МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ АЕРОНАВІГАЦІЇ, ЕЛЕКТРОНІКИ ТА ТЕЛЕКОМУНІКАЦІЙ
КАФЕДРА ЕЛЕКТРОНІКИ, РОБОТОТЕХНІКИ І ТЕХНОЛОГІЙ
МОНІТОРИНГУ ТА ІНТЕРНЕТУ РЕЧЕЙ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____Володимир ШУТКО
«___»_____2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА
ЗІ СПЕЦІАЛЬНОСТІ 171 «ЕЛЕКТРОНІКА»
ОПП«ЕЛЕКТРОННІ ТЕХНОЛОГІЇ ІНТЕРНЕТУ РЕЧЕЙ»

Тема:« Система управління 3D принтером»

Виконавець:
студент групи ІР-307Б стн _____ Тарасенко Артур Павлович

Керівник:
к.т.н. доцент _____ Морозова І.В.

Нормоконтролер: _____ Сініцин Р.Б.

КИЇВ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ АЕРОНАВІГАЦІЇ, ЕЛЕКТРОНІКИ ТА ТЕЛЕКОМУНІКАЦІЙ

КАФЕДРА ЕЛЕКТРОНІКИ, РОБОТОТЕХНІКИ І ТЕХНОЛОГІЙ МОНІТОРИНГУ ТА ІНТЕРНЕТУ РЕЧЕЙ

171 «ЕЛЕКТРОНІКА»,
ОПП«ЕЛЕКТРОННІ ТЕХНОЛОГІЇ ІНТЕРНЕТУ РЕЧЕЙ»

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____Володимир Шутко
«___»_____2023 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи

Тарасенко Артур Павлович

(П.І.Б., випускника)

1. Тема кваліфікаційної роботи: «Система управління 3D принтером» затверджена наказом ректора від «23» березня 2023 р. № 387/ст.
2. Термін виконання роботи: з«23» березня 2023р. по«21» травня 2023р
3. Вихідні дані до роботи: розробити систему управління 3д прінтером.
4. Зміст пояснювальної записки: 1 Теоретичні основи розробки, 2 Апаратна реалізація, 3 Програмна реалізація.
5. Перелік обов’язкового графічного (ілюстрованого) матеріалу: таблиці, рисунки, зображення сенсорів, модулів, пристрою, код програми.

6. Календарний план-графік

№ п/п	Завдання	Термін виконання етапів	Відмітка про виконання
1.	Затвердження теми бакалаврської роботи	23.03.2023р	
2.	Вивчення літератури	24.03.2023р.-03.04.2023р.	
3.	Теоретичні основи розробки	04.04.2023р.-09.04.2023р.	
4.	Вибір технічних засобів	10.04.2023р.-22.04.2023р.	
5.	Апартно-програмна реалізація	23.04.2023р.-14.05.2023р.	
6.	Оформлення та усунення недоліків кваліфікаційної роботи	15.05.2023р.-21.05.2023р.	

Дата видачі завдання: «23» березня 2023 р.

Керівник кваліфікаційної роботи _____ Морозова І.В.

(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання _____ Тарасенко А.П.

(підпис випускника) (П.І.Б.)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Система управління 3D принтером» містить: 69 сторінок, 31 рисунок, 3 таблиці, 14 використаних джерел.

Актуальність теми полягає у створенні 3D принтера. Оскільки 3D друк стає все більш поширеним та доступним, зростає і потреба відповідних знань та навичок в галузі 3D друку, включаючи знання про системи управління 3D принтером. Також є потреба у поліпшенні цих систем для забезпечення ще більш ефективного та якісного друку.

Мета роботи – розробка системи управління 3D принтером.

Об'єкт дослідження система управління 3D принтер на базі Arduino Mega.

Предмет дослідження – система трьох вимірного друку.

Мета кваліфікаційної роботи – розробити систему управління 3D принтером.

Основним завданням даної роботи є створення системи управління 3D принтером. Робота включає в себе аналіз сучасних принтерів, огляд методів управління прінтером, аналіз ринкових моделей.

Матеріали цієї кваліфікаційної роботи можуть бути використані для проведення наукових досліджень, у навчальному процесі, а також з можливістю застосування у практичній діяльності для побудови системи управління 3D принтером.

Ключові слова: МІКРОКОНТРОЛЕР, ЕЛЕКТРОНІКА, ARDUINO. EEPROM.

ЗМІСТ

ВСТУП

Ще не так давно ідея про те, що на домашньому принтері можна надрукувати будь-що, була чимось із розряду наукової фантастики. Однак сьогодні фантастична мрія починає перетворюватися на споживчу реальність: 3D-принтери вийшли на масовий ринок і сьогодні практично кожен може придбати таку "домашню міні-фабрику".

3D-принтери були винайдені задовго до появи інтернету. Ось невеличкий екскурс в історію. Все почалося в 1981 році, коли доктор Хідео Кодама з Наґойського індустріального дослідницького інституту, що знаходиться в Японії, винайшов систему швидкого прототипування за допомогою фотополімерів. Він створював моделі шляхом нанесення шарів. Лише через три роки, у 1984 році, відбувся справжній прорив у цій галузі. Американський дослідник та засновник компанії 3D Systems Чарльз Халл винайшов стереолітографічний апарат (або SLA, що англійською розшифровується як Stereolithography Apparatus), який дозволив друкувати 3D-об'єкти з попередньо розроблених на комп'ютері моделей Матеріал являє собою рідкий акриловий полімер, який миттєво застигає і приймає необхідну форму при опроміненні ультрафіолетовим лазером. Таким чином з полімерного розчину нашаровується необхідна модель.

Повномасштабна комерційна модель SLA-1, випущена в 1987 році, зробила революцію серед винахідників, дозволивши їм створювати прототипи без величезних початкових інвестицій. А вже з початку 90-х років 3D Systems налагодила серійне виробництво SLA. В той самий час, інший стартап - DTM - отримав патент на технологію селективного лазерного спікання (англ. Selective

Laser Sinterning - SLS), створену Карлом Декардом з Техаського університету. Замість рідкого полімеру в цій технології використовувався металевий порошок.

Згодом компанія Stratasys під керівництвом Скота Крампа розробила нову технологію - моделювання методом наплавлення (Fused Deposition Modeling - FDM). Цей метод використовується для створення тривимірних об'єктів шляхом послідовного нанесення шарів матеріалу, які повторюють контури цифрової моделі. Зазвичай для цього використовуються термопластики, які завантажуються в принтер у вигляді спеціальних котушок ниток або прутиків. Зауважимо, що більшість сучасних базових 3D-принтерів використовують саме цей метод.

А от як не дивно сам термін "3D-друк" придумали - у 1995 році в Массачучетському технологічному інституті. І відтоді всі машини, що забезпечують 3D-друк, ми називаємо 3D-принтерами. Отже, для того щоб створити фізичний об'єкт за допомогою 3D-друку, спочатку необхідно створити його цифрову 3D-модель на комп'ютері і зберегти її у спеціальному форматі - STL. Потім цей файл обробляється принтером, який вираховує необхідні розміри та відтворює заданий об'єкт. Друкуюча головка наносить шар матеріалу, який потім може бути випечений спеціальним лазером або розплавлений відповідно до методу друку та характеристик принтера. Цей процес повторюється безперервно шар за шаром стільки разів, скільки необхідно, доки 3D-модель не буде повністю відтворена. [13].

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ

1.1. Огляд предметної області

3D-принтер - це спеціальний пристрій для відображення 3D-даних. На відміну від звичайного принтера, який відображає двовимірну інформацію на аркуші паперу, 3D-принтер дозволяє відображати 3D-інформацію, тобто можуть створюватися конкретні фізичні об'єкти. Технологія 3D-принтерів базується на принципі пошарового створення твердотільних моделей.

Перевагами такого 3D-друку зазвичай є швидкість, простота та низька вартість. Наприклад, залежно від складності продукту, створення моделі вручну може зайняти тижні або місяці. Як наслідок, значно зростають витрати на розробку і подовжуються терміни виготовлення готового продукту. За допомогою 3D-принтера моделі майбутніх виробів можна створити всього за кілька годин ,усуваючи при цьому можливість помилки, властиві "людському фактору".

Зазвичай 3D-принтери використовуються для швидкого виготовлення прототипів і використовуються в різних областях. Робота з реальними фізичними моделями пропонує безліч переваг тим, хто використовує технологію 3D-друку. Перш за все, це можливість оцінити ергономічність майбутнього виробу, його функціональність і збір, а також виключити можливість прихованих помилок перед запуском виробу в серію. Таким чином, можете заощадити значну суму грошей і часу за рахунок скорочення виробничого циклу.

Крім того, прототипи дозволяють тести, які не рекомендуються для готового зразка. Наприклад, Porsehe використовував прозору пластикову модель передачі 911 GTI для вивчення потоку масла під час його розробки. Слід зазначити, що така модель може бути зроблена дуже швидко-і в наші дні

високі швидкості дуже важливі. Деякі технології 3D-друку дозволяють виготовляти готові вироби з різних матеріалів.

Крім того, можливість швидко створити необхідну кількість освітніх моделей дає можливість вирішити багато проблем освіти. Крім того, 3D-друк широко використовується в медицині для створення моделей внутрішніх органів людини, протезів та імплантатів. Також великий інтерес становлять маркетингові аспекти 3D-друку. Завдяки цьому ви зможете підвищити якість роботи з клієнтами, демонструючи повноцінний прототип продукції. Ця технологія також використовується в тривимірній рекламі. Серед екзотичних застосувань 3D друку слід відзначити виробництво взуття. Поки що ця послуга розрахована на професійних спортсменів. Нога майбутнього власника сканується за допомогою лазера для створення цифрової моделі. На підставі цієї інформації створюється взуття . Таким чином, 3D друк є однією з найбільш перспективних технологій, які дозволять заощадити величезну кількість часу і зусиль для інженерів і дизайнерів [13].

1.2. Існуючі промислові реалізації принтерів аналогічних за технічними характеристиками

Розглянемо деякі моделі існуючих пристроїв, що подібні за характеристиками до пристрою що проектується. 3D принтер Prusha i3 [8].

Рисунок 1.1- 3D принтер Prusha i3

Таблиця 1.1- Характеристики 3D принтера Prusha i3

Тип	3D-принтери
Матеріал для друку	PLA , PETG-пластик
Технологія друку	моделювання методом наплавлення (FDM/FFF)
Кількість друкувальних головок	1
Інтерфейси	Micro SD , USB
Діаметр сопла	0.4 мм
Товщина шару	від 0.05мм до 0.35мм мм
Швидкість друку	до 200 мм/сек
Дисплей	Монохромний LCD дисплей.
Діаметр нитки	1.75 мм
Програмне забезпечення	CURA, Repetier-Host, Kissslicer
Розміри	420 x 420 x 380
Колір	Black
Гарантія	12 місяців

Розглянемо 3Д принтер Artillery Hornet [7].

Рисунок 1.2- Artillery Hornet

Таблиця 1.2- Artillery Hornet

Тип	3D-принтери
Матеріал для друку	ABS,Co-PET,HIPS,PET,PLA,PVA-пластик
Технологія друку	Моделювання методом наплавлення (FDM/FFF)
Кількість друкувальних головок	1
Інтерфейси	Micro SD , USB
Діаметр сопла	0.4 мм
Сфера побудови	220 x 220 x 250 мм
Товщина шару	От 0.05мм до 0.5мм мм
Швидкість друку	60- 100 мм/сек
Дисплей	Монохромный LCD дисплей, 2.3 дюйма
Діаметр нитки	1.75 мм
Програмне забезпечення	CURA, Repetier-Host, Kissslicer
Розміри	470 x 410 x 450 мм
Колір	Black

Гарантія	12 місяців

1.3 Обґрунтування вибору компонентів для проектування пристрою

Розглянемо вибір мікроконтролера для принтера, а саме : Arduino Mega 2560.

Рисунок 1.3 - Зовнішній вигляд Arduino Mega 2560

Arduino Mega 2560 є пристроєм, що базується на мікроконтролері ATmega 2560. Він має усі необхідні компоненти для зручної роботи з мікроконтролером: 54 цифрових входи / виходи (з них 15 можуть бути використані як ШИМ-виходи), 16 аналогових входів, 4 апаратних UART для послідовних інтерфейсів, 16-мегагерцевий кварцовий резонатор, USB-роз'єм, роз'єм живлення, роз'єм ICSP для внутрішньоінтегрального програмування та кнопку скидання. Щоб почати роботу з пристроєм, достатньо просто подати йому живлення від мережі або батарейки, або підключити його до комп'ютера за допомогою USB-кабелю. Arduino Mega сумісний з більшістю розширювальних плат, розроблених для Arduino Duemilanove і Diecimila. Mega 2560 є оновленою версією Arduino Mega 2560 і відрізняється від попередніх плат тим, що використовує мікроконтролер ATmega16U2 (у версіях плати R1 і R2 використовується ATmega8U2) замість мікросхеми FTDI для конвертації USB-UART інтерфейсу. На платі Mega 2560 версії R2 також додано резистор, який підтягує лінію HWB мікроконтролера

8U2 до землі. Такий підхід спрощує процес оновлення прошивки і перемикання пристрою в режим DFU. Зміни на платі версії R3 перераховані нижче: Терморегулятори 1.0: додані виводи SDA і SCL (поблизу виводу AREF), а також два нових виводи, розташованих біля виводу RESET. Вивід IOREF дозволяє платам розширення підлаштовуватися під робочу напругу Ардуіно. Цей вивід призначений для сумісності з 5В-платами Arduino на базі мікроконтролерів AVR та 3.3В-платами Arduino Due. Другий вивід не підключений до жодного компонента і залишений для майбутніх цілей [12].

Таблиця 1.3 - Технічні характеристики Arduino Mega 2560

Мікроконтролер	ATmega2560
Робоча напруга	5В
Напруга живлення (рекомендований)	7-12В
Напруга живлення (граничне)	6-20В
Цифрові входи / виходи	54 (з яких 15 можуть використовуватися в якості ШИМ-виходів)
аналогові входи	16
Максимальний струм одного виводу	40 мА
Максимальний вихідний струм виводу 3.3V	50 мА
Flash-пам'ять	256 КБ з яких 8 КБ використовуються загрузчиком
SRAM	8 КБ
EEPROM	4 КБ
Тактова частота	16 МГц

Ардуіно Мега може отримувати живлення як від USB, так і від зовнішнього джерела, і сама автоматично вибирає джерело живлення.

Якщо використовується зовнішнє джерело живлення (не USB), то можна використовувати мережевий AC/ДС-адаптер або акумулятор/батарею. Штекер адаптера (діаметр - 2.1 мм, центральний контакт - позитивний) потрібно підключити до відповідного роз'єму живлення на платі. Якщо живлення здійснюється від акумулятора/батареї, провід потрібно під'єднати до виводів Gnd і Vin роз'єму POWER.

Напруга зовнішнього джерела живлення може бути в діапазоні від 6 до 20 В. Проте, зниження напруги живлення нижче 7 В призводить до зменшення напруги на виводі 5V, що може спричинити нестабільну роботу пристрою. Використання напруги більше 12 В може призвести до перегріву стабілізатора напруги і пошкодження плати. Тому рекомендується використовувати джерело живлення з напругою в діапазоні від 7 до 12 В. Виводи живлення, розташовані на платі, перераховані нижче:

a) VIN. Це вивід, через який Arduino отримує напругу безпосередньо від зовнішнього джерела живлення (не пов'язаного з 5В від USB або іншою стабілізованою напругою). Через цей вивід можна подавати зовнішнє живлення на пристрій або споживати струм, коли пристрій живиться від зовнішнього адаптера.

b) 5V. На цей вивід надходить напруга 5В від стабілізатора на платі Arduino, незалежно від джерела живлення: адаптера (7-12В), USB (5В) або виводу VIN (7-12В). Не рекомендується живити пристрій через виводи 5V або 3V3, оскільки в такому випадку не використовується стабілізатор напруги, що може пошкодити плату.

c) 3V3. Цей вивід надає напругу 3.3В від стабілізатора на платі. Максимальний струм, який можна споживати через цей вивід, становить 50 мА.

d) GND. Вивід для заземлення.

e) IOREF. Цей вивід надає інформацію платам розширення про робочу напругу мікроконтролера Arduino. Залежно від напруги на виводі IOREF, плата

розширення може вибрати відповідне джерело живлення або використовувати перетворювачі рівнів, що дозволяє їй працювати з 5В або 3.3В пристроями.

Мікроконтролер ATmega2560 має 256 КБ програмної флеш-пам'яті (з яких 8 КБ використовує загрузчик), 8 КБ оперативної пам'яті SRAM і 4 КБ EEPROM (бібліотека EEPROM використовується для роботи з цією пам'яттю) [12].

Розглянемо пристрій для відображення інформації : Дисплей LCD 12864.

