змінні з'являються в матриці в наслідок отримання нових показників з датчиків робота.

Як було сказано раніше, математично алгоритм описується за допомогою матричного рівняння, рішення якого є знайденою координатою для побудови карти.

Принцип полягає у знаходженні нових перешкод на шляху робота, за рахунок чого створюється локальна карта місцезнаходження [4].

Спочатку формується карта робочої зони робота, при цьому зовнішнє середовище дискретизується. Також передбачається, що для визначення прохідності ділянок середовища використовується сенсорна підсистема сканування на основі звукового далекоміра, а визначення пройденого шляху здійснюється за допомогою одометрії. Побудова карти відбувається одночасно з дослідженням зовнішнього середовища.

Для уточнення прохідності ділянок робот здійснює послідовне сканування ділянок. Якщо ділянка вільна, то виконується переміщення в середовищі в напрямку центру першої ділянки. Причому величина переміщення дорівнює максимальному з габаритних розмірів ділянок, на які дискретизується навколишнє середовище.

Після складання карти середовища робот повинен використовувати її в процесі переміщень до точки. Однак отримана карта не може бути абсолютно точною через похибки вимірювань. Тому робот після кожного переміщення повинен виконувати уточнення карти, позиції на ній. Приклад побудови карти місцевості показано на рис. 1.5. Складання карти місцевості у вигляді алгоритму приведено на рис. 1.6.

Рис.1.5 Приклад карти місцевості побудованої за алгоритмом Graph-Based SLAM

Рис. 1.6 Алгоритм побудови карти місцевості

Для побудови систем локальної навігації і картографії застосовується як підхід на основі розширеного фільтра Калмана, так і на основі FastSLAM і Graph-Based SLAM.

Вибір того чи іншого методу залежить від необхідної точності системи і її швидкості, від параметрів середовища, таких як наявність, кількість і фізичні характеристики орієнтирів і від багатьох інших факторів.

Перед початком роботи отримується інформація з ультразвукових датчиків та вимірюється відстань до кожного маяка, таким чином система отримує значення для визначення свого положення та виконання алгоритму керування кроковими двигунами. За допомогою прикладної програми задаються значення, де повинна знаходитись модель. Структурна схема алгоритму зображена на рис. 1.7.

Рис. 1.7 Структурна схема алгоритму Graph Based SLAM

По мірі пересування робота отримуються нові данні о місцеположення відносно маяків, якщо значення відповідно заданих координат співпадають з

отриманими показниками, починається друк моделі, якщо ні, мобільний робот пересувається далі у межах певного діапазону, які задаються маяками.

Таким чином відбувається пошук співпадінь, та пошук початкового положення перед початком друку [7].

I.6 Принцип роботи прикладної програми – слайсера

Програмне забезпечення необхідно для підготовки та керування процесом друку об'єкта. Процес підготовки до друку складається з 3 частин: - створення і підготовка моделі програмного продукту (в даному випадку Солідворкс) Solidworks; - генерація керуючої програми власний пакет прикладних програм принтера; - завантаження керуючої програми в контролер принтера (інтерфейс). Створення і підготовка моделі. Для створення моделі використовують CAD-системи - програми тривимірного проектування або так звані 3D-редактори. 3D-редактори можна умовно розділити на 2 групи: програми поверхневого моделювання; програми твердотільного моделювання. Створення моделей для друку що розробляється на 3D-принтері здійснюється в програмах КОМПАС-3DV15 і SolidWorks 2018 року, які мають вбудований механізм експорту інформації в stl (stereolithography) - файл.

Для автоматизованого створення керуючої програми принтера застосовуються САМ-системи. Керуюча програма для контролера принтера повинна бути представлена в коді ICO-7 біт (G-code), який описує необхідні переміщення вузлів осей, управління температурою екструдера. G-code являє собою послідовність координат, відповідно яким принтер переміщується для друку моделі.

Завантаження отриманого G-code в контролер принтера може бути здійснена 2 способами: використання вбудованої системи контролю принтера та через програму управління процесом друку по провідному інтерфейсу; на SD-карті пам'яті в модуль, розташований на контролері принтера, і дозволяє працювати з зовнішніми флеш-накопичувачами. Програмне забезпечення, необхідне для управління процесом друку, складається з 2 частин: вбудовується системи контролера принтера; програми управління, встановленої на ПК [12].

Вбудована система – програмне забезпечення низького рівня, так звана прошивка контролера. У прошивці описані базові настройки і алгоритм управління підключених до контролера виконавчих пристроїв. Програма управління містить настройки друку, відображає контрольовані параметри температури екструдера, дозволяє спостерігати за процесом друку, відображаючи траєкторію екструдера в процесі роботи. Деякі виробники програмного забезпечення інтегрують підтримку слайсинга в програми управління процесом друку, що дозволяє використовувати одну програму замість двох. Нижче представлений інтерфейс стандартного продукту на базі вбудованої функції Windows версія з пакету прикладних програм Paint 3D.

Cura - це стандартна програма - слайсер для більшості 3D-принтерів. У програмі повністю відкритий вихідний код, її можливості можна розширювати за допомогою плагінів. Ця програма дуже легка у використанні

і дозволяє управляти найважливішими параметрами 3D-друку через зрозумілий інтерфейс. Програму Cura можна використовувати і для прямого управління принтером, але тоді принтер та комп'ютер повинні бути з'єднані один з одним. Базовий інтерфейс зображений на рис. 1.8.

Рис. 1.8 інтерфейс слайсера Cura

Робота програми складається з 10 етапів.

