**СЪДЪРЖАНИЕ**

**Увод………………………………………………………………………..**

**Глава I: Литературен обзор………………………………….…………**

**1 Вградена система ………………………………………………………**

**1.1Как работят вградените системи? …………………………………**

**1.2 Характеристики на вградените системи …………………………**

**1.3 Видове вградени системи ……………………………………………**

**Глава II: Използвани технологии………………………………………**

**2.1 Arduino Nano…………………………………………………………...**

**2.2 C++………………………………………………………………………**

**2.3 LCD дисплей…………………………………………………………...**

**2.4 Стъпкови мотори………………………………………………………**

**2.5 Arduino IDE…………………………………………………………….**

# 2.6 I2C……………………………………………………………………….

# 2.7 CNC shield V4…………………………………………………………..

# 2.8 Ротационен енкодер……………………………………………………

# 2.9 3D принтиране…………………………………………………………

**Глава III: Практическа разработка…………………………………….**

# 3.1 3D части на проекта……………………………………………………

**3.2** **Калибриране и настройки…………………………………………….**

**3.3 Тестване на системата…………………………………………………**

**3.3.1 Първоначално включване……………………………………**

**3.3.2 Тест на LCD дисплея…………………………………………..**

**3.3.3 Тест на бутона…………………………………………………..**

**3.3.4 Тест на стъпковите мотори……………………………………**

**3.3.5 Тест на EEPROM паметта……………………………………..**

**3.3.6 Тест на цялостната система……………………………………**

**Глава IV: Заключение……………………………………………………….**

**Глава V: Използвана литература…………………………………………****..**

**Глава VI Приложения………………………………………………………..**

**Приложение 1………………………………………………………………….**

**Приложение 2………………………………………………………………….**

**Приложение 3………………………………………………………………….**

**Приложение 4………………………………………………………………….**

**Приложение 5………………………………………………………………….**

**Списък с фигури………………………………………………………………**

**Списък с използваните съкращения…………………………..……………**

# СПИСЪК С ФИГУРИ

Фигура 1 /Arduino Nano/

Фигура 2 /Програмен език C++/

Фигура 3 /LCD дисплей/

Фигура 4 /Стъпков мотор/

Фигура 5 /Платка Arduino IDE/

Фигура 6 /I2C комуникация/

Фигура 7 /CNC V4/

Фигура 8 /Ротационен енкодер/

Фигура 9 /3D принтер/

Фигура 10 /Проект във виртуална среда/

Фигура 11 /3D части/

Фигура 12 /3D части/

Фигура 13 /3D части/

# СПИСЪК С ИЗПОЛЗВАНИТЕ СЪКРАЩЕНИЯ

CNC - Computerised Numerical Control /цифрово програмно управление/

IDE – integrated development environment /интегрирана среда за разработка/

I2C - Inter-Integrated Circuit /Междуинтегрална схема/

LCD - liquid crystal display /дисплей с течни кристали/

3D - 3-Dimensional /3-Измерен/

# УВОД

В съвременното общество, белязано от бързи темпове на живот и нарастващо дълголетие, здравословната поддръжка и спазването на лекарствени режими придобиват все по-голямо значение. Особено за възрастни хора, пациенти с хронични заболявания или хора със затруднения в паметта, ежедневният прием на лекарства в точен час и в правилна доза е не просто необходимост, а жизненоважна рутина.

В този контекст нараства нуждата от автоматизирани и интелигентни решения, които да подпомагат потребителите в управлението на техните медикаменти. Един такъв инструмент е диспенсърът за хапчета – устройство, предназначено да организира, дозира и напомня за приема на лекарства, като същевременно намалява риска от човешки грешки, предозиране или пропуски.

Целта на настоящата дипломна работа е проектиране и разработване на диспенсър за хапчета, който съчетава механични и електронни компоненти с цел постигане на надеждна автоматизация, удобство за потребителя и възможност за отдалечено наблюдение или известяване. Проектът ще разгледа както хардуерната конструкция, така и софтуерната логика, стояща зад управлението на устройството.

Разработването на подобно решение е актуално не само поради социалната му значимост, но и поради технологичния потенциал, който може да бъде приложен в сферата на здравеопазването и домашната грижа.