Інтелектуальний контролер включає в себе пристрій для читання SD-карт, поворотні датчики і 20-символьний × 4-рядковий дисплей LCD. Можна легко підключити його до плати RAMPS 1.4 за допомогою «розумного адаптера». Ця панель з'єднує RAMPS 1.4. Всі подальші операції, такі як калібрування, переміщення осі можуть виконуватися тільки за допомогою контролера, поворотного датчика. Тривимірний друк без комп'ютера просто використовує дизайн коду G, що зберігається на SD-карті.

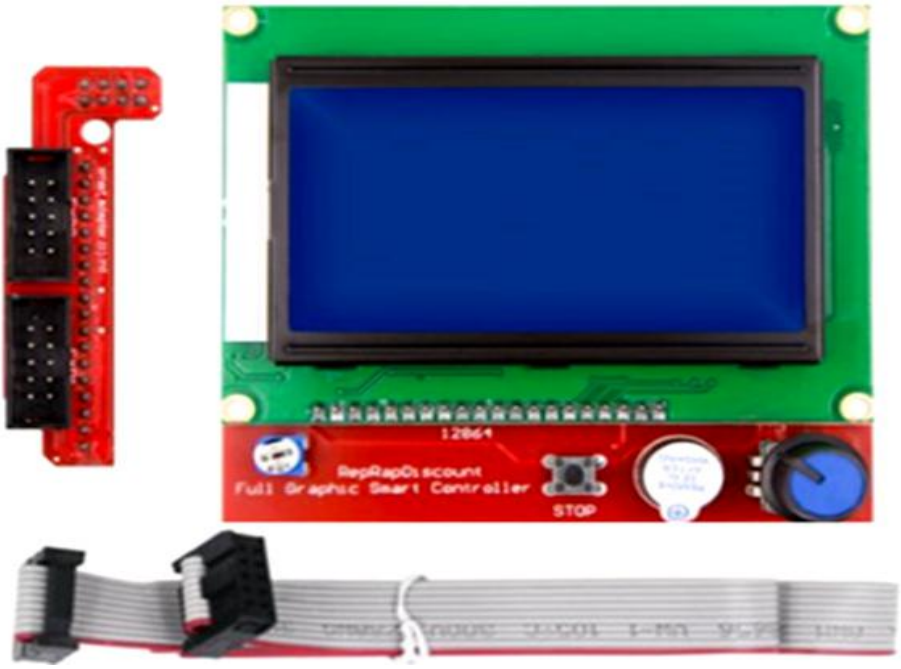

Рисунок 1.4 - Дисплей LCD 12864

При підключенні цього модуля до плати RAMPS 1.4 можна змінювати налаштування і стежити за процесом друку без підключення комп'ютера. [1]

RAMPS 1.4

Плата Arduino Mega є насадкою (шілдом) для контролера і є основою для створення саморобних 3D принтерів.

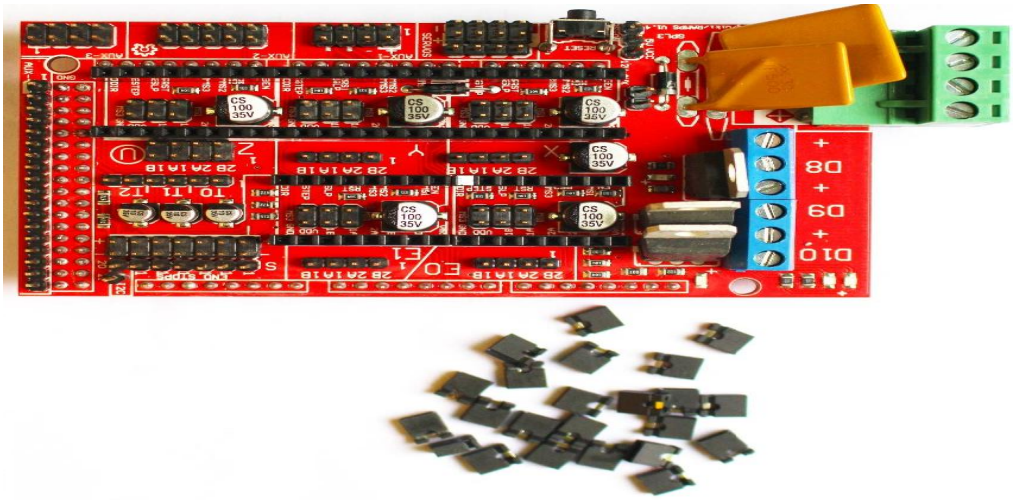

Рисунок 1.5 - RAMPS 1.4

Плата, сумісна з платою драйвера крокового двигуна A4988 або DRV-8825, призначена для управління двигуном у мікро кроковому режимі. На цій платі є роз'єм для підключення драйверів крокових двигунів і контролю екструдера, а також роз'єми електроніки для зручного обслуговування і заміни частин, а також можливість модернізації (розширення). Більше плат розширення Arduino можуть бути додані до системи, поки є вільні роз'єми на платі RAMPS 1.4 для удосконалення. Плата є найпоширенішою для більшості 3D принтерів які є на ринку і забезпечує можливість інтеграції всіх необхідних систем у компактному форматі.

Розглянемо Драйвер DRV8825 для крокових двигунів.

Технічні характеристики:

a) Напруга живлення: від 8,2 до 45 В;
b) Крок : 1, 1/2, 1/4, 1/8, 1/16, 1/32;
c) Напруга логіки: 3.3 В;
d) Захист від перегріву: Є;
e) Максимальний ток на фазу: 1.5 А без радіатора, 2.5 А з радіатором;
f) Розміри: 20 мм x 15 мм x 10 мм;
g) Габарити радіатора: 9 мм x 5 мм x 9 мм.

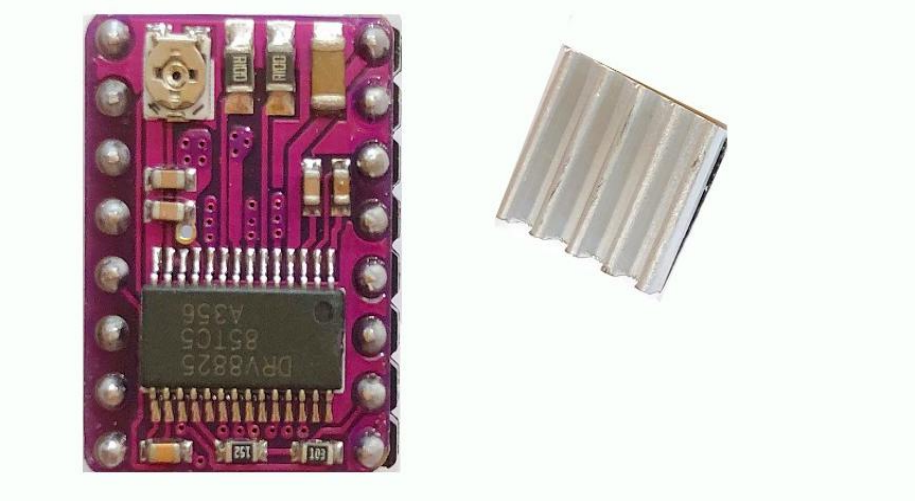

Рисунок 1.6 - Драйвер DRV8825

Основним чіпом модуля є драйвер від TI (Texas Instruments Inc.) DRV8825, який призначений для управління біполярним кроковим двигуном. Цей драйвер повністю сумісний з драйвером A4988. Мікросхема DRV8825 здатна працювати з вихідною напругою до 25 В і струмом до 1.5 А на котушку без використання радіатора, а з використанням радіатора (додаткового охолодження) - до 2.5 А. Модуль також має вбудований стабілізатор напруги, який живить логічну частину модуля напругою 3.3 В від джерела живлення двигуна.

Драйвер надає можливість використовувати шість різних режимів кроку: 1, 1/2, 1/4, 1/8, 1/16, 1/32.

На драйвері DRV8825 розташовано 16 контактів з такими функціями:

a) EN - включення або вимкнення модуля (0 - включено, 5 В - вимкнено);

b) M0, M1 і M2 - вибір режиму мікрокроку;

c) RST - скидання драйвера;

d) SLP - включення сплячого режиму; якщо його підтягнути до низького рівня, драйвер перейде в сплячий режим;

e) STEP - керуючий вивід, кожного позитивного імпульсу здійснюється крок двигуна (залежно від налаштування мікрокроку); чим швидше імпульси, тим швидше обертається двигун;

f) DIR - керуючий вивід; якщо подати +5 В, двигун буде обертатися за годинниковою стрілкою, а якщо подати 0 В, проти годинникової стрілки;

g) VMOT & GND MOT - живлення крокового двигуна від 8.2 до 45 В (обов'язково потрібно мати конденсатор на 100 мкФ);

h) B2, B1, A1 і A2 - підключення обмоток двигуна;

i) FAULT - вихід для захисту; якщо стан "0", це означає, що полеві транзистори H-моста відключені через захист від перевантаження або перегрів;

j) GND LOGIC - заземлення мікроконтролера.

Налаштування мікрокроку для крокового двигуна. Щоб налаштувати мікрокрок для крокового двигуна, використовують драйвер DRV8825, який може працювати в мікрокроковому режимі, дозволяючи подавати живлення на котушки зі змінними рівнями. Наприклад, якщо взяти кроковий двигун NEMA17 з кроком 1.8 або 200 кроків на оберт, у режимі 1/4, двигун буде видаляти 800 кроків за оберт. Для встановлення режиму мікрокроку на драйвері DRV8825 використовуються три виводи: M0, M1 і M2. Шляхом встановлення відповідних логічних рівнів на цих виводах можна вибрати режим мікрокрокування. Виводи M0, M1 і M2 на мікросхемі DRV8825 заземлені через резистор, тому якщо їх не підключати, двигун буде працювати в режимі повного кроку.

Розглянемо систему охолодження DRV8825. При інтенсивній роботі мікросхеми DRV8825 вона починає нагріватися, і якщо температура перевищить максимальні значення, може виникнути небезпека перегріву та пошкодження. Згідно з документацією, DRV8825 може працювати зі струмом до 2.5 А на котушку, але на практиці мікросхема не нагрівається, якщо струм не перевищує 1 А на котушку. Тому, якщо струм перевищує 1 А, необхідно встановити радіатор для забезпечення ефективного охолодження.

Розглянемо налаштування струму DRV8825. Перед використанням крокового двигуна потрібно зробити невеликі налаштування, зокрема обмежити максимальний струм, що протікає через котушки, і утримувати його в межах номінального струму двигуна. Це досягається шляхом регулювання потенціометра. Для налаштування потрібно розрахувати значення напруги Vref.

Далі залишається лише налаштувати його. Беремо викрутку і вольтметр, плюсовий щуп вольтметра підключаємо до потенціометра, а щуп заземлення - до виводу GND, і налаштовуємо потрібне значення струму [2].

Для зупинки двигуна використаємо механічний вимикач (EndStop).

Кінцевий механічний вимикач (EndStop) являє собою мікровимикач, розпаяний на платі роз'ємом 3S JST-XH для зручності підключення і індикаторним світлодіодом. Кінцеві вимикачі використовуються для визначення початку координат по всіх трьох осях (X, У, Z). Найчастіше на кожну вісь встановлюються по 2 вимикача для визначення початку і кінця по осі координат.

Характеристики

- габарити плати вимикача (д х ш х в): 40 х 16 х 7 мм;
- довжина кабелю: 70 см;
- вага плати вимикача: 3 г;

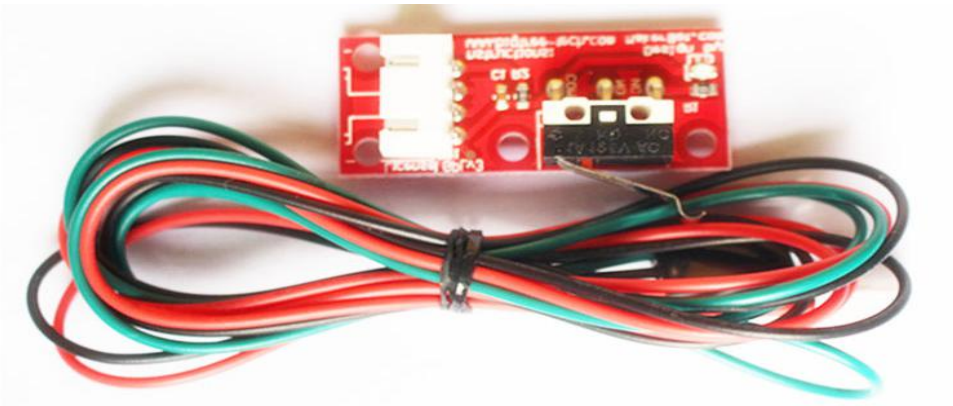

Рисунок 1.7- Механічний вимикач (EndStop)

Основне завдання вимикача полягає в тому, щоб зупинити кроковий двигун в потрібний момент.

Для друку використаємо Mk2b стіл для 3D принтера.

Розглянемо основні переваги столу.

- підтримка 12В, 24В.
- двостороння, на іншій стороні немає проводки, рівномірність нагріву, рівне друкування.

Для друку використаємо хотенд екструдер E3D V6 1.75мм для 3D-принтера[3].

Екструдер 3D-принтера - це компонент, що використовується для розплавлення пластику під впливом високої температури і виходу через отвір у соплі.

Характеристики:

a) Марка екструдера: E3D Chimera HotEnd ;
b) Діаметр вхідного прута матеріалу: 1,75 мм;
c) Діаметр сопла: 0,4 мм;
d) Сопло: 2 шт
e) Терморезистор: 100 ком;
f) Довжина проводів: 1 м;
g) Габарити кулера: 30 х 30 мм;
h) Нагрівальний блок:
i) Діаметр отвору для нагрівального елементу: 6 мм;
j) Діаметр отвору для терморезистора: 2 мм;
k) Монтажні отвори: м3;
l) Матеріал для друку abs, pla-пластик.

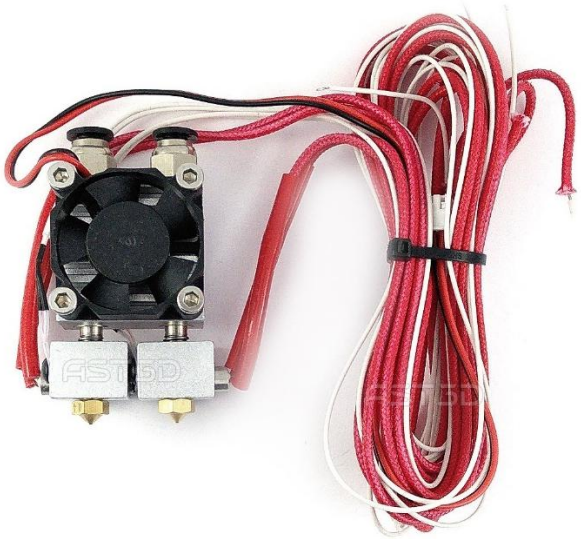

Рисунок 1.9- Экструдер E3D Chimera

Головною особливістю цього хотенду є подвійна подача пластику, тобто. наявність 2-го сопла. Використовується у 3D-принтерах для плавлення поданого пластику та видавлювання його через сопло.

В 3D-принтері буде використана нагрівальна платформа MK2A як основа для друку.

Рисунок 1.10 - платформа MK2A

Нагрівальна платформа MK2A - це модифікація нагрівальних платформ для 3D-принтерів. Мідний резистивний нагрівач зроблено на алюмінієвій підкладці, яка одночасно другим боком слугує рівною поверхнею для друку. Посередині платформи є отвір для монтажу терморезистора, а отвори для болтів кріплення мають конусоподібну форму, щоб шапки болтів були утоплені в рівень з самою поверхнею.

Алюмінієва платформа має два боки: один з чорною маскою і доріжками, що нагріваються, другий - рівний нічим не покритий алюміній, на який легко наклеїти адгезивний термокилимок. З такою комбінацією, легко отримати рівну адгезивну поверхню для друку, яка реагуватиме на індуктивні датчики наближення механізму автокалібрування [11].

Характеристики:

a) Фактичний розмір: 220x220x3мм;
b) Матеріал: алюміній;

c) Живлення: 12В;

d) Потужність: ~110-130Вт;

e) Вага: 395г.

Для пересування по осям використаємо кроковий двигун NEMA 17HS4401S .Двигун 17HS4401S - кроковий двигун (формат NEMA 17) з високим крутним моментом. Призначений для лазерних верстатів і для роботи 3Д - принтерів, плотерів, сканерів і т.д. Відрізняється точністю, що становить за оборот 200 кроків - це забезпечує поворот валу на 1,8 градусів за один крок [8].