Етап 1: Вхідна інформація для створення слайсера. 3D-принтер створює об'єкти з масивами двовимірних шарів, як правило, зі 100% заповнюваністю при друку, і внутрішньої порожнини, де коефіцієнт заповнення варіюється від 10% до 100%. Також обирається форма сітки внутрішнього заповнення. Поверхні зовнішнього контуру і початок основи як правило, мають 100% заповнення при друкуванні.

Після того, як кожен шар генерується і застигає, відбувається корекція - обчислюється фактичне значення обсягу речовини (пластику) було використано на даному кроці, а потім генерується файл G-коду, що містить задані параметри (швидкість подачі матеріалу, температура матеріалу та екструдера і т.п.).

Етап 2: Налаштування середовища програмування. Головною частиною будь-якого проекту по створенню пакету керуючих програм є налагодження середовища або налаштування параметрів.

На даному етапі необхідно обрати програмний продукт, який здатний відтворити всі кроки по налаштуванню параметрів та отриманні інформації для коректної роботи.

Етап 3: Ініціалізація і шляхи руху друкувальної головки. На початку траєкторії руху друкуючої головки можливі чотири способи планування шляху і вихідних параметрів (режимів) друку:

- висота шару – відстань між нанесеними двовірними шарами;
- кількість периметрів: кількість твердих оболонок на зовнішній стороні моделі;
- ширина стінок у вертикальній площині.

Враховуються також параметри, необхідні для налаштування і подальшого функціонування системи:

- діаметр нитки;
- швидкість друку у мм/хв;
- температура сопла, в градусах Цельсія;
- ім'я вихідного файлу (G-коду).

Етап 4: Налаштування імпортованої моделі об'єкта. На даному етапі необхідно імпортувати файл, щоб розбити його на необхідні розділи. Для цього створюємо центральний вузол і з'єднуємо його з сіткою. Так, якщо обрати файл STL, то в засобі перегляду повинен з'явитися проект. Далі використовуємо можливості інтерфейсу і оболонки використовуваного програмного продукту для налаштування моделі.

Етап 5: Налаштування площини роздрукованого шару

Для кожного вузла потрібна точка "початку координат" для площині, а також необхідно вказати "нормальний" вектор руху друкуючої головки,

перпендикулярний базовій площині. Оскільки всі площини паралельні нагрівальній платформі базової моделі, то використовується прив'язка до осі Z. Створюємо точку відліку для початку друку в кожній з площин.

Для побудови точки потрібні три координати, причому координати X і Y не мають значення, тому можна просто встановити їх в нуль. Для координати Z використовується вбудована функція.

Етап 6: Створення полігонів(набір вершин, ребер, та граней, що описують форму багатогранного об'єкта) для одержуваного шару з урахуванням перетинів. На цьому етапі визначається точка (вузол) на зовнішньому боці деталі, де площина кожного шару перетинає сітку. Вузол перетину генерує об'єкти для кожного шару. Таким чином створюється серія полігонів для кожного шару.

Етап 7: Створення вертикальних шарів з урахуванням перетинів. Коли побудована зовнішня сторона деталі на кожному шарі, можна працювати над створенням вертикальних стін деталі. Оскільки не можна обмежуватися тільки однією центральною стіною по межі деталі, то необхідно використовувати полігони з останнього кроку. При побудові зовнішньої стіни, необхідно враховувати один край на межі, і, можливо більшу кількість побудованих стін всередині. Це означає, що повинен бути реалізований процес, який називається полігональним зміщенням. Програма не робить цього за замовчуванням.

Етап 8: Завдання режимів друку і швидкості подачі пластика. При роботі 3D-принтеру необхідно точно знати, скільки пластика необхідно для кожного руху. Цей параметр відповідає довжині нитки матеріалу, яку потрібно вказати в інтерфейсі. На цьому кроці необхідно обчислити ці значення (враховуємо довжину), щоб скорегувати і передати їх до керуючої програми принтера.

Етап 9: Налаштування G-коду. 3D-принтери використовують формат G-Code, який містить всі інструкції по друку деталі. На початку файлу G-Code скористаємося важливими настройками і інструкціями, які можна отримати в програмі. Один зі способів отримати настройки G-коду для свого принтера - це проаналізувати файл G-коду, згенерований програмою. Багато програм мають опцію "докладний", яка додає коментарі до файлу G-коду, що пояснюють кожну команду, або ж можна використовувати посилання на G-код. Для переміщення в G-коді використовується команда "G0" або "G1". Це перші реальні ходи, їх можна використовувати, щоб встановити швидкість. Наступні ходи успадковують цю швидкість, тому потрібно тільки вказати швидкість при її зміні.

Етап 10: наповнення G-коду даними конкретного пристрою 3D-друку. Основна частина файлу G-коду складається з друкованих ходів, які дотримуються тієї ж схеми, що і хід переміщення на останньому кроці. На кожному кроці враховуються значення E (нагрів пластини), контролюється температура та швидкість подачі матеріалу. Далі йде перевірка файлу G-коду, а також його перевірка в тестовому режимі і на дослідній деталі [7].

На рис. 1.9 наведено алгоритм роботи прикладної програми для завантаження 3D-моделі.

У даному розділі було показано алгоритм роботи мобільного робота та його системи позиціонування, розглянуто основні алгоритми SLAM, показано використання алгоритму Graph-Based SLAM за яким відбувається позиціонування робота за допомогою маяків, а також побудова карти місцевості. Показано використання програми-слайсера та принцип його роботи та базовий інтерфейс програми.

Рис. 1.9 Алгоритм роботи програми-слайсера

2. РОЗРОБКА СХЕМИ ЕЛЕКТРИЧНОЇ ПРИНЦИПОВОЇ

2.1 Огляд Мікроконтролера STM32 Discovery

У даній роботі використовується мікроконтролер STM32F4.