# ГЛАВА 1. ЛИТЕРАТУРЕН ОБЗОР

## 1 Вградени системи

Вградената система или Embedded system е комбинация от компютърен хардуер

и софтуер, предназначени за специфична функция. Вградените системи могат

също да функционират в рамките на по-голяма система. Системите могат да бъдат

програмируеми или с фиксирана функционалност. Промишлени машини,

потребителска електроника, селскостопански и преработвателни устройства,

автомобили, медицинско оборудване, фотоапарати, цифрови часовници, домакински

уреди, самолети, автомати и играчки, както и мобилни устройства, са възможни места

за вградена система.

Въпреки че вградените системи са изчислителни системи, те могат да варират от

липса на потребителски интерфейс (UI) - например на устройства, проектирани да

изпълняват една задача - до сложни графични потребителски интерфейси (GUI), като

например в мобилни устройства. Потребителските интерфейси могат да включват

бутони, светодиоди (светодиоди) и сензорен екран. Някои системи използват и

отдалечени потребителски интерфейси.

**1.1 Как работят вградените системи?**

Вградените системи представляват специализирани компютърни системи, които са интегрирани в рамките на по-големи устройства с цел изпълнение на конкретна функция или набор от функции. Те се различават от универсалните компютри по това, че не са предназначени за общо използване, а за специфични задачи, които често изискват надеждност, ефективност и работа в реално време.

Сърцевината на една вградена система обикновено включва микроконтролер или микропроцесор, който управлява хардуерни компоненти – сензори, мотори, дисплеи, бутони и други периферни устройства. Софтуерът, който управлява тази хардуерна част, е предварително програмиран и обикновено не подлежи на промяна от крайния потребител.

Тези системи се използват в огромен брой устройства от ежедневието – от домакински уреди като перални машини и микровълнови фурни до автомобили, медицинска апаратура, индустриални машини и дори космически технологии. Основното им предимство е, че позволяват висока степен на автоматизация и оптимизация, като същевременно работят при ограничени ресурси – ниска консумация на енергия, малко място и ограничена изчислителна мощност.

Проектирането на вградена система изисква добро разбиране както на хардуерната архитектура, така и на софтуерното програмиране. Обикновено се използват езици като C или C++, като се залага на стабилност, минимално време за реакция и ефективно използване на ресурсите.

## 

## 1.2 Характеристики на вградените системи

Вградените системи се отличават със специфичен набор от характеристики, които ги правят особено подходящи за изпълнение на конкретни задачи в рамките на по-големи устройства. Една от основните им черти е тяхната специализация – те са проектирани да извършват точно определена функция и често работят в строго определена среда. Това позволява оптимизация както на хардуера, така и на софтуера, така че системата да бъде възможно най-ефективна.

Друга съществена характеристика е работата в реално време. Много от вградените системи трябва да реагират мигновено на външни събития – например при управление на въздушни възглавници в автомобил или при контрол на индустриален процес. В тези случаи дори минимално закъснение може да доведе до сериозни последици, което изисква прецизно програмиране и бързодействие.

Типично за вградените системи е и тяхната компактност. Те заемат малко пространство, използват ограничено количество памет и изчислителна мощност, и често работят с минимално енергопотребление. Това ги прави особено подходящи за мобилни устройства, носима електроника или батерийно захранвани приложения.

Надеждността и устойчивостта също са ключови характеристики. Тъй като вградените системи често функционират в критични условия – в медицински апарати, в авиацията, в автомобилостроенето – от тях се изисква безотказна работа за дълги периоди от време без нужда от намеса.

Освен това, повечето вградени системи са тясно свързани с хардуера, в който са интегрирани. Това налага внимателно съгласуване между хардуер и софтуер още на етапа на проектиране. Не на последно място, сигурността често играе важна роля, особено когато системите са свързани към мрежа или обработват чувствителна информация.