Рисунок 1.11 - Кроковий двигун NEMA 17HS4401S

Характеристики:

a) Кутовий крок (кут повороту за 1 крок): 1.8 °;

b) Число фаз: 2;

c) Виводи мотора: дроти;

d) Номінальний струм: 1.7 А;

e) Опір фази: 1.5 Ом

f) Індуктивність фази: 2.8 мГн

g) Момент інерції ротора: 54 г х см2

h) Момент утримання: 4 кг х см

i) Масса: 0.28 кг ;

j) Зовнішні розміри: 42 мм х 42 мм х40 мм.

Для рами 3D принтера використаний рамовий профіль 20*40мм що дозволяє зробити стійку конструкцію.

Рисунок 1.12-Рама з металопрофілю

Для визначення необхідної потужності блока живлення ми проведемо розрахунок електроспоживання обраних компонентів 3D-принтера.

Для розрахунку напруги живлення та струму споживання компонентів системи 3D-принтера з урахуванням додаткової квадратної нагрівальної платформи, необхідно виконати наступні кроки:

a) Arduino 2560: За замовчуванням, Arduino 2560 живиться від напруги 5 вольт. Однак, можемо використовувати 12-вольтове джерело живлення для живлення Arduino за допомогою внутрішнього регулятора напруги. Приблизне споживання струму для Arduino 2560 при 12 вольтах становить приблизно 50-100 мА.
b) RAMPS 1.4: RAMPS 1.4 використовується як інтерфейс для керування крововими двигунами і нагрівальною платформою. Зазвичай, споживання струму RAMPS 1.4 є дуже незначним, близько 20-30 мА при 12 вольтах.
c) LCD 2004: LCD-дисплей 2004 зазвичай живиться від напруги 5 вольт. Для використання з 12-вольтовим джерелом живлення, можна використовувати внутрішній регулятор напруги або логіку рівня напруги для зниження напруги до 5 вольтів. Споживання струму LCD 2004

залежить від підсвічування та активності дисплея, і зазвичай становить близько 20-40 мА при 12 вольтах.

d) DRV8825: DRV8825 використовується для керування кроковими двигунами. Кожен DRV8825 може споживати близько 25-36 мА у бездіяльному стані, і до 2 А при максимальному навантаженні. Якщо у нас є 5 DRV8825, загальне споживання струму для них може бути близько 125-180 мА у бездіяльному стані та до 8 А при максимальному навантаженні.
e) NEMA17: Споживання струму крокових двигунів NEMA17 залежить від конкретної моделі та налаштувань. Вони можуть споживати від 0,5 до 2 А при 12 вольтах. Залежно від конфігурації крокових двигунів, загальне споживання струму може бути від 2 до 8 А.
f) MK2A плата нагрівання: Споживання струму MK2A плати нагрівання залежить від бажаної температури нагрівання та розмірів нагрівальної поверхні. Загальне споживання струму MK2A плати нагрівання може бути від 5 до 15 А.
g) Экструдер E3D Chimera: Споживання струму екструдера залежить від конкретної моделі та налаштувань. Вони можуть споживати від 4 до 5 А при 12 вольтах.

Загальне максимальне споживання струму системи складається зі споживання кожного компонента: 0,100 + 0,03 + 0,04 + 0,18 + 8 + 15 + 5 = 28,35 А

Отже, максимальне споживання струму 3D-принтера з урахуванням всіх компонентів та нагрівальної платформи може становити до 28,35 А при напрузі 12 вольт.

Для живлення використаємо імпульсний блок живлення JOTTA S-360-12

Рисунок 1.13- Імпульсний блок живлення JOTTA S-360-12

Блок живлення JOTTA S-360-12 — спеціально розроблений імпульсний блок живлення для 3D прінтерів, світодіодних лент . Робота блоку доволі проста, він приєднується до мережі живлення 220 В, на виході блоку після трансформації з'являється напруга 12 В. Вхід ~ 220 В, приєднується клемами. Блок живлення виготовлений у металевому перфорованому корпусі забезпечує прилади якісною стабілізованою напругою у 12 В за максимального навантаження до 30 А. Вихідна напруга водночас піддається регулюванню. Даний блок живлення призначений для внутрішньої установки і може працювати при температурі від -15°С до +50°С. На платі приладу присутня світлова індикація. Завдяки поліпшеним параметрами моделі, що відповідає світовим стандартам, забезпечується тривала стабільна працездатність блоку. Окремо тішить наявність захисту від коротких замикань або перевантажень, що підвищує в такий спосіб працездатність усієї системи загалом [2].

РОЗДІЛ 2. АПАРАТНА РЕАЛІЗАЦІЯ

2.1 Постановка задачі

Завданням даного проекту є розробка 3D принтера з ситемою керування на основі Arduino. 3D-принтер — пристрій, що використовує метод пошарового створення фізичного об'єкта за цифровою 3D-моделлю.Для реалізації даного завдання потрібно визначитися з технологією друку для принтера і сферою застосування.

Технології 3D друку :

a) Стереолітографія ;
b) Селективне лазерне спікання ;
c) Моделювання плавленням ;
d) Пошарове формування об'ємних моделей з листового матеріалу;
e) Струменева полімеризація;

Застосування 3D-друку;

f) Створення прототипів;
g) Швидке виробництво;
h) Дрібносерійне виробництво;
i) Навчальні моделі;
j) Медицина;
k) Реклама [13].

В проектуємого 3Д принтера використовуватиметься технологія FDM (Моделювання методом наплавлення) і даний принтер можливо буде використовувати для створення прототипів , дрібносерійного виробництва та виготовлення навчальних моделів.

2.2 Технічне завдання на розробку пристрою

В рамках даного проекту слід проаналізувати сегмент ринку і даній галузі та придбати всі комплектуючі разом в один готовий виріб.

3D прінтер призначений для створення фізичного об'єкта за цифровою 3D-моделлю. Головною платою даного пристрою повинна виступати Ардуіно Mega 2560, яка повинна бути запрограмована за допомогою комп'ютера в IDE Arduino. До її цифрових входів-виводів повинна підключатися Rams 1.4. До Rams 1.4 підключається драйвер крокового двигуна, кроковий двигун nema 17, нагрівальний стіл, термістори для контролю температури, кінцеві вимикачі, екструдер, LCD дисплей.

Вся інформація про прінтер виводиться на LCD екран. Також за допомогою енкодера задаються параметри принтера.

Живлення пристрою повинно бути реалізовано за рахунок імпульсний блок живлення 12V 30A і підключено до Rams 1.4.

Габарити пристрою повинні бути 40*40*30 см, дані характеристики залежать від рами принтера який використаний в даному проекті.

Потрібно підключити до комп'ютера та залити на Arduino Mega 2560 прошивку, потім налаштувати ток крокових двигунів. Після того потрібно налаштувати місце парковки і налаштувати стіл. Потім запустити тестовий друк.

2.3. Етапи проектування пристрою

На початковому етапі проектування слід розібратися, як підключити всі модулі між собою. Для виконання даної цілі було спроектовано схему, дану схему можна переглянути на рисунку 2.1.

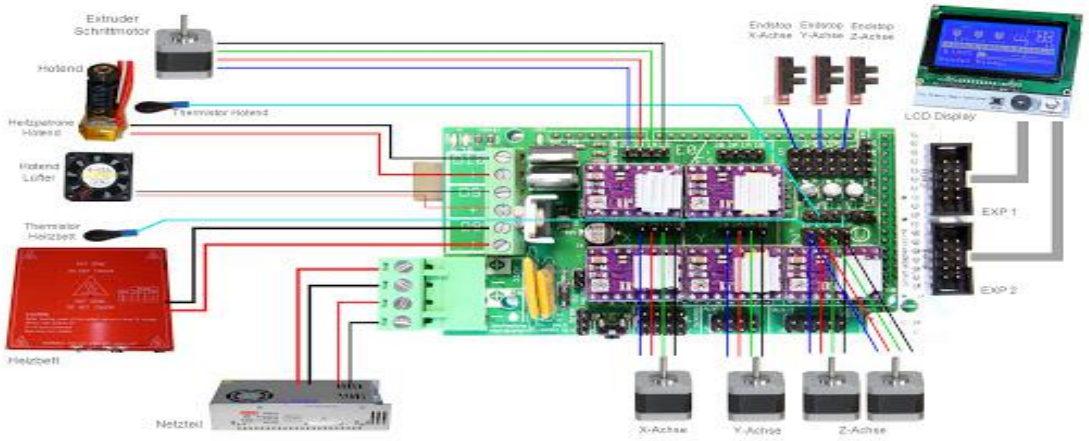

Рисунок 2.1. - Схема електрична функціональна підключення 3д принтера

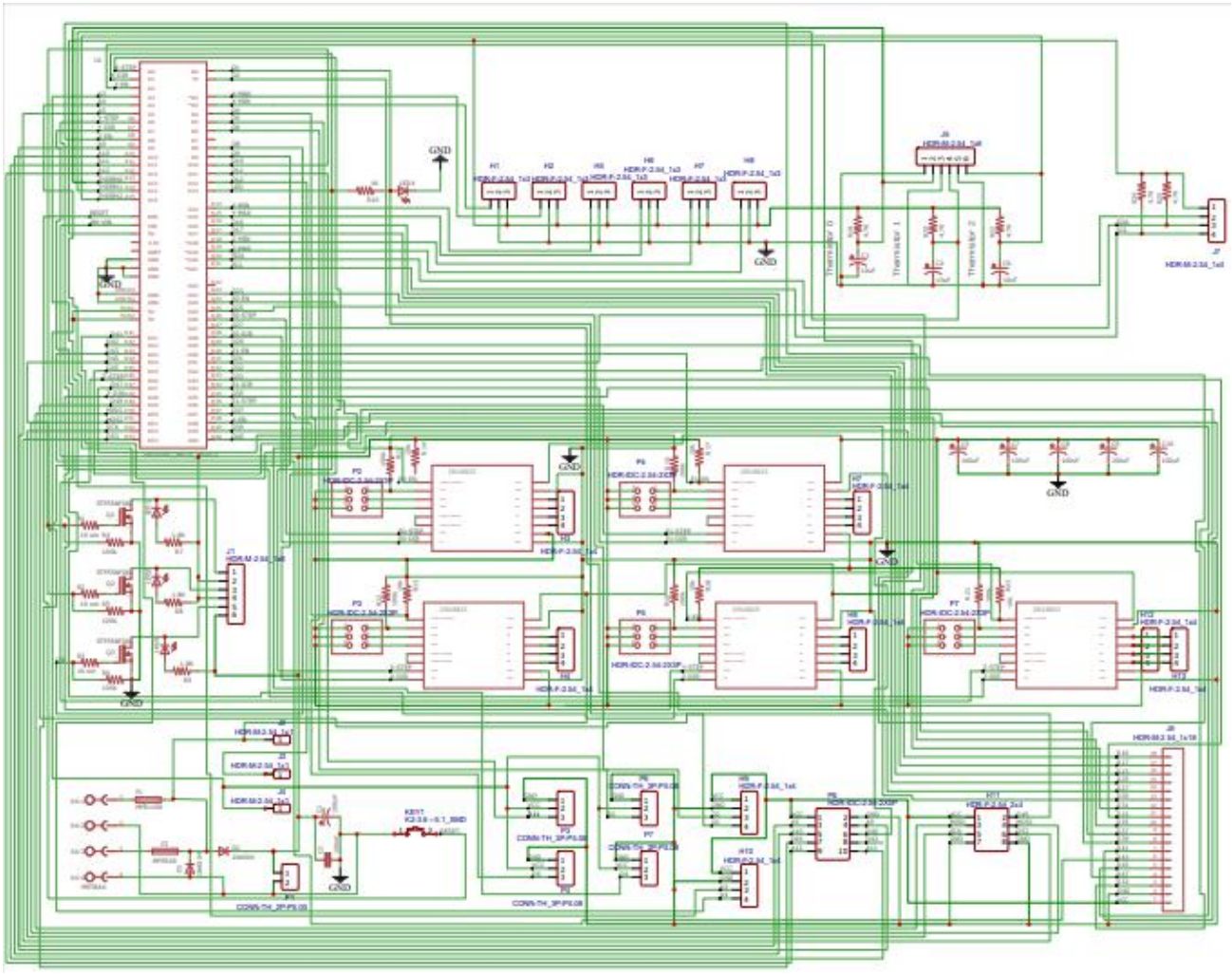

Рисунок 2.2. - Схема електрична принципова 3д принтера

Для ефективного та надійного функціонування системи, важливим є створення також алгоритму роботи 3D принтера, що базуються на розроблених елементах. Це дозволить забезпечити оптимальне взаємодію між компонентами принтера

та забезпечити його безперебійну роботу. Крім того, розробка такого алгоритму та схеми відкриває можливість для подальшого вдосконалення системи, включаючи можливість розширення функціональності, підвищення продуктивності та покращення якості друку.