Обов'язковою атрибутом даних плат є велике число контактів, які дозволяють підключити відносно велику кількість компонентів для реалізації пристроїв на мікроконтролері, так як при цьому дається доступ майже до всіх портів мікроконтролера.

Вивідні колодки для всіх контактів введення / виведення мікроконтролера для швидкого підключення до макетної плати і простого проведення вимірювань [11].

Основні характеристики плати :

- 32-бітний мікроконтролер STM32F407VGT6 з ядром ARM Cortex-M4F з 1 Мб пам'яті програм і 193 Кб ОЗУ в 100-вивідному корпусі LQFP100 з тактовою частотою 168 МГц. Вбудовані операції з плаваючою крапкою (FPU);
- Вбудований програматор / відладчик ST-LINK / V2 з можливістю вибору режиму роботи (дозволяє програмувати зовнішні мікросхеми, використовуючи SWD-коннектор для програмування і налагодження);
- Живлення плати: через шину USB або від зовнішнього 5В джерела живлення;
- Живлення для зовнішніх пристроїв: 3В і 5В;
- 3-х осьовий MEMS акселерометр на базі мікросхеми LIS3DSH компанії ST;
- Цифровий MEMS мікрофон на базі мікросхеми MP45DT02 компанії ST;
- Аудіо ЦАП CS43L22 з вбудованим підсилювачем класу D;
- Дві кнопки (Reset і User);
- USB OTG з роз'ємом micro-AB;

Сімейство ARM Cortex - покоління процесорів, які виконані за стандартною архітектурою і відповідають різним технологічним вимогам. На відміну від інших центральних процесорних пристроїв (ЦПП) ARM, сімейство Cortex є завершеним процесорним ядром, яке об'єднує стандартне ЦПП і системну архітектуру.

Сімейство Cortex є в трьох основних профілях: профіль А для високопродуктивних застосувань, профіль R для реально-тимчасових застосувань і профіль М для чутливих до вартості і мікроконтролерних

застосувань. Мікроконтролери STM32 виконані на основі профілю Cortex-M3, яке спеціально розроблене для застосувань, де необхідні розвинені системні ресурси і, при цьому, мале енергоспоживання. Вони характеризуються настільки низькою вартістю, що можуть конкурувати з традиційними 8 і 16-бітними мікроконтролерами [11].

Cortex-M3 є стандартним мікроконтролерним ядром, яке крім ЦПП, містить всі інші складові основу мікроконтролера елементи (система переривань, системний таймер SysTick, відлагодна система і карта пам'яті). 4 гігабайтний адресний простір Cortex-M3 розділений на області коду програми, статичного оперативного запам'ятовуючого пристрою (ОЗП), пристроїв введення-виведення і системних ресурсів [10].

Cortex-M3 виконано по Гарвардської архітектурі, має кілька шин, що дозволяють виконувати операції паралельно. Сімейство Cortex має можливість оперувати з фрагментованими даними (unaligned data), що також відрізняє його від попередніх архітектур ARM. Цим гарантується максимальна ефективність використання внутрішнього статичного ОЗП. Сімейство Cortex також підтримує можливість установки і скидання біт в межах двох областей пам'яті розміром 1 Мбайт.

Основою STM32 є процесор Cortex-M3. Він являє собою стандартний мікроконтролер, інтегруючий 32-бітне ЦПП, шинну структуру, блок вкладених переривань, Відлагодну систему і зумовлену організацію пам'яті.

Ядро Cortex-M3 розроблялося як недороге ядро, воно залишається 32-бітовим ЦПП і, в зв'язку з цим, підтримує два режими роботи: потоковий режим (Thread) і режим обробника (Handler). В ядро Cortex також входить 24-бітний автоматично перезавантажувати таймер, призначений для генерації періодичних переривань і використовуваний ядром ОСРВ [4].

У сімейства Cortex передбачена підтримка набору інструкцій ARM Thumb-2.

Він являє собою суміш 16- і 32-бітних інструкцій, що дозволяють добитися продуктивності 32-бітного набору інструкцій ARM і щільності коду, властивій 16-бітного набору інструкцій Thumb. Thumb-2 - великий набір інструкцій, орієнтований на компілятори мов C / C ++. Це означає, що програма для Cortex-мікроконтролера може бути повністю написана на Cі.

Для розробки було обрано мікроконтролер STM32F4 (рис.2.1), який має наступні переваги:

- Висока продуктивність;
- Великий вибір модельного ряду;
- Великий вибір периферійних пристроїв;
- Програмна конфігурація процесора;
- Низьке енергоспоживання та розвинуте керування живленням;
- Низька вартість.

На сьогодні ARM Cortex-M3 стає, можливо, найпопулярнішим 32-розрядним процесорним ядром для вбудованих систем, а мікроконтролери на його базі мають універсальну, добре збалансовану архітектуру. Багато фахівців вважають STM32 одним з найпоширеніших, дешевих і

перспективних 32-розрядних мікроконтролерів. Мікроконтролер використовується для обробки даних з датчиків та керування роботою логіки датчика [9].

Рис 2.1 STM32 DiscoveryF407

2.2 Програма для налаштування STM32CubeMX

До сих пір середовища розробки для мікроконтролерів значно поступалися спеціалізованим середовищам розробки прикладного програмного забезпечення (ПЗ). Але візуальний редактор конфігурації і генератора коду STM32CubeMX, пропонуваний STMicroelectronics, дозволяє здійснювати тонке автоматизоване налаштування ініціалізації і генерувати базові проекти.

Варто відзначити, що інтеграція між складовими частинами виконано вдало, оскільки встановлювати необхідні бібліотеки можна прямо з редактора.