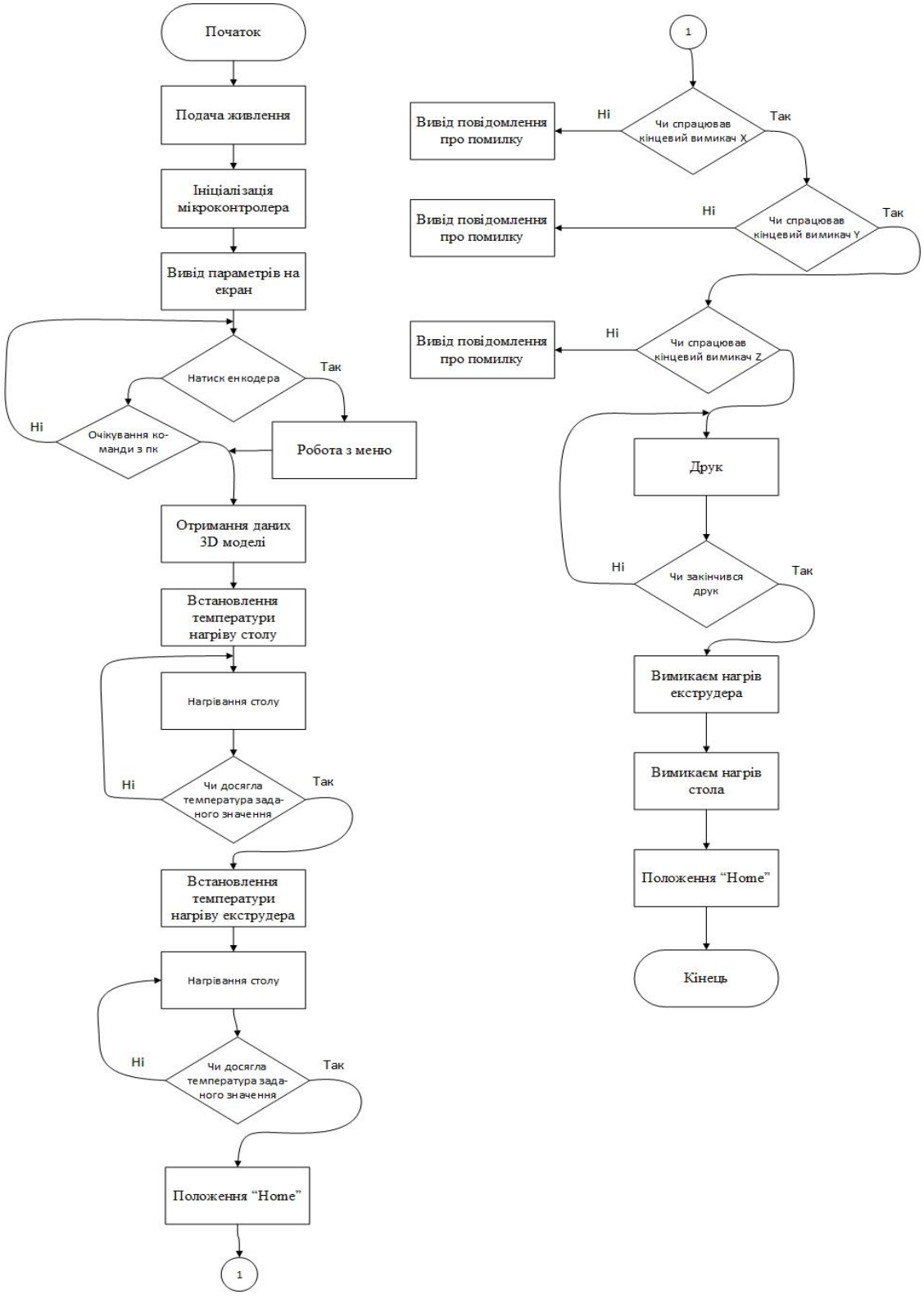

Рисунок 2.3. - Алгоритм роботи 3D принтера

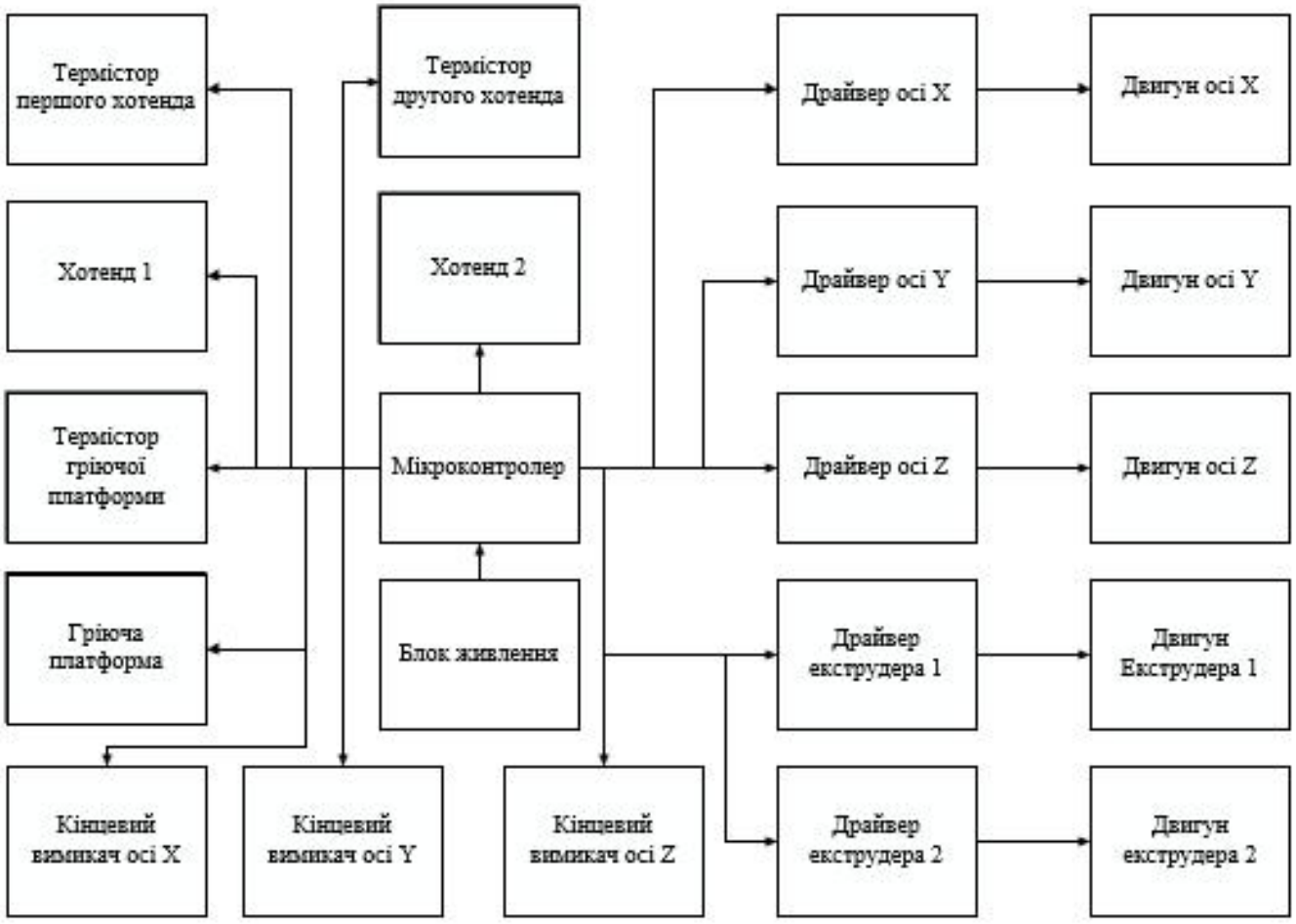

Рисунок 2.4. - Схема структурна 3д принтера

Збірка 3D-принтера із зазначеними компонентами може бути дещо складною, але я дам загальну інструкцію, яка допоможе зрозуміти процес.

a) Рама: Зберіть раму 3D-принтера, використовуючи профілі 30x30x40 см або схожі компоненти. Інструкції зі збірки повинні бути надані виробником рами. Дотримуйтеся кроків, описаних в інструкціях, щоб правильно збудувати раму принтера.
b) Електроніка: Підготуйте Arduino Mega 2560, RAMPS 1.4 і LCD 2004 для підключення. Перед підключенням компонентів ретельно прочитайте документацію і забезпечте правильну орієнтацію та встановлення кожного компонента.
c) Підключіть Arduino Mega 2560 до RAMPS 1.4 згідно рисунка 2.1.

d) Встановіть DRV8825 відповідно до схеми на плату RAMPS 1.4. Пам'ятайте, що у DRV8825 є певна орієнтація, яку слід дотримуватися при підключенні. Потім потрібно налаштувати ток драйвера 0,68В. Мінусовий щуп мультиметра приєднуємо до контакту GND (мінус, він загальний), плюсовим торкаємося до корпусу підлаштування резистора на драйвері. Крутимо підлаштування резистор викруткою і заміряємо розрахункове напруга Vref . Таким чином ми виставляємо правильний струм для драйвера крокового двигуна. Для кожного виду драйвера своя формула розрахунку Vref .

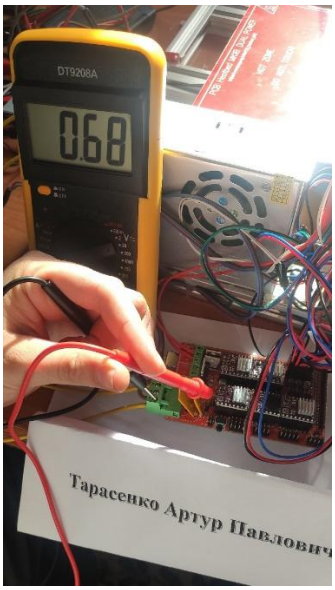

Рисунок 2.5. - Налаштування току двигуна

e) Підключіть NEMA17 крокові двигуни до DRV8825.

Рисунок 2.6. - Підключення крокового двигуна для пластика і екструдера

f) Підключіть LCD 2004 до відповідних контактів.

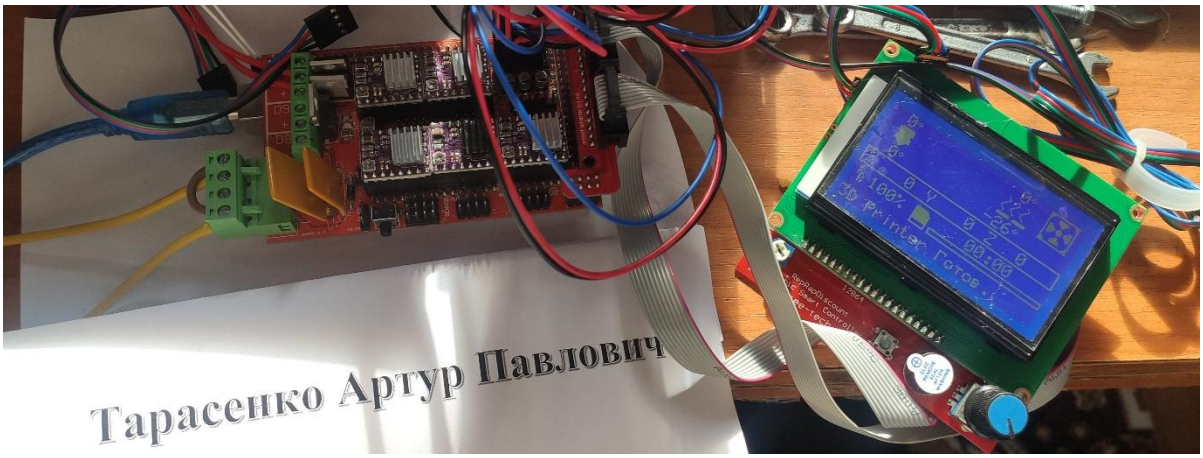

Рисунок 2.7. - Підключення LCD дисплея

g) Потім потрібно підключити всі кінцеві вимикачі як показано на рисунку 2.1 і розмістити їх у відповідних місцях на рамі принтера.

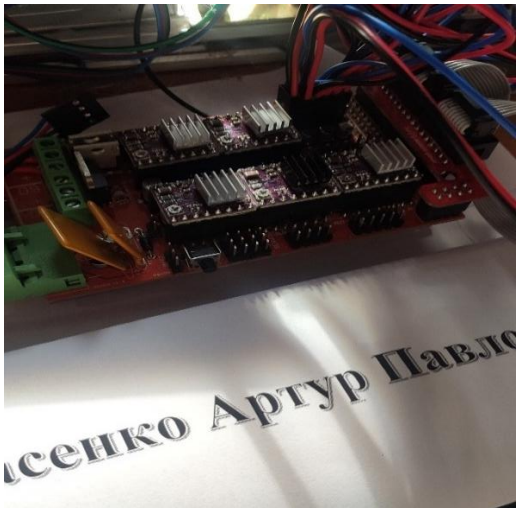

Рисунок 2.8. - Підключеня вимикачів

h) Плата нагрівання: Встановіть MK2A плату нагрівання на нижню частину рами принтера. Залежно від моделі плати можуть бути різні кріплення або способи кріплення, тому проконсультуйтеся з документацією плати нагрівання.

i) Блок живлення: Підключіть блок живлення JOTTA S-360-12 до RAMPS 1.4. Цей блок живлення забезпечує потрібну напругу для принтера. Дотримуйтеся схеми підключення та пам'ятайте про безпеку при роботі з електрикою.

j) Підключення інших компонентів: Якщо у вас є ще інші компоненти, такі як датчики рівняння столу або вентилятор охолодження, підключіть їх відповідно до схеми підключення зображеної на рисунку 2.1

k) Завершення: Після підключення всіх компонентів перевірте їх правильність підключення та орієнтацію. Переконайтеся, що всі кабелі забезпечують надійне з'єднання.

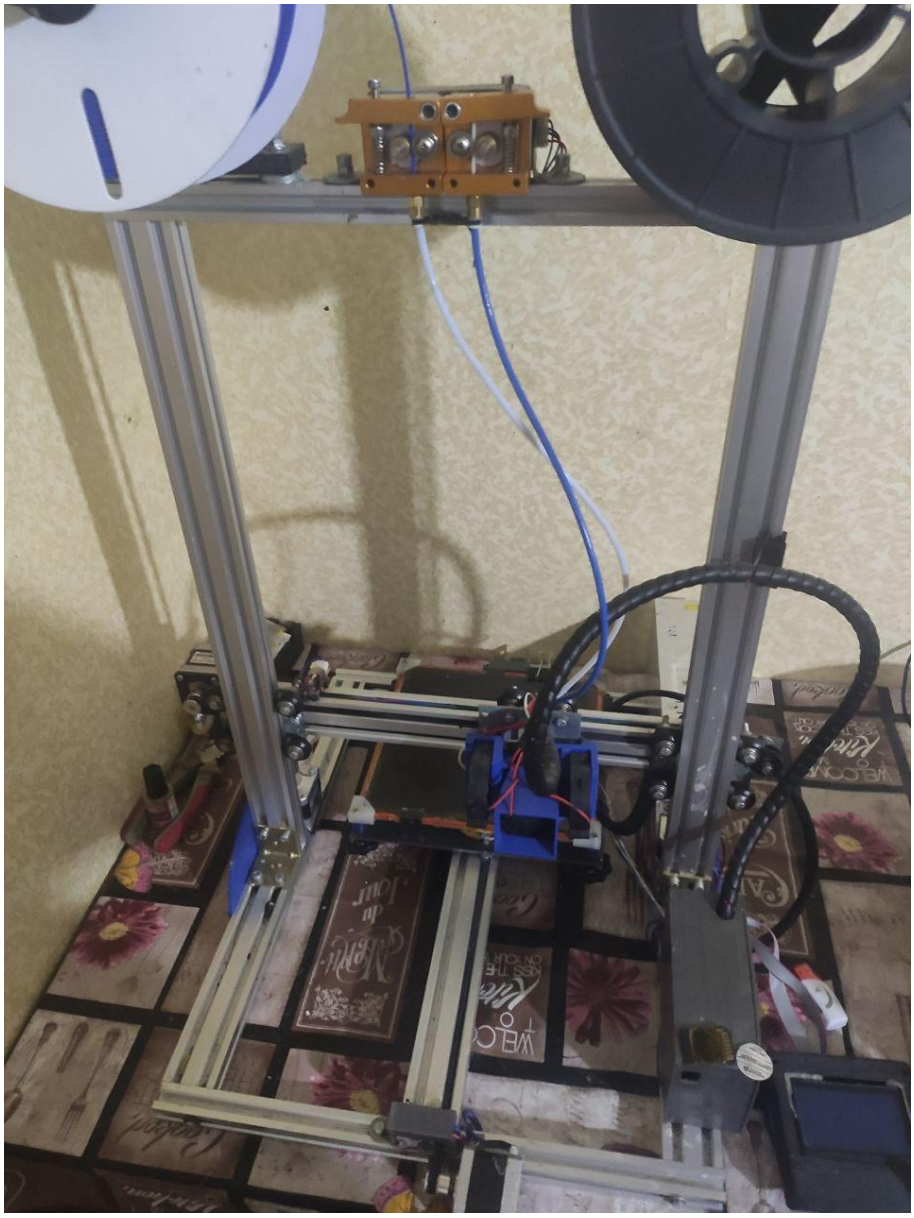

Рисунок 2.9. – 3д прінтер

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Опис середовища написання програмного коду

Інтегроване середовище розробки (IDE), що походить від англійського терміну "Integrated Development Environment", представляє собою програму або набір програм, які призначені для створення, налаштування, тестування та обслуговування програмного забезпечення. IDE характеризується розширеним функціоналом, що включає можливість редагування та компіляції вихідного коду, створення програмних ресурсів, роботу з базами даних і багато іншого. У рамках проекту Arduino було створено програмне забезпечення, яке відповідає основним вимогам типового IDE. Це не є потужним програмним забезпеченням, таким як Eclipse або NetBeans, але це проста та функціональна програма, яка дозволяє писати, компілювати та завантажувати програми до мікроконтролера. Проста структура Arduino IDE є його перевагою, оскільки вона дозволяє швидко ознайомитись з програмою та перейти до розробки додатків для Arduino. Незважаючи на свою простоту та інтуїтивне керування, варто звернути увагу на найважливіші елементи програми. Після запуску програми ви знайдете чотири основних функціональних елемента:

Меню програми, яке дозволяє управляти проектом, наприклад, створювати нові проекти, зберігати поточний проект, друкувати вихідний код на принтері та інші операції.

Панель швидкого доступу до найважливіших функцій, яка спрощує доступ до часто використовуваних опцій.

Редактор, в якому можна розміщувати код програми.

Панель повідомлень і статусу програми, яка відображає повідомлення про компіляцію та завантаження програми. Цікавою особливістю Arduino IDE є наявність вбудованого набору прикладів програм. Це дуже зручно, оскільки можна відразу перевірити приклади, завантаживши їх на мікроконтролер. За

потреби можна зберегти приклад та змінити його відповідно до власних потреб [12].

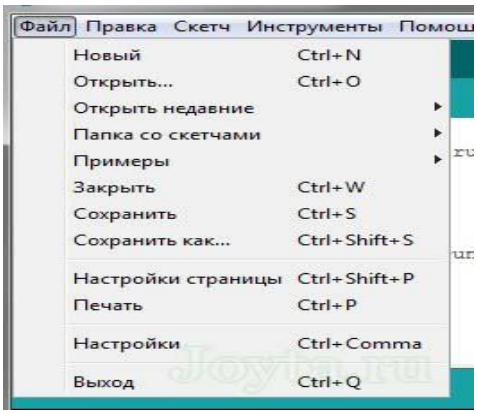

Рисунок 3.1 - Меню програми

Меню «Скетч» містить параметри для компіляції проекту і імпорту необхідних бібліотек.

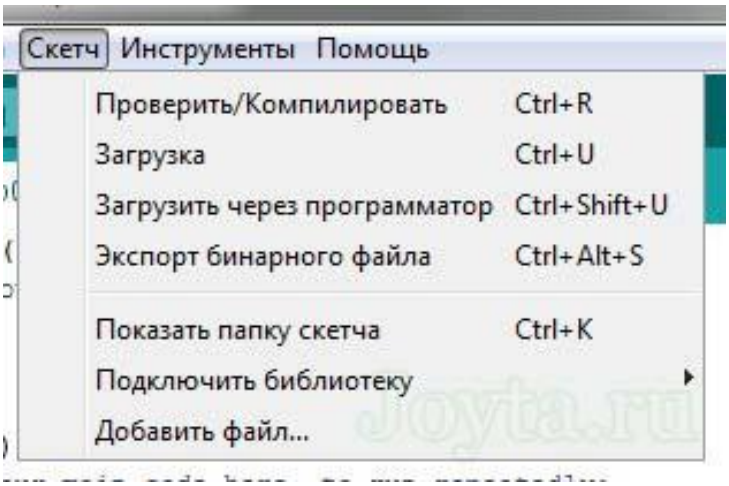

Рисунок 3.2 - Скетч

У IDE Arduino існує корисне і захоплююче меню під назвою "Інструменти". Воно включає в себе різні функції, які полегшують роботу з кодом, збереженням проектів та моніторингом послідовного порту (USB-порт на Arduino розглядається як звичайний послідовний порт). У цьому меню можна знайти функцію автоматичного форматування коду, яка допомагає зробити код більш організованим і зрозумілим, а також функцію архівування проекту, яка дозволяє зберегти проект у зручному форматі.

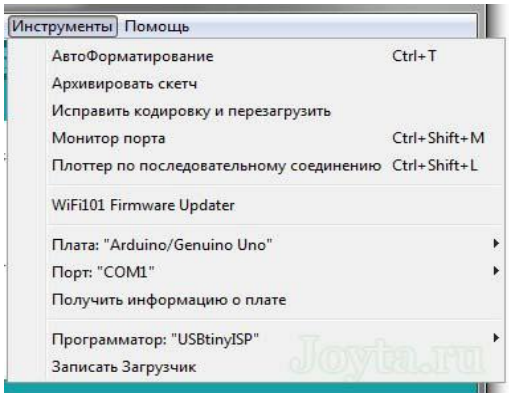

Рисунок 3.3 - Інструменти програми

Одним із найважливіших пунктів у меню "Інструменти" є можливість вибору відповідної плати, тобто вашої Arduino-системи, яка підключена до комп'ютера. У цьому списку містяться всі офіційні версії Arduino. Якщо ваш тип плати відсутній у списку, ви можете додати його, змінивши один із програмних файлів.

У меню "Інструменти" також можна встановити порт, до якого підключена плата Arduino. Зазвичай пакет Arduino IDE автоматично визначає порт, але іноді може знадобитися вручну встановити номер порту в налаштуваннях.

За допомогою Arduino IDE також можна завантажити Bootloader (завантажувач) на новий, чистий мікроконтролер Atmega, що дозволяє клонувати чіпи або просто замінити несправний мікроконтролер в Arduino.

Для зручної роботи з Arduino IDE використовується панель швидкого доступу, яка містить найважливіші кнопки. Це рішення спрощує роботу з пакетом IDE і надає прямий доступ до практично всіх необхідних параметрів під час написання та тестування програми.

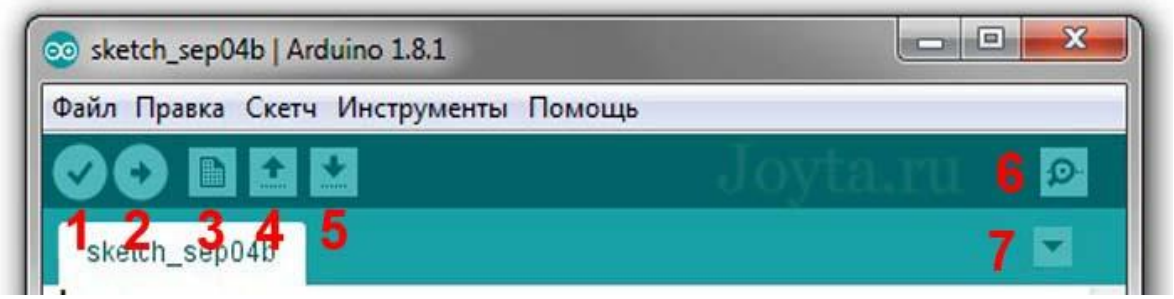

Рисунок 3.4 - Панель швидкого доступу

Панель швидкого доступу в Arduino IDE надає такі можливості (від ліва до права):

- Компіляція програми;
- Завантаження програми до мікроконтролера (після компіляції програмний код прошивається);
- Початок роботи над новим проектом;
- Відкриття існуючого проекту;
- Збереження проекту на диск;
- Увімкнення монітора послідовного порту.

Усі ці опції також доступні у меню програми.

Додатковим корисним елементом, що знаходиться під кнопкою увімкнення монітора послідовного порту, є меню для управління вкладками.

Найбільша частина вікна програми призначена для написання програмного коду. Редактор в Arduino IDE не має складних функцій, але включає в себе важливі елементи, які полегшують написання простих програм. Серед цих елементів можна виділити підсвічування синтаксису і блоків коду (дужки). Це можливості, які досить для простих проектів.

Останнім елементом програми є вікно повідомлень і статусу. Тут відображається інформація, яка допомагає користувачу знайти помилки у програмному коді і отримати підтвердження про компіляцію і завантаження програми до мікроконтролера.

Загалом, Arduino IDE є простим програмним пакетом, який дозволяє програмувати будь-яку відому плату Arduino, взаємодіяти з послідовним портом та легко управляти проектами. [12].

3.2 Програмне забезпечення для роботи пристрою

```
#include "Marlin.h"
#include "planner.h"
```

```
#include "temperature.h"
#include "ultralcd.h"
#include "ConfigurationStore.h"
void _EEPROM_writeData(int &pos, uint8_t* value, uint8_t size)
{ do { eeprom_write_byte((unsigned char*)pos, *value);
pos++;
value++;
  }while(--size);}
#define EEPROM_WRITE_VAR(pos, value) _EEPROM_writeData(pos,
(uint8_t*)&value, sizeof(value))
void _EEPROM_readData(int &pos, uint8_t* value, uint8_t size)
{do{   *value = eeprom_read_byte((unsigned char*)pos);
 pos++;
 value++;
}while(--size);}
#define EEPROM_READ_VAR(pos, value) _EEPROM_readData(pos,
(uint8_t*)&value, sizeof(value))
#define EEPROM_OFFSET 100
#define EEPROM_VERSION "V13"
#ifdef EEPROM_SETTINGS
void Config_StoreSettings() {
 char ver[4]= "000";
 int i=EEPROM_OFFSET;
 EEPROM_WRITE_VAR(i,ver); // invalidate data first
 EEPROM_WRITE_VAR(i,axis_steps_per_unit);
 EEPROM_WRITE_VAR(i,max_feedrate);
 EEPROM_WRITE_VAR(i,max_acceleration_units_per_sq_second);
 EEPROM_WRITE_VAR(i,acceleration);
 EEPROM_WRITE_VAR(i,retract_acceleration);
```

```
        EEPROM_WRITE_VAR(i,minimumfeedrate);
        EEPROM_WRITE_VAR(i,mintravelfeedrate);
        EEPROM_WRITE_VAR(i,minsegmenttime);
        EEPROM_WRITE_VAR(i,max_xy_jerk);
        EEPROM_WRITE_VAR(i,max_z_jerk);
        EEPROM_WRITE_VAR(i,max_e_jerk);
        EEPROM_WRITE_VAR(i,add_homing);
        #ifdef DELTA
        EEPROM_WRITE_VAR(i,endstop_adj);
        EEPROM_WRITE_VAR(i,delta_radius);
        EEPROM_WRITE_VAR(i,delta_diagonal_rod);
        EEPROM_WRITE_VAR(i,delta_segments_per_second);
        #endif
        #ifndef ULTIPANEL
        int plaPreheatHotendTemp = PLA_PREHEAT_HOTEND_TEMP,
plaPreheatHPBTemp = PLA_PREHEAT_HPB_TEMP, plaPreheatFanSpeed =
PLA_PREHEAT_FAN_SPEED;
        int absPreheatHotendTemp = ABS_PREHEAT_HOTEND_TEMP,
absPreheatHPBTemp = ABS_PREHEAT_HPB_TEMP, absPreheatFanSpeed =
ABS_PREHEAT_FAN_SPEED;
        #endif
        EEPROM_WRITE_VAR(i,plaPreheatHotendTemp);
        EEPROM_WRITE_VAR(i,plaPreheatHPBTemp);
        EEPROM_WRITE_VAR(i,plaPreheatFanSpeed);
        EEPROM_WRITE_VAR(i,absPreheatHotendTemp);
        EEPROM_WRITE_VAR(i,absPreheatHPBTemp);
        EEPROM_WRITE_VAR(i,absPreheatFanSpeed);
        EEPROM_WRITE_VAR(i,zprobe_zoffset);
        #ifdef PIDTEMP
```

```
  EEPROM_WRITE_VAR(i,Kp);
  EEPROM_WRITE_VAR(i,Ki);
  EEPROM_WRITE_VAR(i,Kd);
#else
        float dummy = 3000.0f;
  EEPROM_WRITE_VAR(i,dummy);
        dummy = 0.0f;
  EEPROM_WRITE_VAR(i,dummy);
  EEPROM_WRITE_VAR(i,dummy);
#endif
#ifndef DOGLCD
  int lcd_contrast = 32;
#endif
EEPROM_WRITE_VAR(i,lcd_contrast);
#ifdef SCARA
EEPROM_WRITE_VAR(i,axis_scaling);       // Add scaling for SCARA
#endif
#ifdef FWRETRACT
EEPROM_WRITE_VAR(i,autoretract_enabled);
EEPROM_WRITE_VAR(i,retract_length);
#if EXTRUDERS > 1
EEPROM_WRITE_VAR(i,retract_length_swap);
#endif
EEPROM_WRITE_VAR(i,retract_feedrate);
EEPROM_WRITE_VAR(i,retract_zlift);
EEPROM_WRITE_VAR(i,retract_recover_length);
#if EXTRUDERS > 1
EEPROM_WRITE_VAR(i,retract_recover_length_swap);
#endif
```

```
    EEPROM_WRITE_VAR(i,retract_recover_feedrate);
    #endif
    // Save filament sizes
    EEPROM_WRITE_VAR(i, volumetric_enabled);
    EEPROM_WRITE_VAR(i, filament_size[0]);
    #if EXTRUDERS > 1
    EEPROM_WRITE_VAR(i, filament_size[1]);
    #if EXTRUDERS > 2
    EEPROM_WRITE_VAR(i, filament_size[2]);
    #endif
    #endif
    char ver2[4]=EEPROM_VERSION;
    i=EEPROM_OFFSET;
    EEPROM_WRITE_VAR(i,ver2); // validate data
    SERIAL_ECHO_START;
    SERIAL_ECHOLNPGM("Settings Stored");
  }
  #endif //EEPROM_SETTINGS
  #ifndef DISABLE_M503
  void Config_PrintSettings()
  {  // Always have this function, even with EEPROM_SETTINGS disabled, the current values will be shown
    SERIAL_ECHO_START;
    SERIAL_ECHOLNPGM("Steps per unit:");
    SERIAL_ECHO_START;
    SERIAL_ECHOPAIR("  M92 X",axis_steps_per_unit[X_AXIS]);
    SERIAL_ECHOPAIR(" Y",axis_steps_per_unit[Y_AXIS]);
    SERIAL_ECHOPAIR(" Z",axis_steps_per_unit[Z_AXIS]);
    SERIAL_ECHOPAIR(" E",axis_steps_per_unit[E_AXIS]);
```

```
        SERIAL_ECHOLN("");
        SERIAL_ECHO_START;
      #ifdef SCARA
      SERIAL_ECHOLNPGM("Scaling factors:");
        SERIAL_ECHO_START;
        SERIAL_ECHOPAIR("  M365 X",axis_scaling[X_AXIS]);
        SERIAL_ECHOPAIR(" Y",axis_scaling[Y_AXIS]);
        SERIAL_ECHOPAIR(" Z",axis_scaling[Z_AXIS]);
        SERIAL_ECHOLN("");
        SERIAL_ECHO_START;
      #endif
        SERIAL_ECHOLNPGM("Maximum feedrates (mm/s):");
        SERIAL_ECHO_START;
        SERIAL_ECHOPAIR("  M203 X", max_feedrate[X_AXIS]);
        SERIAL_ECHOPAIR(" Y", max_feedrate[Y_AXIS]);
        SERIAL_ECHOPAIR(" Z", max_feedrate[Z_AXIS]);
        SERIAL_ECHOPAIR(" E", max_feedrate[E_AXIS]);
        SERIAL_ECHOLN("");
        SERIAL_ECHO_START;
        SERIAL_ECHOLNPGM("Maximum Acceleration (mm/s2):");
        SERIAL_ECHO_START;
        SERIAL_ECHOPAIR("  M201
X" ,max_acceleration_units_per_sq_second[X_AXIS] );
        SERIAL_ECHOPAIR(" Y" ,
max_acceleration_units_per_sq_second[Y_AXIS] );
        SERIAL_ECHOPAIR("
Z" ,max_acceleration_units_per_sq_second[Z_AXIS] );
        SERIAL_ECHOPAIR("
E" ,max_acceleration_units_per_sq_second[E_AXIS]);
```

```
        SERIAL_ECHOLN("");
        SERIAL_ECHO_START;
        SERIAL_ECHOLNPGM("Acceleration: S=acceleration, T=retract
acceleration");
        SERIAL_ECHO_START;
        SERIAL_ECHOPAIR("  M204 S",acceleration );
        SERIAL_ECHOPAIR(" T" ,retract_acceleration);
        SERIAL_ECHOLN("");
        SERIAL_ECHO_START;
        SERIAL_ECHOLNPGM("Advanced variables: S=Min feedrate (mm/s),
T=Min travel feedrate (mm/s), B=minimum segment time (ms), X=maximum XY
jerk (mm/s),  Z=maximum Z jerk (mm/s),  E=maximum E jerk (mm/s)");
        SERIAL_ECHO_START;
        SERIAL_ECHOPAIR("  M205 S",minimumfeedrate );
        SERIAL_ECHOPAIR(" T" ,mintravelfeedrate );
        SERIAL_ECHOPAIR(" B" ,minsegmenttime );
        SERIAL_ECHOPAIR(" X" ,max_xy_jerk );
        SERIAL_ECHOPAIR(" Z" ,max_z_jerk);
        SERIAL_ECHOPAIR(" E" ,max_e_jerk);
        SERIAL_ECHOLN("");
        SERIAL_ECHO_START;
        SERIAL_ECHOLNPGM("Home offset (mm):");
        SERIAL_ECHO_START;
        SERIAL_ECHOPAIR("  M206 X",add_homing[X_AXIS] );
        SERIAL_ECHOPAIR(" Y" ,add_homing[Y_AXIS] );
        SERIAL_ECHOPAIR(" Z" ,add_homing[Z_AXIS] );
        SERIAL_ECHOLN("");
      #ifdef DELTA
        SERIAL_ECHO_START;
```

```
        SERIAL_ECHOLNPGM("Endstop adjustement (mm):");
        SERIAL_ECHO_START;
        SERIAL_ECHOPAIR("  M666 X",endstop_adj[X_AXIS] );
        SERIAL_ECHOPAIR(" Y" ,endstop_adj[Y_AXIS] );
        SERIAL_ECHOPAIR(" Z" ,endstop_adj[Z_AXIS] );
      SERIAL_ECHOLN("");
      SERIAL_ECHO_START;
      SERIAL_ECHOLNPGM("Delta settings: L=delta_diagonal_rod,
R=delta_radius, S=delta_segments_per_second");
      SERIAL_ECHO_START;
      SERIAL_ECHOPAIR("  M665 L",delta_diagonal_rod );
      SERIAL_ECHOPAIR(" R" ,delta_radius );
      SERIAL_ECHOPAIR(" S" ,delta_segments_per_second );
      SERIAL_ECHOLN("");
    #endif
    #ifdef PIDTEMP
      SERIAL_ECHO_START;
      SERIAL_ECHOLNPGM("PID settings:");
      SERIAL_ECHO_START;
      SERIAL_ECHOPAIR("   M301 P",Kp);
      SERIAL_ECHOPAIR(" I" ,unscalePID_i(Ki));
      SERIAL_ECHOPAIR(" D" ,unscalePID_d(Kd));
      SERIAL_ECHOLN("");
    #endif
    #ifdef FWRETRACT
      SERIAL_ECHO_START;
      SERIAL_ECHOLNPGM("Retract: S=Length (mm) F:Speed (mm/m) Z: ZLift
(mm)");
      SERIAL_ECHO_START;
```

```
        SERIAL_ECHOPAIR("   M207 S",retract_length);
        SERIAL_ECHOPAIR(" F" ,retract_feedrate*60);
        SERIAL_ECHOPAIR(" Z" ,retract_zlift);
        SERIAL_ECHOLN("");
        SERIAL_ECHO_START;
        SERIAL_ECHOLNPGM("Recover: S=Extra length (mm) F:Speed (mm/m)");
        SERIAL_ECHO_START;
        SERIAL_ECHOPAIR("   M208 S",retract_recover_length);
        SERIAL_ECHOPAIR(" F", retract_recover_feedrate*60);
       SERIAL_ECHOLN("");
        SERIAL_ECHO_START;
        SERIAL_ECHOLNPGM("Auto-Retract: S=0 to disable, 1 to interpret
extrude-only moves as retracts or recoveries");
        SERIAL_ECHO_START;
        SERIAL_ECHOPAIR("   M209 S", (unsigned long)(autoretract_enabled ? 1 :
0));
        SERIAL_ECHOLN("");
      #if EXTRUDERS > 1
        SERIAL_ECHO_START;
        SERIAL_ECHOLNPGM("Multi-extruder settings:");
        SERIAL_ECHO_START;
        SERIAL_ECHOPAIR("   Swap retract length (mm):    ", retract_length_swap);
        SERIAL_ECHOLN("");
        SERIAL_ECHO_START;
        SERIAL_ECHOPAIR("   Swap rec. addl. length (mm): ",
retract_recover_length_swap);
        SERIAL_ECHOLN("");
      #endif
       SERIAL_ECHO_START;
```

```
        if (volumetric_enabled) {
        SERIAL_ECHOLNPGM("Filament settings:");
        SERIAL_ECHO_START;
        SERIAL_ECHOPAIR("   M200 D", filament_size[0]);
        SERIAL_ECHOLN("");
       #if EXTRUDERS > 1
        SERIAL_ECHO_START;
        SERIAL_ECHOPAIR("   M200 T1 D", filament_size[1]);
        SERIAL_ECHOLN("");
       #if EXTRUDERS > 2
        SERIAL_ECHO_START;
        SERIAL_ECHOPAIR("   M200 T2 D", filament_size[2]);
        SERIAL_ECHOLN("");
        #endif
        #endif
        } else {SERIAL_ECHOLNPGM("Filament settings: Disabled"); }
       #endif}
       #endif
       #ifdef EEPROM_SETTINGS
       void Config_RetrieveSettings()
       {  int i=EEPROM_OFFSET;
       char stored_ver[4];
       char ver[4]=EEPROM_VERSION;
       EEPROM_READ_VAR(i,stored_ver); //read stored version
       //  SERIAL_ECHOLN("Version: [" << ver << "] Stored version: [" <<
stored_ver << "]");
       if (strncmp(ver,stored_ver,3) == 0){
       // version number match
       EEPROM_READ_VAR(i,axis_steps_per_unit);
```

```
    EEPROM_READ_VAR(i,max_feedrate);
    EEPROM_READ_VAR(i,max_acceleration_units_per_sq_second);
    // steps per sq second need to be updated to agree with the units per sq second (as they are what is used in the planner)
    reset_acceleration_rates();
    EEPROM_READ_VAR(i,acceleration);
    EEPROM_READ_VAR(i,retract_acceleration);
    EEPROM_READ_VAR(i,minimumfeedrate);
    EEPROM_READ_VAR(i,mintravelfeedrate);
    EEPROM_READ_VAR(i,minsegmenttime);
    EEPROM_READ_VAR(i,max_xy_jerk);
    EEPROM_READ_VAR(i,max_z_jerk);
    EEPROM_READ_VAR(i,max_e_jerk);
    EEPROM_READ_VAR(i,add_homing);
    #ifdef DELTA
    EEPROM_READ_VAR(i,endstop_adj);
    EEPROM_READ_VAR(i,delta_radius);
    EEPROM_READ_VAR(i,delta_diagonal_rod);
    EEPROM_READ_VAR(i,delta_segments_per_second);
    #endif
    #ifndef ULTIPANEL
    int plaPreheatHotendTemp, plaPreheatHPBTemp, plaPreheatFanSpeed;
    int absPreheatHotendTemp, absPreheatHPBTemp, absPreheatFanSpeed;
    #endif
    EEPROM_READ_VAR(i,plaPreheatHotendTemp);
    EEPROM_READ_VAR(i,plaPreheatHPBTemp);
    EEPROM_READ_VAR(i,plaPreheatFanSpeed);
    EEPROM_READ_VAR(i,absPreheatHotendTemp);
    EEPROM_READ_VAR(i,absPreheatHPBTemp);
```

```
EEPROM_READ_VAR(i,absPreheatFanSpeed);
EEPROM_READ_VAR(i,zprobe_zoffset);
#ifndef PIDTEMP
float Kp,Ki,Kd;
#endif
// do not need to scale PID values as the values in EEPROM are already scaled
EEPROM_READ_VAR(i,Kp);
EEPROM_READ_VAR(i,Ki);
EEPROM_READ_VAR(i,Kd);
#ifndef DOGLCD
int lcd_contrast;
 #endif
EEPROM_READ_VAR(i,lcd_contrast);
#ifdef SCARA
EEPROM_READ_VAR(i,axis_scaling);
#endif
#ifdef FWRETRACT
EEPROM_READ_VAR(i,autoretract_enabled);
EEPROM_READ_VAR(i,retract_length);
#if EXTRUDERS > 1
EEPROM_READ_VAR(i,retract_length_swap);
#endif
EEPROM_READ_VAR(i,retract_feedrate);
EEPROM_READ_VAR(i,retract_zlift);
EEPROM_READ_VAR(i,retract_recover_length);
#if EXTRUDERS > 1
EEPROM_READ_VAR(i,retract_recover_length_swap);
#endif
EEPROM_READ_VAR(i,retract_recover_feedrate);
```

```
#endif
EEPROM_READ_VAR(i, volumetric_enabled);
EEPROM_READ_VAR(i, filament_size[0]);
#if EXTRUDERS > 1
EEPROM_READ_VAR(i, filament_size[1]);
#if EXTRUDERS > 2
EEPROM_READ_VAR(i, filament_size[2]);
#endif
#endif
calculate_volumetric_multipliers();
// Call updatePID (similar to when we have processed M301)
updatePID();
SERIAL_ECHO_START;
SERIAL_ECHOLNPGM("Stored settings retrieved");}
 else
{
Config_ResetDefault();}
#ifdef EEPROM_CHITCHAT
Config_PrintSettings();
#endif
}
#endif

void Config_ResetDefault(){
float tmp1[]=DEFAULT_AXIS_STEPS_PER_UNIT;
float tmp2[]=DEFAULT_MAX_FEEDRATE;
long tmp3[]=DEFAULT_MAX_ACCELERATION;
for (short i=0;i<4;i++) {
axis_steps_per_unit[i]=tmp1[i];
```

```
max_feedrate[i]=tmp2[i];
max_acceleration_units_per_sq_second[i]=tmp3[i];
#ifdef SCARA
  axis_scaling[i]=1;
#endif
    }
// steps per sq second need to be updated to agree with the units per sq second
 reset_acceleration_rates();
acceleration=DEFAULT_ACCELERATION;
retract_acceleration=DEFAULT_RETRACT_ACCELERATION;
minimumfeedrate=DEFAULT_MINIMUMFEEDRATE;
minsegmenttime=DEFAULT_MINSEGMENTTIME;
mintravelfeedrate=DEFAULT_MINTRAVELFEEDRATE;
max_xy_jerk=DEFAULT_XYJERK;
max_z_jerk=DEFAULT_ZJERK;
max_e_jerk=DEFAULT_EJERK;
add_homing[X_AXIS] = add_homing[Y_AXIS] = add_homing[Z_AXIS] = 0;
#ifdef DELTA
endstop_adj[X_AXIS] = endstop_adj[Y_AXIS] = endstop_adj[Z_AXIS] = 0;
delta_radius= DELTA_RADIUS;
delta_diagonal_rod= DELTA_DIAGONAL_ROD;
delta_segments_per_second= DELTA_SEGMENTS_PER_SECOND;
recalc_delta_settings(delta_radius, delta_diagonal_rod);
#endif
#ifdef ULTIPANEL
plaPreheatHotendTemp = PLA_PREHEAT_HOTEND_TEMP;
plaPreheatHPBTemp = PLA_PREHEAT_HPB_TEMP;
plaPreheatFanSpeed = PLA_PREHEAT_FAN_SPEED;
absPreheatHotendTemp = ABS_PREHEAT_HOTEND_TEMP;
```

```
absPreheatHPBTemp = ABS_PREHEAT_HPB_TEMP;
absPreheatFanSpeed = ABS_PREHEAT_FAN_SPEED;
#endif
#ifdef ENABLE_AUTO_BED_LEVELING
zprobe_zoffset = -Z_PROBE_OFFSET_FROM_EXTRUDER;
#endif
#ifdef DOGLCD
lcd_contrast = DEFAULT_LCD_CONTRAST;
#endif
#ifdef PIDTEMP
 Kp = DEFAULT_Kp;
 Ki = scalePID_i(DEFAULT_Ki);
 Kd = scalePID_d(DEFAULT_Kd);
#ifdef PID_ADD_EXTRUSION_RATE
Kc = DEFAULT_Kc;
#endif//PID_ADD_EXTRUSION_RATE
#endif//PIDTEMP
#ifdef FWRETRACT
autoretract_enabled = false;
retract_length = RETRACT_LENGTH;
#if EXTRUDERS > 1
retract_length_swap = RETRACT_LENGTH_SWAP;
#endif
retract_feedrate = RETRACT_FEEDRATE;
retract_zlift = RETRACT_ZLIFT;
retract_recover_length = RETRACT_RECOVER_LENGTH;
#if EXTRUDERS > 1
retract_recover_length_swap = RETRACT_RECOVER_LENGTH_SWAP;
#endif
```

```
retract_recover_feedrate = RETRACT_RECOVER_FEEDRATE;
#endif
volumetric_enabled = false;
filament_size[0] = DEFAULT_NOMINAL_FILAMENT_DIA;
#if EXTRUDERS > 1
filament_size[1] = DEFAULT_NOMINAL_FILAMENT_DIA;
#if EXTRUDERS > 2
filament_size[2] = DEFAULT_NOMINAL_FILAMENT_DIA;
#endif
#endif
calculate_volumetric_multipliers();
SERIAL_ECHO_START;
SERIAL_ECHOLNPGM("Hardcoded Default Settings Loaded");
```