Бібліотеки можуть використовуватися самостійно при роботі з іншим програмним забезпеченням [7]. Пакет програмного забезпечення, який пропонується для використання, включає в себе наступні основні компоненти:

- бібліотеку CMSIS;
- бібліотеку для роботи з периферійними пристроями (Hardware Abstraction Layer - HAL);
- бібліотеки, що спрощують використання можливостей популярних плат (Board Support Package - BSP);
- бібліотеки вищого прикладного рівня, що дозволяють реалізувати спеціалізовані можливості (система реального часу FreeRTOS, підтримка файлової системи FAT, робота зі стеком протоколів TCP / IP завдяки бібліотеці lwIP, використання можливостей USB і ін.).
- Особливо слід відзначити бібліотеку HAL, яку можна назвати продовженням розвитку відомої бібліотеки Standard Peripheral Library (SPL), з якої вона має багато спільного в концепції організації програмного коду. Також архів з бібліотеками містить документацію, приклади коду і корисні додатки.

Основними функціями, які надає візуальний редактор конфігурації, є:

- налаштування тактування;
- налаштування периферійних пристроїв;
- розрахунок споживання енергії відповідно до планованими режимами роботи. Інтерфейс самої програми простий, зручний і зрозумілий. Незважаючи на наявність великої кількості налаштувань, він не перевантажений непотрібними функціями, а візуальне уявлення різних систем МК дозволяє повністю скласти картину про кінцевий результат налаштувань [8].

Візуальний редактор налаштувань STM32CubeMX (рис 2.2) є складовою частиною інфраструктури STM32Cube, яка також включає спеціалізовані бібліотеки для роботи з різними лініями мікроконтролерів STM32.

Рис 2.2 інтерфейс STM32 CubeMX

Програма CubeMXSTM32 дозволяє обрати початкові параметри, а також провести ініціалізацію портів перед початком написання програми, задати частоту тактування (рис.2.3).

Рис 2.3 налаштування таймера STM32

Після початку проекту можна побачити візуальне уявлення використовуваних виводів, вони позначаються зеленим, і неактивних, вони позначені сірим (рис. 2.4)

Також в лівій частині програми показані модулі мікроконтролера, які доступні для конфігурації. Використовувані відзначені зеленим кольором.

В даному випадку мікроконтролер вже налаштований під проект. Видно, що використовується модуль системи тактування.

Плюсом використання даного генератора проектів є те, що в будь-який момент часу можна підключити додаткові необхідні модулі, налаштувати їх, внести корективи в код програми, що робить програму більш зручною для використання [11].

Рис 2.4 Ініціалізація портів у STM32 CubeMX

2.3 Середовища розробки для STM32

Keil uVision дозволяє працювати з проектами будь-якого ступеня складності, починаючи з введення і редагування вихідних текстів і закінчуючи налагодженням коду і програмуванням ПЗУ мікроконтролера.

Серед основних програмних засобів Keil uVision можна відзначити.

- Базу даних мікроконтролерів, що містить докладну інформацію про всі підтримувані пристроях.
- Менеджер проектів, слугує для об'єднання окремих текстів програмних модулів і файлів в групи, оброблювані за єдиними правилами.
- Вбудований редактор, який полегшує роботу з вихідним текстом за рахунок використання багатовіконного інтерфейсу, виділення синтаксичних елементів шрифтом і кольором.
- Засоби автоматичної компіляції, компонування проекту, які призначені для створення виконуваного (завантажувального) модуля програми.
- Відладчик-симулятор, відладжує роботу компільованої програми на віртуальній моделі мікропроцесора.

Середовище розробки Atollic TrueSTUDIO була створена на базі популярної платформи з відкритим вихідним кодом - Eclipse [12].

Програма включає в себе: 0

- редактор коду (з підтримкою мов C / C ++ і асемблера);
- менеджер проектів, C / C ++ компілятор і систему збирання вихідного коду (з можливістю паралельної компіляції)
- відладчик (з підтримкою функції багатоядерного налагодження), а також різні додаткові інструменти для: аналізу коду (відповідно до стандарту кодування MISRA-C), управління завданнями, відстеження змін / оновлень, пристрій автоматично перевіряє створений код.

2.4 Вибір крокових двигунів

В сучасних принтерах застосовуються два типи приводів: крокові двигуни і сервоприводи. Принцип їх дії багато в чому схожий. Основна відмінність полягає у кроковому (дискретному) режимі роботи крокового двигуна (n - кроків на один оберт ротора) і плавності обертання синхронного двигуна.

Більша частина серед крокових двигунів гібридні. Гібридний двигун, це двигун в якому присутні постійні магніти з великим числом полюсів. Найчастіше на практиці двигуни мають 100 або 200 кроків на оберт, відповідно крок дорівнює 3.6 град. або 1.8 град. Найчастіше мікроконтролери дозволяють використовувати кроковий режим та мікрокроковий режим, тим самим зменшуючи кут удвічі.

За конфігурацією обмоток двигуни поділяють на біполярні та уніполярні. Біполярний двигун має одну обмотку в кожній фазі, для того щоб змінити напрямок магнітного поля, повинен бути підключений драйвер крокового двигуна. У випадку біполярного крокового двигуна доречним буде використати напівмостовий драйвер, драйвер з двополярним живленням.

Біполярний двигун має дві обмотки і, відповідно, чотири виводи. Уніполярний двигун відрізняється від біполярного, тим, що уніполярний двигун має вивід в середині обмотки.

Таким чином, для зміни напрямку магнітного поля достатньо перемикання між пів обмотками. Таке рішення дозволяє спростити схему драйвера. Саме тому в уніполярних двигунах мають інший спосіб зміни напрямку магнітного поля [11].

Завдяки об'єднанню середніх виводів обмоток, можна досягти наявності 5 або 6 виводів в середині двигуна. Іноді через те що уніполярні двигуни мають 4 обмотки, їх помилково називають 4-фазними. Через наявність окремих виводів в кожній обмотці, двигун має 8 виводів.