3.2 Порядок роботи з пристроєм

Перед початком роботи з 3D принтером, для самостійної зборки необхідно «залити» і налаштувати прошивку. Прошивка являє собою програмний код, основними завданнями якого є: зчитування і відтворення G-code, управління принтером через різні інтерфейси, вивід інформації про процес друку. Іншими словами, прошивка необхідна, щоб залізо і набір електроніки «ожили» і можна було ними управляти. Заливається прошивка на плату управління. У різних 3D принтерів різноманітні плати управління, відповідно, прошивки теж різні.

У 3D принтерах Prusa i3 Steel використовується зв'язка плат Arduino Mega 2560 і Ramps 1.4, тому докладно розглянемо і розберемо настройки підходящої для них прошивки, Marlin.

Дана прошивка є однією з найпопулярніших, в тому числі, тому що розробники регулярно додають в неї нові можливості: автоматичне

регулювання зазору, датчик закінчення прутка і багато іншого. Крім того, ця прошивка абсолютно безкоштовна, і її можна завантажити з офіційного сайту.

Остання версія прошивки Marlin викладена на офіційному сайті розробника https://github.com/MarlinFirmware/Marlin . Також на сайті присутні багато різних версій, але ми рекомендуємо завантажувати останню версію, позначену як Latest release.

Розглянемо установки програми Arduino IDE. Після того, як ви завантажили прошивку, потрібно її відредагувати і записати на мікроконтролер плати управління (Arduino mega 2560). Для цих цілей знадобитися програма Arduino IDE, скачати яку можна безкоштовно з офіційного сайту Arduino. Дана програма Arduino IDE регулярно оновлюється і можливий такий варіант, що при завантаженні прошивки на плату, з новими версіями Arduino IDE можуть виникнути проблеми, а саме будуть з'являтися помилки, і ви не зможете записати прошивку в мікроконтролер. Тому, при виникненні проблем, спробуйте завантажити старішу версію програми, наприклад версію 1.6.0.

Розглянемо редагування прошивки Marlin. Загрузимо безпосередньо саму прошивку Marlin і програму Arduino IDE, за допомогою якої можна редагувати. Відкрийте папку з прошивкою "Marlin", знайдіть файл "Marlin" з розширенням .ino.

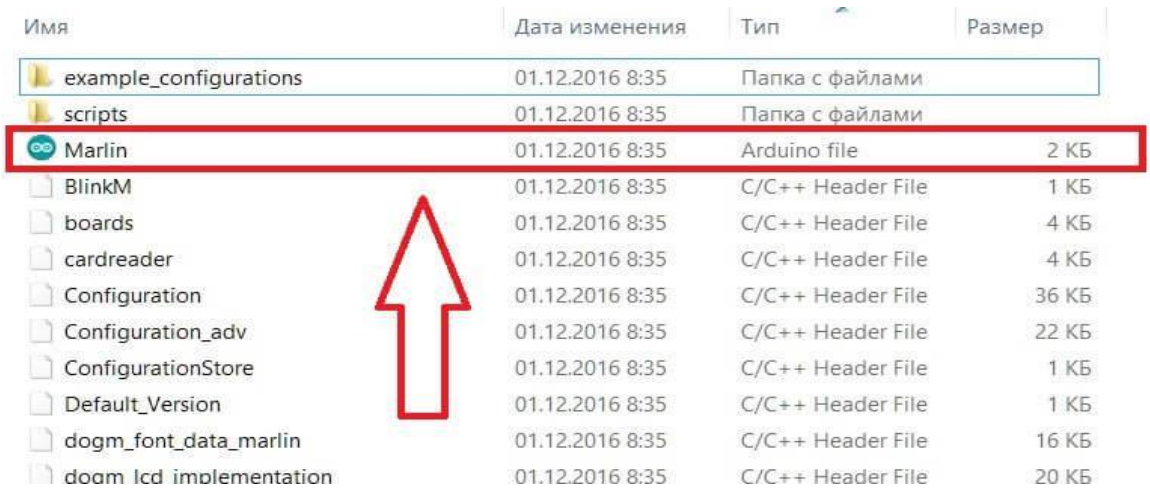

Рисунок 3.5. - завантаження прошивки Marlin

Вгорі вікна програми перебувати багато вкладок, в кожній з яких розташовуються код, від якого і залежить робота 3D принтера. Буде потрібна тільки вкладка "Configuration.h"

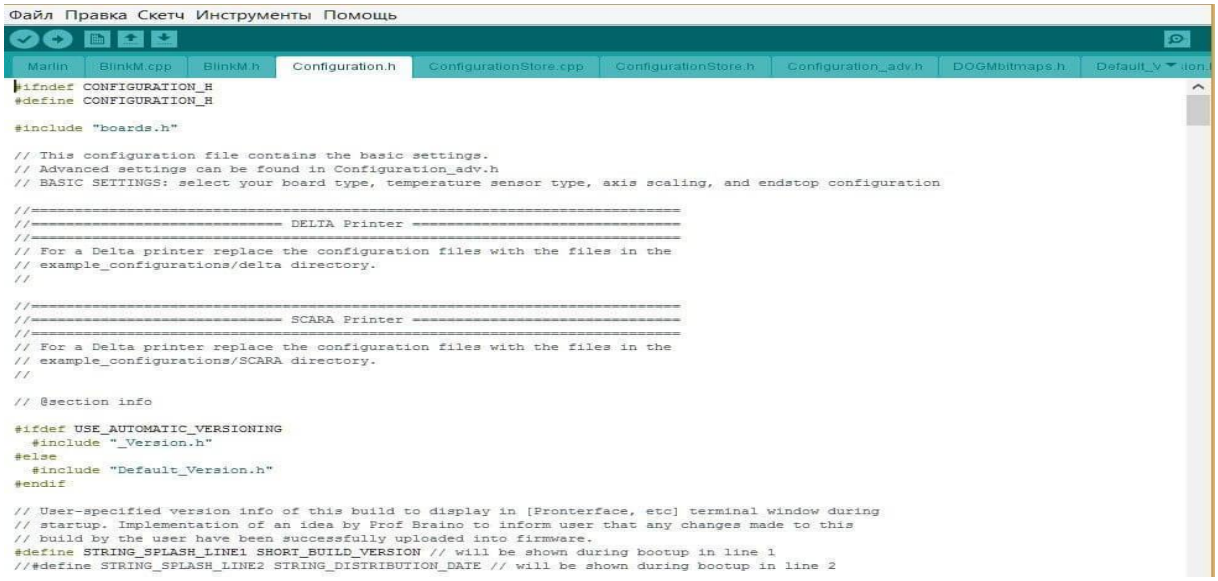

Рисунок 3.6. - вкладка налаштувань "Configuration.h"

Цей конфігураційний файл, який містить основні настройки. Саме в цій вкладці необхідно провести основні зміни.

Розглянемо встановлення необхідної швидкості в бодах. Перше, що треба щось міняти - швидкість в бодах. За замовчуванням швидкість варто 250000 (47 рядок коду) [4].

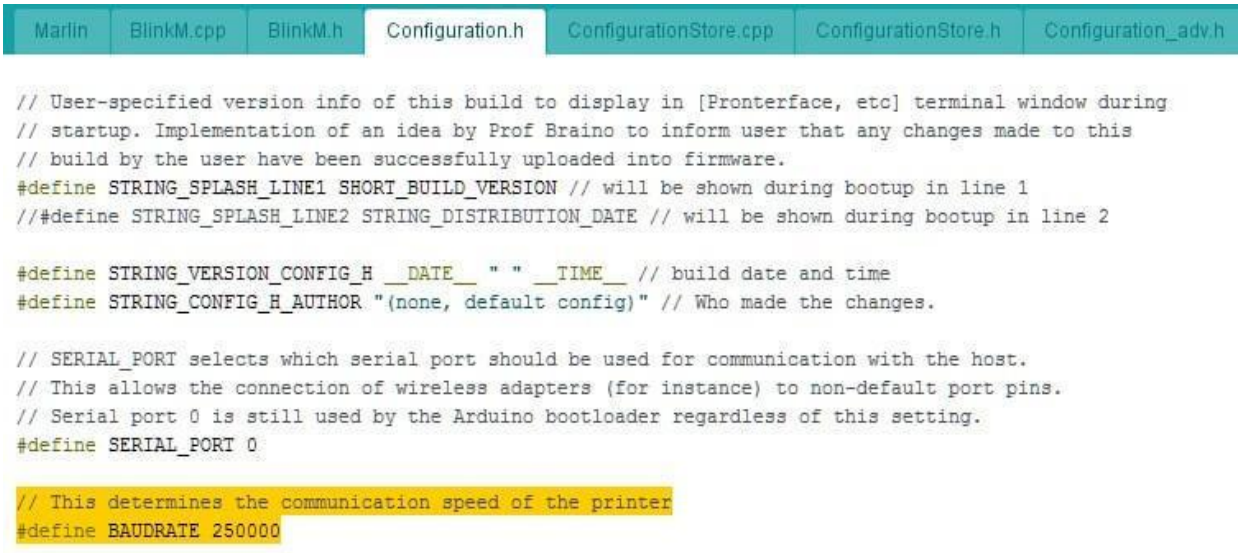

Рисунок 3.7. - Встановлення необхідної швидкості в бодах

Для кожної плати виробник рекомендує свої швидкості, тому для зв'язки Arduino mega 2560 і Ramps 1.4 необхідно поставити 115200, тобто ділянку коду у нас повинен прийняти такий вигляд:

// This determines the communication speed of the printer #define BAUDRATE 115200

Якщо ви використовуєте плату Gen V1.4 , то швидкість повинна бути 250000.

Вибираємо керуючу плату

Після установки швидкості в бодах, необхідно вказати використовувану плату управління (55 рядок коду).

#ifndef MOTHERBOARD

#define MOTHERBOARD BOARD_ULTIMAKER

#endif

```
// This enables the serial port associated to the Bluetooth interface
//#define BTENABLED              // Enable BT interface on AT90USB devices

// The following define selects which electronics board you have.
// Please choose the name from boards.h that matches your setup
#ifndef MOTHERBOARD
 #define MOTHERBOARD BOARD_ULTIMAKER
#endif

// Define this to set a custom name for your generic Mendel,
// #define CUSTOM_MENDEL_NAME "This Mendel"

// Define this to set a unique identifier for this printer, (Used by some programs to differentiate between machines)
// You can use an online service to generate a random UUID. (eg http://www.uuidgenerator.net/version4)
// #define MACHINE_UUID "00000000-0000-0000-0000-000000000000"
```

Рисунок 3.8. - Встановлення використовуваної плати управління

За замовчуванням плата 3D принтера Ultimaker - BOARD_ULTIMAKER, тому необхідно поміняти плату. Весь список плат знаходиться у вкладці "BOARDS_H"

Там наданий величезний список різних плат, але вам необхідні тільки такі:

#define BOARD_RAMPS_13_EFB 33 // RAMPS 1.3 / 1.4 (Power outputs: Extruder, Fan, Bed)

#define BOARD_RAMPS_13_EEB 34 // RAMPS 1.3 / 1.4 (Power outputs: Extruder0, Extruder1, Bed)

#define BOARD_RAMPS_13_EFF 35 // RAMPS 1.3 / 1.4 (Power outputs: Extruder, Fan, Fan)

#define BOARD_RAMPS_13_EEF 36 // RAMPS 1.3 / 1.4 (Power outputs: Extruder0, Extruder1, Fan)

Ці плати відносяться до Arduino mega 2560 і Ramps 1.4. Залежно від модифікації вашого 3D принтера, необхідно вибрати відповідну плату. Наприклад зв'язка 2 екструдера + обдув робочої області + нагрівальний стіл відповідає платі BOARD_RAMPS_13_EEB

Назва плати необхідно скопіювати і замінити на вкладці "Configuration.h", міняємо наступні рядки:

```
// The following define selects which electronics board you have.
// Please choose the name from boards.h that matches your setup
#ifndef MOTHERBOARD
  #define MOTHERBOARD BOARD_RAMPS_13_EFB
#endif
```

Рисунок 3.9 - Вибір плати для управління

При налаштуванні придумайте назву своєму 3D принтера і вкажіть це в прошивці. Назва принтера відображається на його LCD дисплеї, така можливість точно передбачена на такому дисплеї.

Знайдіть рядки: (59 рядок)

// #define CUSTOM_MENDEL_NAME "This Mendel"

Перед #define стоять "//" - це означає, що ці рядки не використовуються в коді, а служать в якості пояснень. Щоб зробити її активною рядок, необхідно розкоментувати рядок, приберіть // перед рядком.

Змініть назву за замовчуванням "This Mendel" на ваше назву 3D принтера, наприклад, "P3Steel". Отримуємо наступні:

```
// Define this to set a custom name for your generic Mendel,
 #define CUSTOM_MENDEL_NAME "P3Steel"
```

Рисунок 3.10. -Задаємо назву принтера

У вище наведених налаштуваннях прошивки було згадано про наявність одного нагрівального елемента та нагрівального столу в 3D-принтері, що означає наявність двох елементів, температуру яких потрібно регулювати. Контроль температури здійснюється за допомогою термісторів - датчиків

температури. Існує великий вибір різних типів термісторів з різними характеристиками, тому в прошивці необхідно вказати конкретний тип термістора, який використовується в вашому принтері. Це потрібно, щоб в подальшому принтер показував вірну температуру. У прошивці знайдіть список підтримуваних термісторов:

//// Temperature sensor settings:

// -2 is thermocouple with MAX6675 (only for sensor 0)

// -1 is thermocouple with AD595

// 0 is not used

//1 : 100kΩ EPCOS - Best choice for EPCOS thermistors

// 2 is 200k thermistor - ATC Semitec 204GT-2 (4.7k pullup)

// 3 Mendel-parts thermistor

//4 : 10kΩ Generic Thermistor !! DO NOT use for a hotend - it gives bad resolution at high temp.

списку знайдіть свій, запам'ятайте цифру зліва. Як правило, багато хто використовує китайський термістор 100 кОм, для нього підходить термістор під номером "1" [5].

// 1 is 100k thermistor - best choice for EPCOS 100k (4.7k pullup)

Внесіть зміни в потрібному місці (рядки 115-118)

#define TEMP_SENSOR_0 -1

#define TEMP_SENSOR_1 -1

#define TEMP_SENSOR_2 0

#define TEMP_SENSOR_BED 0

За замовчуванням в прошивці активовані два перших термистора:

TEMP_SENSOR_0 - відповідає за термистор першого екструдера

TEMP_SENSOR_1 - відповідає за термистор другого екструдера

TEMP_SENSOR_BED - відповідає за термистор столу

Поміняй рядки і отримаєш наступне:

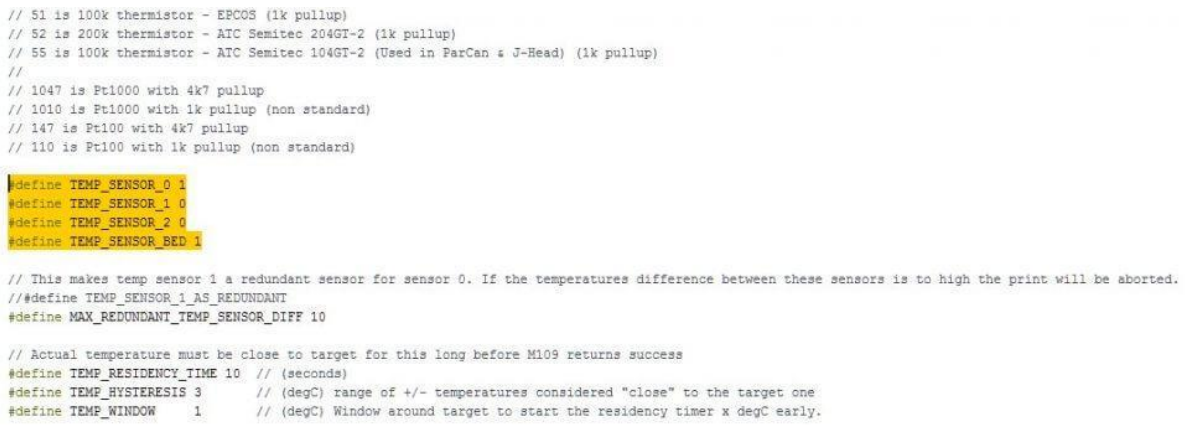

Рисунок 3.11. -Налаштування термісторів

TEMP_SENSOR_1 і TEMP_SENSOR_2 не використовуються і тому їм присвоєно значення "0".

Обмеження максимальної допустимої температури

Для обмеження максимальної температури необхідні наступні рядки (140-143)

#define HEATER_0_MAXTEMP 280

#define HEATER_1_MAXTEMP 280

#define HEATER_2_MAXTEMP 280

#define BED_MAXTEMP 140

Числа стоять праворуч, а саме 280 і 140 - це максимальні температури екструдера і нагрівального столу відповідно.

Коли температура перевищує максимальну температуру , нагрівач буде вимкнений. Ця функція існує для того, щоб захисту екструдера від випадкового перегріву. Якщо ви використовуєте хотенд з тефлоном всередині, то рекомендуємо обмежити температурою 265 градусів [4].

Обмеження мінімальної температури

Також, у прошивці за замовчуванням встановлено обмеження мінімальної температури екструдера на рівні 170 градусів. Це означає, що якщо температура екструдера буде нижче 170 градусів, то двигун екструдера не буде працювати, і пластик не буде подаватися. Захист від проштовхування не прогрітий пластика (рядок 230).

#define EXTRUDE_MINTEMP 170

Якщо хочете відключити цю функцію, то перед рядком поставте "//"

Розглянемо налаштування логіки роботи вимикачів. В першу чергу потрібно звернути увагу на принцип роботи вимикачів. У прошивці необхідно правильно вказати логіку роботи вимикачів. Знайдіть такі рядки (301-306)

const bool X_MIN_ENDSTOP_INVERTING = true; // Встановіть значення "true" для інвертування логіки кінцевого перемикача.

const bool У_MIN_ENDSTOP_INVERTING = true; // Встановіть значення "true" для інвертування логіки кінцевого перемикача.

const bool Z_MIN_ENDSTOP_INVERTING = true; // Встановіть значення "true" для інвертування логіки кінцевого перемикача.

const bool X_MAX_ENDSTOP_INVERTING = true; // Встановіть значення "true" для інвертування логіки кінцевого перемикача.

const bool У_MAX_ENDSTOP_INVERTING = true; // Встановіть значення "true" для інвертування логіки кінцевого перемикача.

const bool Z_MAX_ENDSTOP_INVERTING = true; // Встановіть значення "true" для інвертування логіки кінцевого перемикача.

Якщо у вас механічні концовики, то при спрацьовування ланцюг замикається, навпаки кожного рядка відповідної осі поставте значення "true". Якщо ви використовуєте оптичні концовики, то при спрацьовуванні ланцюг розмикається, навпаки кожного рядка відповідної осі поставте значення "false" [5].

За замовчуванням в прошивці навпроти кожного вимикача стоять значення "true", що відповідають механічним вимикачам. Після настройки роботи вимикачів можна перевірити командою M119 в консолі. У відповідь повинен прийти текст:

- x_min: open - вимикач не спрацював;
- x_min: TRIGGERED - вимикач спрацював.

Установка положення "HOME" - будинок

У прошивці підтримуються 3 пари вимикачів: для кожної осі X, У і Z по два вимикача min і max. Як правило, ставляться вимикачі тільки для мінімального положення кожної осі, а максимальне задається в прошивці.

Положення будинок (початкове положення), буде перебувати в мінімальних положеннях вимикача і це задається в прошивці: (рядки 337-339)

#define X_HOME_DIR -1

#define У_HOME_DIR -1

#define Z_HOME_DIR -1

1 = MAX, -1 = MIN

Зміни напрямку обертання двигунів

При складанні 3D принтера, а саме при підключення крокових двигунів до плати, можлива така ситуація: коли ви все налаштували і підключили, при натисканні "home" (будинок), каретка однієї з осей їде в іншу сторону (не вимикача), тоді необхідно перевернути конектор крокової двигуна на 180 градусів або поміняти значення в прошивці:

#define INVERT_X_DIR true // for Mendel set to false, for Orca set to true

#define INVERT_У_DIR false // for Mendel set to true, for Orca set to false

#define INVERT_Z_DIR true // for Mendel set to false, for Orca set to true

#define INVERT_E0_DIR false // for direct drive extruder v9 set to true, for geared extruder set to false

#define INVERT_E1_DIR false // for direct drive extruder v9 set to true, for geared extruder set to false

#define INVERT_E2_DIR false // for direct drive extruder v9 set to true, for geared extruder set to false

Наприклад, якщо у вас каретка осі У в іншу сторону, то необхідно знайти рядок

#define INVERT_У_DIR false // for Mendel set to true, for Orca set to false

і поміняти "false" на "true". І так з кожною віссю і екструдером.

Установка габаритів переміщення

Щоб 3D принтер визначав робочу область, необхідно вказати розміри друкованої області в прошивці:

#define X_MAX_POS 200

#define X_MIN_POS 0

#define У_MAX_POS 200

#define У_MIN_POS 0

#define Z_MAX_POS 200

#define Z_MIN_POS 0

Навпроти кожного рядка вкажіть відповідні габарити, за замовчуванням робоча область задана 200x200x200 мм

Налаштування кроків переміщення по осях

Вказівка кількості кроків крокових двигунів - одна з головних параметрів прошивки (рядок 490):

#define DEFAULT_AXIS_STEPS_PER_UNIT {78.7402,78.7402,200.0 * 8 / 3,760 * 1.1} // default steps per unit for Ultimaker

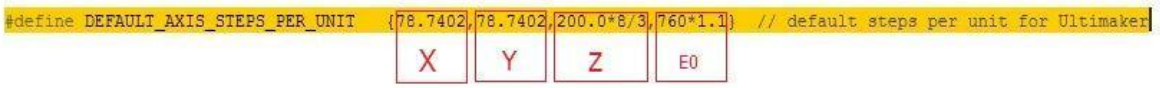

Рисунок 3.12. - Налаштування крокових двигунів

У дужках, розділених комою, вказано кількість кроків, які кроковий двигун повинен виконати, щоб каретка перемістилася на 1 мм.

Налаштування подачі матеріалу екструдера залежить від коефіцієнта передачі та діаметра шестерні. Кількість кроків, які кроковий двигун екструдера повинен виконати для подачі 1 мм пластику, встановлюється експериментально після першої програмної прошивки 3D-принтера [4].

Для зниження обмеження мінімальної температури сопла до 5 градусів необхідно відкрутити сопло.

#define EXTRUDE_MINTEMP 5

Тепер екструдер буде працювати при холодному соплі. Не змінюючи налаштувань екструдера, натисніть прогнати пластик на 100 мм. Виміряйте довжину прутка пройшов через екструдер лінійкою або штангенциркулем.

Підбираючи настройку екструдера добийтеся точної цифри на розумної довжині прутка, наприклад 200 мм. Після настройки поверніть обмеження мінімальної температури:

#define EXTRUDE_MINTEMP 170

Обмеження максимальної швидкості переміщення по осях

#define DEFAULT_MAX_FEEDRATE {500, 500, 5, 25} // (mm / sec)

За замовчуванням стоять швидкості 500,500,5, 25 мм / с на осі X, У, Z і екструдер відповідно. Рекомендуємо знизити швидкість з 500 до 200.

Налаштування прискорення переміщень по осях

Ще однією з важливих налаштувань є завдання прискорень для різних осей, так як через некоректну настройки цього моменту часто бувають проблеми при друку, а саме зміщення шарів з причини пропуску кроків двигуна. Якщо поставити занадто великі прискорення, то будуть пропуски[5].