Двигун можна використовувати як уніполярний, так і біполярний в залежності від об'єднання обмоток.

Якщо виводи уніполярного двигуна, який має дві обмотки, два виводи залишити не підключеними, його також можна використати як біполярний. При цьому струм обмоток не повинен перевищувати максимальної потужності розсіювання.

Біполярні двигуни мають більш високу питому потужність у порівнянні з уніполярними. Біполярні крокові двигуни мають більший момент при тих самих розмірах. При створенні статором магнітного поля, момент, який створює кроковий двигун, є пропорційним величині магнітного поля. Для збільшення величини магнітного поля, треба збільшити число витків або величину струму. При цьому треба враховувати, що збільшення струму може призвести до насичення залізного осердя.

Однак на практиці це обмеження діє рідко. Ефективніше обмежити підвищення температури двигуна, за допомогою використання оммічних втрат в обмотках. Саме в цьому біполярні двигуни мають перевагу. У той час як в уніполярному двигуні використовується половина обмоток у кожній

момент часу. Через те що інша половина обмоток не задіяна, треба робити проводи іншого діаметра.

На відміну від біполярних двигунів, в яких весь час задіяні всі обмотки, тому їх використання є більш ефективним. Завдяки тому, що перетин окремих обмоток вдвічі більший, омичний опір вдвічі менший. Ця перевага дозволяє збільшити величину струму, при тих самих втратах, що дозволяє збільшити момент на 40% [10].

Якщо збільшення моменту не потрібно, біполярний двигун дозволяється зменшити габарити, або працювати з меншими втратами. Але завдяки тому, що уніполярні двигуни вимагають більш простих схем керування обмотками, їх застосування є більш розповсюдженим.

Дану перевагу використовують, якщо драйвер виконано на простих компонентах.

Драйвер, що використовується для крокової двигуна повинен не тільки забезпечувати комутацію необхідної кількості фаз, але відповідати за параметрами струму силового кола і забезпечувати можливість гасіння ЕРС самоіндукції, яка виникає в результаті комутації обмоток двигуна.

Для керування маніпулятором були обрані крокові двигуни Nema17 (рис. 2.5), для гусеничної платформи були обрані крокові двигуни Nema34 (рис. 2.6). Основні характеристики двигунів наведені у табл. 2.1 та табл. 2.2 відповідно.

Таблиця 2.1

Експлуатаційні характеристики Nema17

Рис 2.5 Призначення виводів крокового двигуна NEMA17

Таблиця 2.2

Експлуатаційні характеристики Nema34

Рис 2.6 Призначення виводів крокового двигуна NEMA 34

2.5 Алгоритм керування кроковими двигунами

Як середовище розробки програмного забезпечення було використувано безкоштовний пакет AtollicStudio с компілятором мови Cі . Змінна "лічильник інкрементується в обробнику переривання таймера T0. Таймер працює на частоті 1МГц з коефіцієнтом перерахунку 256. Тобто частота виникнення переривань $1000000/256 = 3906.25$ Гц.

Алгоритм приведений на рис. 2.7 ілюструє роботу крокових двигунів під час руху платформи. Спочатку проводиться ініціалізація портів мікроконтролера, після чого визначається довжина шляху, яку необхідно подолати до визначеної точки. Якщо обертання закінчилось, то спрацьовує система переривань мікроконтролера, якщо ні, дія продовжується.

Інформацію про кількість кроків дають одометричні пристрої, а місце зупинки платформи визначає координата, яка була задана перед початком роботи за допомогою прикладної програми [11].

Рис. 2.7 Алгоритм керування кроковими двигунами

2.6 Вибір драйверів крокового двигуна

В даний час існують спеціалізовані мікросхеми драйверів для біполярних двигунів, з використанням яких драйвер виходить не складніше, ніж для уніполярного двигуна. Наприклад, це мікросхеми L293E, L298N або

L6202 фірми SGS-Thomson, PBL3770, PBL3774 фірми Ericsson, NJM3717, NJM3770, NJM3774 фірми JRC, A3957 фірми Allegro, LMD18T245 фірми National Semiconductor.

Драйвер крокового двигуна – це мікросхема, яка дозволяє керувати кроковими двигунами шляхом подання високого та низького рівнів напруги на відповідні обмотки двигуна за прописаним у мікроконтролері алгоритмом, тим самим забезпечуючи безперебійну та чітку роботу електроприводів.

Силова частина контролера містить в собі напівпровідниковий підсилювач потужності, мета якого перетворювати імпульси струму, які подаються на фази у переміщення ротора. Керування кроковим двигуном здійснюється за допомогою контролера STM32F4, а також може реалізовуватись за допомогою карти пам'яті із завантаженим програмним кодом. Якщо параметри в ході виконання переміщення змінюються, режими роботи кроковим двигуном теж змінюються.

В залежності від виконуваних задач під час друку можна обирати режими керування кроковими двигунами. При необхідності керування малопотужними приводами, коли потрібно у кожен момент часу подавати імпульси на одну котушку статора, використовується одиночний режим керування.

Для досягання більш високого моменту обертання застосовують подвійний імпульс–у кожний момент часу на дві обмотки подається по одному імпульсу.

У даному випадку використовується блок керування кроковим двигуном, який дозволяють швидко інтегрувати кроковий двигун для виконання циклічних переміщень.

Перевагою даних блоків є синхронізація із зовнішніми пристроями та підтримка автоматичного включення та керування.

В якості драйвера крокових двигунів була обрана мікросхема типу DM556D (рис 2.8).