За замовчуванням в прошивці стоять наступні значення:

#define DEFAULT_MAX_ACCELERATION {9000,9000,100,10000} // X, У, Z, E maximum start speed for accelerated moves. E default values are good for Skeinforge 40+, for older versions raise them a lot.

#define DEFAULT_ACCELERATION 3000 // X, У, Z and E max acceleration in mm / s ^ 2 for printing moves

#define DEFAULT_RETRACT_ACCELERATION 3000 // X, У, Z and E max acceleration in mm / s ^ 2 for retracts

Для осей X і У стоять прискорення 9000 мм / с ^ 2 - це дуже багато.

Для первинної настройки встановіть не більше 1000 і для DEFAULT_ACCELERATION поставте 1500, замість 3000.

Рисунок 3.13. - Налаштування прискорення переміщень по осях

Активація дисплея

Останнє, що залишається зробити - це активувати потрібний дисплей. Один з найпопулярніших дисплеїв, це RepRapDiscount Smart Controller . Знайдіть і розкоментуйте наступні рядки:

#define ULTRA_LCD

#define SDSUPPORT

#define ULTIPANEL

#define REPRAP_DISCOUNT_SMART_CONTROLLER

Перед цими рядками, не повинні стояти "//". Має вийти наступне:

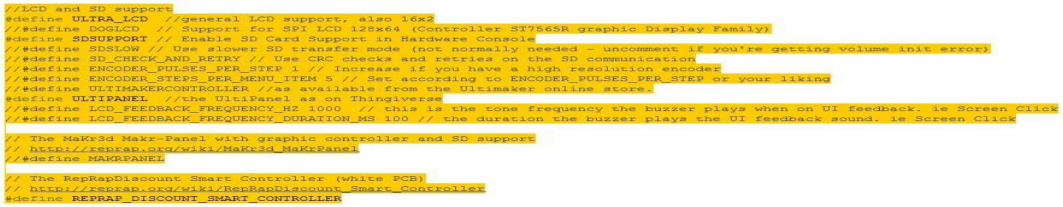

Рисунок 3.14. - Активація дисплея

Розглянемо заливку прошивки.Після всіх основних змін прошивки, можна її заливати. У програмі Arduino IDE зайдіть у вкладку "Інструменти" -> "Плата" і виберіть "Arduino / Genuino Mega or Mega 2560"

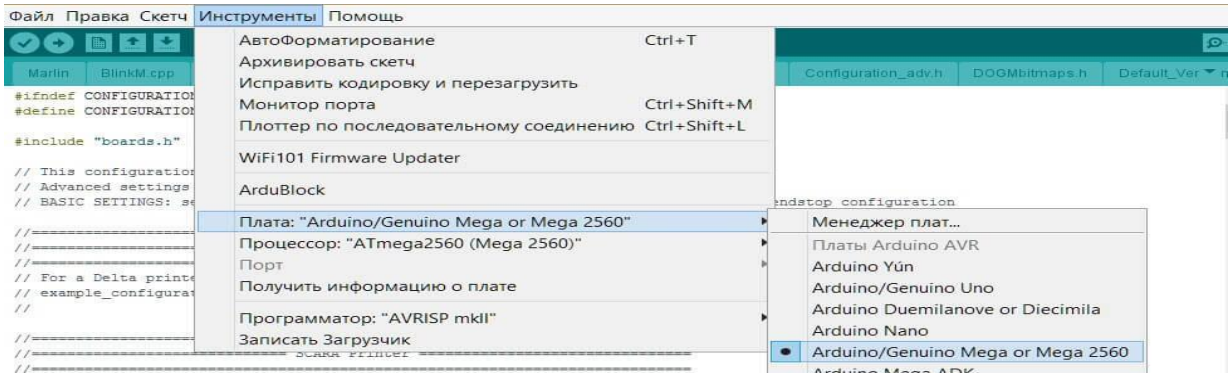

Рисунок 3.15. - Вибір плати для загрузки прошивки

Щоб завантажити прошивку, необхідно правильно встановити COM-порт вашого 3D-принтера. Для цього натисніть на іконку зі стрілкою в колі. Прогрес

завантаження прошивки відображатиметься індикатором, а після успішного завершення на екрані з'явиться підтвердження. Потім ви можете спробувати запустити свій 3D-принтер. Після того, як все почне працювати, необхідно калібрувати PID-регулятор нагріву головки і стола. Я використовую Pronterface для цієї операції. Введіть команду ' M303 E0 C8 S260 '. Тут M303 - команда для калібрування, E0 - головка, C8 - кількість циклів нагрівання-охолодження, S260 - типова температура роботи сопла. [5].

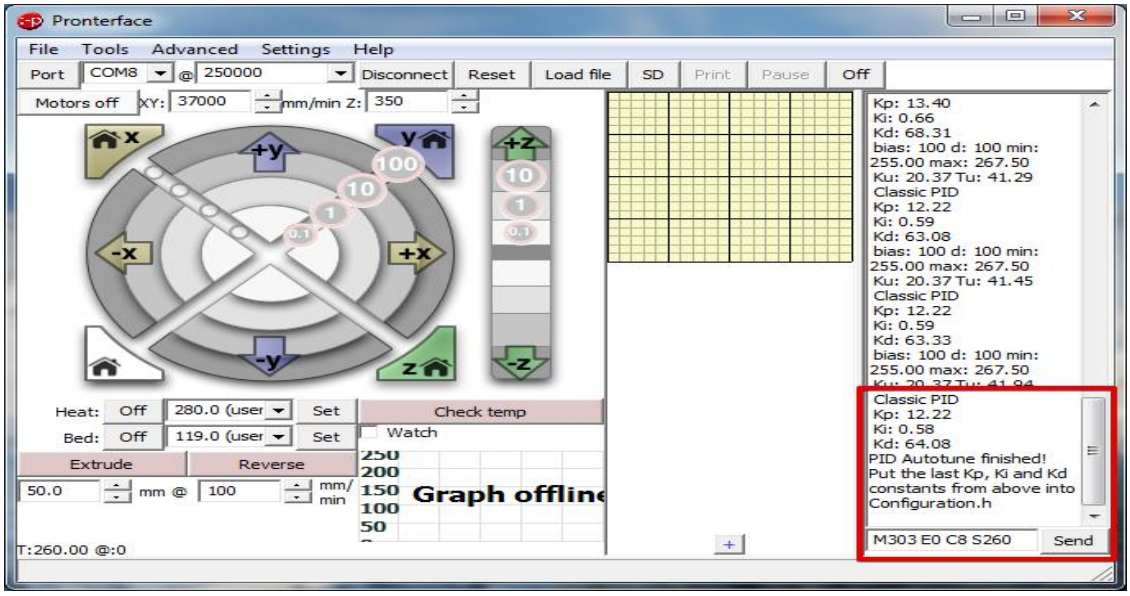

Рисунок 3.16. - Відкалібрування PID нагріву хотенда і столу

Останні результати записуємо в прошивку .

#define	DEFAULT_Kp	12.22
#define	DEFAULT_Ki	0.58
#define	DEFAULT_Kd	64.08

За такою ж схемою калібруємо PID столу . Команда ' M303 E-1 C8 S110 '. Де E-1 - стіл, S110 - типова температура нагріву столу. Останні результати записуємо в прошивку . Як стіл нагрівається дуже повільно і тому доводиться перезапускати команду через помилку Timeout.Дані результати потрібно записати в прошивку[5].

#define DEFAULT_bedKp 105.94

#define DEFAULT_bedKi 4.97

#define DEFAULT_bedKd 563.11

Щоб дізнатися вже прошиті в 3D принтер параметри щоб не калібрувати повторно, потрібно завантажити програму Repetier-Host. Потім через перейти по вкладкам : меню-Конфігурація-Конфігурація EEPROM . Попередньо потрібно вказати COM порт в настройках і натиснути кнопку ' Приєднати '.

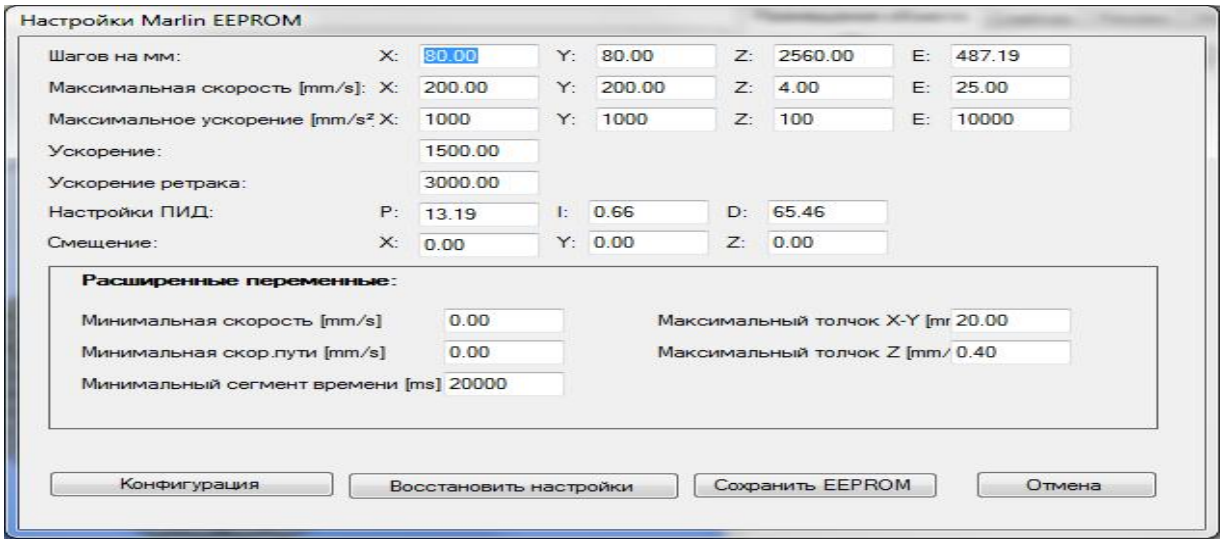

Рисунок 3.17. - Ввод даних для прошивки

При підключенні 3D принтера до програми Pronterface в правій частині вікна виводиться списку параметрів прошивки.

Також параметри можна побачити на LCD дисплеї . Через меню Control-Motion [5].

3.3. Засоби пошуку та усунення несправностей

Для діагностики електричної складової прінтера використаємо мультиметр DT-9208A.

Мультиметр DT9208A призначений для вимірювання струму, напруги, опору, параметрів діодів і транзисторів, а також частоти і температури. DT9208A є одним з найкращих мультиметрів за співвідношенням ціна / якість. Тому багато людей обирають цей пристрій як заміну простішому DT830. Для початку роботи з мультиметром перевірте заряд батареї і увімкніть пристрій. Якщо батарея розряджена, на дисплеї з'явиться знак "[- +]". Замініть батарею. Знак "!" біля роз'ємів мультиметра вказує на те, що вхідні струми і напруги не

повинні перевищувати вказаних значень. Це зроблено для захисту схеми мультиметра від пошкоджень. Перед вимірюванням необхідно налаштувати перемикач діапазонів на потрібну межу вимірювання. Якщо межа струму або напруги, яку потрібно виміряти, невідома, встановіть перемикач на максимум, а потім знижуйте його при необхідності. Якщо на дисплеї з'являється символ "1" (перевантаження), необхідно переключитися на вищу межу вимірювань. [10].

Рисунок 3.18. -Цифровий мультиметр DT-9208A

Для вимірюваня температури стола і екструдера використаємо пірометр GM320. Пірометр - це безконтактний термометр. Область застосування пірометрів дуже широка. Але в основному пірометри набувають для вимірювання температури стін, склопакетів, пошуку джерел тепловтрати, температури двигунів, для харчової промисловості, вимірювання температури при куванні металу.

Рисунок 3.19. - пірометр GM320

ВИСНОВОК

У кваліфікаційній роботі було спроектовано 3D прінтер під керуванням Ардуіно Mega 2560. В ході кваліфікаційної роботи було зроблено повне дослідження технології 3D-друку, історія її походження, області застосування, технології друку. Так само вищенаведене дослідження дозволило зробити наступні висновки: 3D принтери стрімко удосконалюються і дешевшають збільшується кількість 3D моделей які можна скачати і роздрукувати безкоштовно, при цьому досить високо цінується вміння створювати і редагувати 3D моделі. Уміння моделювати в 3D дає можливість створювати унікальні речі на 3D принтерах вже друкують органи, зброя, деталі роботів, одяг, будинки. Можливості обмежені тільки вашою фантазією 3D принтери почали з'являтися у продажу в великих гіпермаркетах в недалекому майбутньому принтер стане класичним незамінним предметом побутової техніки, існують і активно розробляються принтери, здатні відтворювати більшу частину себе самих . Отримані знання та навички можуть бути корисними при подальшому використанні 3D-друку у різних галузях, таких як медицина, архітектура, інженерія та інші.

Таким чином, проект було успішно виконано, і результати роботи можуть бути використані для подальшого розвитку та вдосконалення 3D-друку.

СПИСОК ЛІТЕРАТУРИ

1. AliExpress [Електронний ресурс] : [Веб-сайт] - Електронні дані. -Reprap Ramps 1,4 комплект з Mega 2560 r3 ,Heatbed MK2B, 12864 ЖК-контролер, DRV8825, механічний перемикач - Режим доступу : https://aliexpress.ru /item/32924529176.html?spm=a2g0s.9042311.0.0.69c133edWpWItp (дата звернення 06.04.2023). - Назва з екрана.
2. AliExpress [Електронний ресурс] : [Веб-сайт] - Електронні дані. - Імпульсний блок живлення JOTTA S-360-12 - Режим доступу: https://aliexpress.ru /item/32633322222.html?spm=a2g0s.9042311.0.0.119633ediУ1FVb (дата звернення 08.04.2023). - Назва з екрана.
3. Ast3d [Електронний ресурс] : [Веб-сайт] - Електронні дані. E3D Chimera HotEnd - Режим доступу: https://ast3d.com.ua/product/ekstruder-himera3 (дата звернення 08.04.2023). - Назва з екрана.
4. 3DiУ [Електронний ресурс] : [Веб-сайт]. - Електронні дані. -Настройка прошивки Marlin- Режим доступу: https://3d-diy.ru/blog/3dprintery/nastrojka-proshivki-marlin/(дата звернення 30.04.2023). - Назва з екрана.
5. 3d_print [Електронний ресурс] : [Веб-сайт]. - Електронні дані. -Настройка прошивки Marlin- Режим доступу: https://3d_print.jofo.me/1618962.html (дата звернення 30.04.2023). - Назва з екрана.
6. Joyta [Електронний ресурс] : [Веб-сайт]. - Електронні дані. Arduino IDE - Режим доступу : http://www.joyta.ru/10685-arduino-IDE-opisanie-skachat/ (дата звернення 30.04.2023). - Назва з екрана.
7. Rozetka [Електронний ресурс] : [Веб-сайт]. - Електронні дані. -3D принтер Artillery Hornet- Режим доступу:https://rozetka.com.ua/ua/91116584/p911 16584/ (дата звернення 30.04.2023). - Назва з екрана.
8. 3dreams [Електронний ресурс] : [Веб-сайт]. - Електронні дані. - 3D принтер Original Prusa i3 MK3S+- Режим доступу:https://3dreams.com.ua/prod uct/3d-

%D0%BF%D1%80%D0%B8%D0%BD%D1%82%D0%B5%D1%80-original-prusa-i3-mk3/(дата звернення 30.03.2023). - Назва з екрана.

9. Rozetka [Електронний ресурс] : [Веб-сайт]. - Електронні дані. - Кроковий двигун Resheto NEMA17 1.7A- Режим доступу : https://rozetka.com.ua/ua/168838405/p168838405/characteristics/ (дата звернення 08.04.2023). - Назва з екрана.
10. Cityset [Електронний ресурс] : [Веб-сайт]. - Електронні дані. -DT9208A - Режим доступу : https://cityset.com.ua/dt9208a (дата звернення 09.04.2023). - Назва з екрана.
11. Uamper [Електронний ресурс] : [Веб-сайт]. - Електронні дані. - платформа MK2A - Режим доступу : https://uamper.com/3D-platforms/220mm-round-corner-aluminum-heat-bed (дата звернення 09.05.2023). - Назва з екрана.
12. Wiki.tntu [Електронний ресурс] : [Веб-сайт]. - Електронні дані. Arduino Mega- Режим доступу : https://wiki.tntu.edu.ua/Arduino_Mega_2560 (дата звернення 30.05.2023). - Назва з екрана.
13. Вікіпедія [Електронний ресурс] : [Веб-сайт] - Електронні дані. - Адитивні технології - Режим доступу: https://uk.wikipedia.org/wiki/%D0%90%D0%B4 %D0%B8%D1%82%D0%B8%D0%B2%D0%BD%D1%96_%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D1%97 (дата звернення 07.04.2023). - Назва з екрана.