Рис 2.8 Кресленик драйвера крокового двигуна DM556D

Мікросхеми DM556D представляють собою набір потужних складових ключів з захисними діодами на виході. Наявність захисних діодів дозволяє підключати індуктивні навантаження без додаткового захисту від викидів зворотної напруги (рис 2.9) [9].

Рис 2.9 Схема підключення драйвера крокового двигуна DM556D

2.7 Вибір датчиків положення ротора двигуна

Датчик положення ротора або енкодер є невід'ємною частиною системи керування електродвигунами. Основним завданням датчика є визначення відносного положення магнітної осі ротора і еквівалентної осі статора, а також видача сигналів для керування.

Енкодер призначений для перетворення кута повороту обертового об'єкта в електричні сигнали. Він необхідний для управління синхронним і асинхронним двигунами, а також двигунами постійного струму [11].

Абсолютний енкодер має два контакти, порядок замикання яких залежить від напрямку обертання. При обертанні енкодер формує імпульси,

число яких пропорційно куту повороту. Число даних імпульсів дорівнює величині кута повороту вала енкодера, щодо його початкового положення.

Для даного випадку була обрана модель E40H12-15-3-N-24-C. Схема підключення енкодера представлена рис. 2.10. Сигнальні виводи енкодера підключаються до порту таймер-лічильника мікроконтролера. Оскільки контакти механічні, то при замиканні за рахунок пружності матеріалів, виникають багаторазові неконтрольовані замикання і розмикання.

Рис 2.10 Схема підключення датчика положення ротора двигуна

2.8 Вибір ультразвукового далекоміра

Далекомір – це пристрій для визначення відстані до об'єкта (рис 2.11).

Рис. 2.11 Ультразвуковий далекомір HC-SR04

Принцип роботи полягає у вимірюванні часу проходження відбитої звукової хвилі. Датчик генерує звуковий сигнал частотою 40 кГц у визначеному напрямку, потім сприймає його після відбиття від об'єкта та визначає час проходження звукової хвилі від датчика до перешкоди та назад.

Знаючи швидкість звуку в повітрі і час переміщення звуку до перешкоди, можна розрахувати пройденому звуком відстань за формулою:

де

- швидкість звуку в м/с,
- час в секундах.

У даному випадку обраний далекомір HC-SR04, який має 60 градусів кут огляду та низьке енергоспоживання. Характеристики далекоміра представлені у табл. 2.3.

Таблиця 2.3

Технічні характеристики ультразвукового далекоміра

Далекомір складається з двох п'єзо елементів. Один з них генерує сигнал, інший приймає. Відстань вимірюється від випромінювача до перешкоди і назад до приймача.

Головна перевага ультразвукового далекоміра у порівнянні від інфрачервоного полягає у тому, що на нього не впливають погодні умови: туман, блики сонця та ін.

2.9 Проектування джерела живлення

Оскільки напруга живлення мікроконтролера становить 3,3В, то для його досягнення необхідна мікросхема регулятора з малим падінням напруги. Для її вибору був врахований струм, споживаний мікроконтролером, величина якого становить 160 мА.

Для запасу по струму, дане значення було збільшено вдвічі, таким чином, мінімальне значення струму навантаження в обраній мікросхемі повинно перевищувати 320 мА.

В якості джерела живлення був обраний імпульсний блок живлення FlatPack 1500.

Технологія перетворення режимів, що передбачає помірну і високу частоту перемикавання, дозволяє мінімізувати обсяг і масу виробу і забезпечує швидке регулювання напруги.

Модуль сумісний з високим діапазоном вхідної напруги і забезпечує струм з плавним перемиканням живлення.

При функціонуванні в паралелі активного струму між модулями, працює при температурі навколишнього середовища від -40 до +70

С, при високих температурах вихідна потужність знижується і відбувається відключення, для забезпечення безпеки при температурах, що перевищують норму.

Технічні характеристики блоку живлення представлені в табл. 2.4:

Таблиця. 2.4

Технічні характеристики FlatPack 1500

2.10 Принципова схема робота-маніпулятора

На рис. 2.14 представлена структурна схема мобільного робота. При виборі основних компонентів для керування моторами потужністю 12 Вт було використано мікросхему драйвера для двигуна «DM556D», що включає два Н-моста і не вимагає великої обв'язки, крім байпас-конденсатора і набору захисних діодів. Таке під'єднання здатне видавати постійний струм до 2 А (потужність до 24 Вт) на канал, що відповідає критерію. До цифрових виводів мікроконтролера підключені драйвери крокових двигунів, які здійснюють керування кроковими двигунами. Також у схемі присутній ультразвуковий датчик, під'єднаний до цифрових виводів.

В якості виконавчих механізмів було обрано 6 крокових двигунів Nema 17, які обрані для керування маніпулятором, та 2 крокових двигуна Nema 34 для керування гусеничною частиною. Для перетворення вхідної напруги 24 В у 3 В.

Також використовується імпульсний стабілізатор «LM2569» на 3 А, що вимагає в якості обв'язки захисний діод, котушку індуктивності та два електролітичних конденсатори.

На рис 2.12 зображено схему підключення стабілізатора вихідної напруги до мікроконтролера STM32. Призначення виводів стабілізатора та компонентів наведені в табл. 2.5 та табл. 2.6 відповідно.

Таблиця 2.5

Призначення виводів стабілізатора напруги LM2569

Таблиця 2.6

Призначення компонентів стабілізатора напруги LM2569

Рис 2.12 Схема підключення стабілізатора напруги

На рис. 2.13 показано схему підключення крокового двигуна Nema17 та драйвера DM556D, всі інші крокові двигуни аналогічним чином приєднані до цифрових виводів мікроконтролера.

Рис 2.13 схема підключення драйвера та крокового двигуна

У табл. 2.7 наведено призначення виводів драйвера крокового двигуна.

Таблиця 2.7

Призначення виводів драйвера крокового двигуна

В даному розділі було показано проектування схеми на мікроконтролері STM32 Discovery, а також компоненти необхідні для роботи схеми.

В дану схему входять крокові двигуни, які забезпечують пересування робота та зміщення відповідних вузлів для друку, драйвери крокових

двигунів для керування кроковими двигунами, стабілізатор напруги, який забезпечує необхідну напругу на вхід мікроконтролера, джерело живлення, ультразвуковий далекомір.

Показано вибір перерахованих вище елементів та принцип їх роботи, а також їх основні характеристики. Показано основні етапи розробки за допомогою програм STM32 CubeMX та AtollicStudio.

Рис 2.14 Принципова схема робота

ВИСНОВКИ

Була проведена робота по дослідженню, конструювання та проектування друкувального пристрою. В ході виконання роботи були висвітлені основні типи систем позиціонування мобільних роботів, спроектована структурна схема, опис підібраних компонентів та пристроїв для коректної роботи. Для реалізації комп'ютерної інтегрованої системи керування 3D-принтером був запропонований і використаний програмований мікроконтролер STM32.

Був запропонований алгоритм для керування кроковими двигунами, спеціалізована програма для завантаження, обробки та реалізації 3D-моделей. Також запропонований алгоритм персональної системи позиціонування для знаходження початкового положення та подальшої роботи робототехнічної системи.

SUMMARY

Control and positioning system of the mobile manipulating robot

The diploma project of first educational level "Bachelor" by specialty 171 Electronics, specialization Electronic Systems Osokin Vladyslav Olexandrovich. National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute». Faculty of Electronics, Department of Electronic Devices and Systems. Academic group DS-61. - Kyiv: Igor Sikorsky Kyiv Polytechnic Institute, 2020. - 23 p., Ill. , tables 6.

Keywords: positioning system, 3D-printer, control system, mobile robot positioning, graph-based SLAM

Summary of the project:

Summary of the work: The problem of this work is that at the moment there is no technology that can provide mobility, speed and ease of use and transportation.

Therefore, this paper presents a solution for building and implementing a positioning system for a mobile construction 3D printer and building a map of the area using odometric indicators. Algorithms of operation of the main control devices are shown, as well as the choice of the main components and the principle of their operation, the main characteristics. Algorithms of work of the basic programs for work both with the 3D printer, and with the microcontroller are shown. The basic and structural scheme of the control system is shown.

The system is based on standard printer firmware, an application program for processing 3D models and the Graph-Based SLAM algorithm, which is used to implement positioning and construction of the terrain map.

This method was chosen for the reason that this mobile robot requires a very accurate positioning system to build a model.

Also shown is a schematic diagram of the robot control, which includes the following elements: microcontroller STM32 Discovery F4035, for movement, as well as for control of the main nodes selected stepper motors because they are more reliable in operation compared to servos.

To control the stepper motors, stepper motor drivers were selected, with the help of stepper motors high and low level signals are transmitted to the windings of the stepper motor, which rotates according to an algorithm written in C. Also added to the schematic sensors ultrasonic sensors and position sensors (encoders). Ultrasonic sensors measure the path to the beacons, which restrict the movement of the robot, position sensors are needed to measure the angle of rotation and their conversion into electrical pulses.

Consider the Graph-Based SLAM algorithm in more detail. This method is used for positioning mobile robots and at the same time from the construction of the terrain map, is used both for complex solutions (unmanned vehicles, aircraft, etc.) and for simpler robotic systems. This method is implemented both in the global positioning system (GPS, GLANAS) and in local and personal positioning systems.

In this case, a personal positioning system was created for the construction 3D-printer, which includes specialized software for finding the coordinates of the 3D-model and control of stepper motors.

The principle of this algorithm is based on the extended Kalman filter and involves entering in the initial state vector of all positioning reference points (which are beacons) and constructing a graph for more accurate positioning.

The construction of a graph, which is a set of points connected to each other, thus reflecting the trajectory of the robot, is carried out by entering the odometry, which are used as variables of the matrix equation, the solution of which is the coordinate of the robot.

Thus, first the data is obtained from the sensors, after which the data is updated and the next observation is performed. The next move compares already saved results with new results, adding new objects to the map. Summarizing the algorithm described above, this is how the Graph Based SLAM algorithm works.

Measurements are performed continuously, so the assessment of the position of the robot at this time is constantly updated, as well as objects on the map.

The map of the area is also built on the basis of ultrasonic rangefinders, but objects are added to the map based solely on the distance relative to the position of the robot.

The main tool for building and locating 3D objects is an application program with which the downloaded 3D model is divided into layers and thus translated into G-code format, which is a sequence of coordinates obtained by a 3D-printer to build a model according to these coordinates. This program also determines the amount of material needed for construction.

First, the program binds the starting point of the coordinates to the plane, then creates the basic polygons (a set of basic figures that make up the model) and the amount of material for each movement. This is followed by setting up the G-code,

which acts as an instruction for printing the part. The last step is to fill the G-code with data for 3D printing and determine the temperature of the part.

Also for setting the basic configurations, the printer firmware is used, where the microcontroller is selected, the temperature sensor and the rotation of the stepper motors are changed along the corresponding axes. The firmware also adjusts the limit switches, which are used to record the movements of stepper motors. With the help of the basic firmware, the basic adjustment of the extruder is performed by selecting the minimum temperature and distance of the nozzle to the ground, as well as the acceleration of the extruder along the axes.

Construction of the schematic diagram

When selecting the main components for controlling 12 W motors, the DM556D motor driver chip was used, which includes two H-bridges and does not require much strapping, except for a bypass capacitor and a set of protective diodes.

. Stepper motor drivers are connected to the digital outputs of the microcontroller, which control the stepper motors. An ultrasonic sensor connected to digital pins is also added to the circuit.

As actuators were selected 6 stepper motors Nema 17, which are selected to control the manipulator, and 2 stepper motors Nema 34 to control the caterpillar. To convert the input voltage of 24 V to 5 V.

The absolute encoder has two contacts, the order of closure of which depends on the direction of rotation. When rotating, the encoder generates pulses, the number of which is proportional to the angle of rotation. The number of these pulses is equal to the magnitude of the angle of rotation of the encoder shaft relative to its initial position.

Схожість

Схожість із джерелами з Інтернету

60

2	https://acts.kpi.ua/app/uploads/2017/05/%D0%9F%D0%9E%D0%A1%D0%86%D0%91%D0%9D%D0%98%D0%9A-ARM-	5 Джерело	1.71%
10	http://masters.donntu.org/2017/etf/buzulov/diss/indexu.htm		0.93%
11	http://www.dgma.donetsk.ua/docs/books/erasmus/2018-03-28/book2.pdf		0.92%
12	https://ela.kpi.ua/bitstream/123456789/31858/3/PKEMS-2_KonspLek.pdf		0.88%
13	http://dl.khadi.kharkov.ua/pluginfile.php/69721/mod_resource/content/17/L_15.pdf	2 Джерело	0.84%
20	https://docplayer.net/81689761-Referat-v-daniy-magisterskiy-kontrolni-roboti-rozglyanuti-instrumentalni-zasobi-ta-a	2 Джерело	0.38%
24	https://tef.kpi.ua/files/pdf/tezi_2017_tom2posl_var_1495705733.pdf		0.28%
25	https://www.stud24.ru/programming-computer/aparatnoprogramne-zabezpechennya-dlya-avtomatizac-spektrofotometra/504...		0.22%
26	https://studopedia.su/9_68665_printsip-dii-krokovih-dviguniv.html		0.2%
27	https://ela.kpi.ua/bitstream/123456789/28338/1/Zavalniuk_bakalavr.pdf	30 Джерело	0.18%
29	https://ela.kpi.ua/bitstream/123456789/23985/1/Varava_magistr.pdf		0.17%
32	https://uk.m.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%96%D0%B3%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D...		0.12%
36	https://pgm.kpi.ua/uk/2014-03-05-13-36-22	4 Джерело	0.11%
37	https://ela.kpi.ua/bitstream/123456789/33819/1/Davydenko_magistr.pdf	2 Джерело	0.11%
38	https://ela.kpi.ua/bitstream/123456789/30247/1/Spasova_bakalavr.pdf	3 Джерело	0.09%
39	https://stud.wiki/transport/3c0b65635b2ac78a5d43b89421316c36_0.html	3 Джерело	0.09%
40	https://ela.kpi.ua/bitstream/123456789/31426/1/Romanov_bakalavr.pdf		0.09%

Схожість по Бібліотеці акаунту

144

1	Студентська робота	ID файлу: 1003485409	Institution: Cherkasy State Technological University	2 Джерело	2.26%
3	Максим К.Є. КВ-73мп Маг. дис. 2018	ID файлу: 8245014	Institution: National Technical University of Ukraine	7 Джерело	1.59%

4	Студентська робота	ID файлу: 1000991565	Institution: National Aviation University	3 Джерело	1.52%
5	Студентська робота	ID файлу: 1960115	Institution: Lviv Polytechnic National University		1.38%
6	Студентська робота	ID файлу: 5958787	Institution: Donetsk National Technical University		1.17%
7	Студентська робота	ID файлу: 6032995	Institution: Lviv Polytechnic National University	2 Джерело	1.12%
8	Студентська робота	ID файлу: 6009774	Institution: Lviv Polytechnic National University	4 Джерело	0.95%
9	ПЗБД_КПІ_АУТС_2019_МолнарЕЕ	ID файлу: 1000095511	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		0.93%
14	Hrabovskyy_Kontroler parametriv tekhnolohichnoyi ridyny-fch	ID файлу: 1000087219	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		0.81%
15	Антиплагіат_Сороколат_Володимир (1)	ID файлу: 1004032800	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	3 Джерело	0.75%
16	Студентська робота	ID файлу: 1000073530	Institution: Lviv Polytechnic National University		0.68%
17	Диплом Гулак	ID файлу: 5627599	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		0.65%
18	2020-bachelor-EDD_Kirik_Nanotubes_fch	ID файлу: 1004076732	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	59 Джерело	0.57%
19	Мастенко ПБ-51	ID файлу: 1000057692	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		0.42%
21	Студентська робота	ID файлу: 1003704626	Institution: National University of Life and Environmental Sciences of Ukraine		0.35%
22	Диплом робота (Афзель С.С.)	ID файлу: 1000073952	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	3 Джерело	0.33%
23	+Patlaj_isz61	ID файлу: 1004048585	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	2 Джерело	0.28%
28	Varava	ID файлу: 5349331	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		0.17%
30	Гирич А.С	ID файлу: 1000073829	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	40 Джерело	0.13%
31	Plagiat_Bugayov	ID файлу: 5992511	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		0.12%
33	guzev_bakalavr	ID файлу: 1000063645	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		0.11%
34	Панасюк_УВ-41	ID файлу: 5862516	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	3 Джерело	0.11%
35	Студентська робота	ID файлу: 1103289	Institution: Lviv Polytechnic National University	5 Джерело	0.11%

Цитати

Цитати

1

- ¹ The positioning system is based on the SLAM algorithm (Simultaneous Localization And Mapping), thanks to which it is possible to simultaneously implement both robot localization (positioning within significant limits) and creating a terrain map based on the obtained measurements of odometry and sensors that allows you to predict the presence of obstacles in the way of the robot.