**1.3 Видове вградени системи**

Вградените системи могат да бъдат изцяло професионални и индустриални, както и

експериментални. В настоящия контекст ще разгледаме развойните системи,

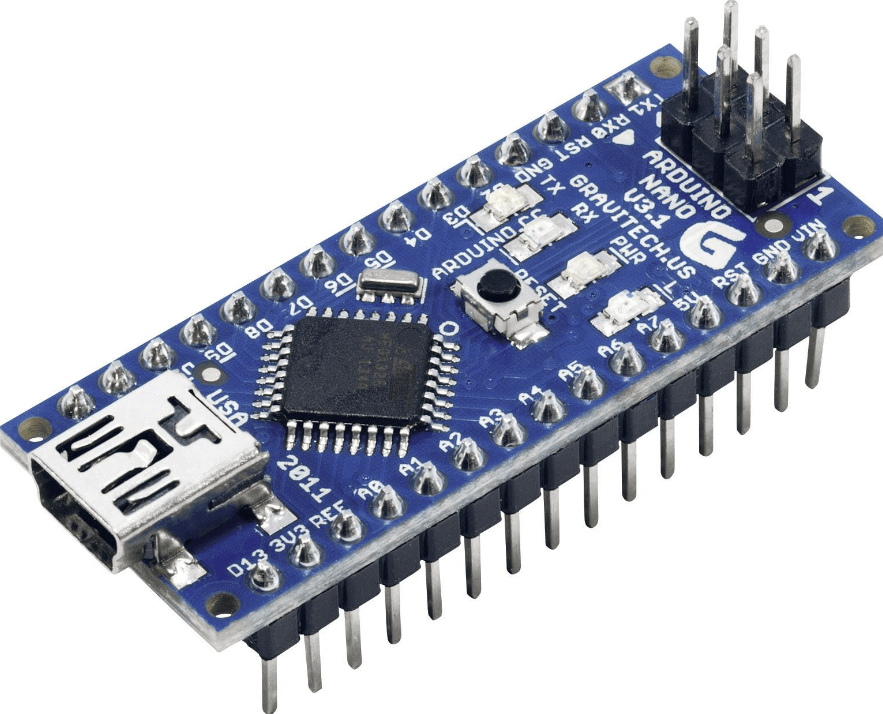
предназначени за разработка на експериментални постановки в различни среди, без да

се ограничаваме до конкретни професионални приложения.

# ГЛАВА 2. ИЗПОЛЗВАНИ ТЕХНОЛОГИИ

**2.1 Arduino Nano**

Arduino Nano е компактен и енергийно ефективен микроконтролерен модул, който се базира на чипа ATmega328. Благодарение на малките си размери той е изключително подходящ за проекти с ограничено пространство, като носими устройства, роботика и вградени системи. Платката разполага с достатъчен брой цифрови и аналогови входове и изходи, което я прави гъвкава при управление на различни сензори и изпълнителни механизми. Arduino Nano може да се програмира лесно чрез USB кабел с помощта на Arduino средата за разработка, като използва езика C или C++ със специфични библиотеки. Устройството се захранва чрез USB или външен източник на напрежение, като поддържа работа на ниско напрежение, което го прави подходящо за енергийно чувствителни приложения. Сред предимствата на Arduino Nano са неговата достъпност, голямата общност от разработчици и богатата документация, които улесняват бързото прототипиране и обучение в сферата на електрониката и програмирането.



*Фигура 1*

**2.2 C++**

C++ е мощен език за програмиране, който комбинира процедурно и обектно-ориентирано програмиране. Той е създаден като разширение на езика C, като добавя възможности за по-структурирано и модулно писане на код чрез използване на класове и обекти. Това прави C++ подходящ както за писане на ниско ниво на код, близък до хардуера, така и за изграждане на сложни софтуерни системи и приложения.

C++ позволява директен достъп до паметта чрез указатели, което дава голям контрол на програмиста, но също така изисква внимание при управлението на ресурсите. Езикът поддържа наследяване, полиморфизъм и капсулация – основните принципи на обектно-ориентираното програмиране, които помагат за по-добра организация и повторна употреба на кода.

Често се използва в разработката на операционни системи, игри, вградени системи, симулатори и приложения, където се изисква висока производителност. Въпреки че е по-сложен за усвояване в сравнение с някои съвременни езици, C++ остава един от най-използваните и ценени езици в софтуерната индустрия, благодарение на своята гъвкавост, ефективност и близост до хардуера.

****

*Фигура 2*

**2.3 LCD дисплей**

LCD означава *Liquid Crystal Display* – дисплей с течни кристали. Това е един от най-разпространените видове екрани, използвани в електроника като калкулатори, часовници, телефони, телевизори, дисплеи за Arduino и други устройства.

LCD дисплеят работи чрез блокиране или пропускане на светлина, а не чрез нейното излъчване. Вътре има два поляризиращи филтъра и слой от течни кристали между тях. Когато се подаде електрически сигнал, кристалите променят ориентацията си и така влияят на преминаването на светлина – това позволява на определени пиксели да изглеждат тъмни или светли.

Този тип дисплеи са много енергийно ефективни, особено когато се използват в комбинация с LED подсветка. Те могат да бъдат с различен тип: символни LCD (например 16x2 или 20x4 символа, често срещани при проекти с Arduino) или графични LCD, които показват пиксели и изображения.

LCD е популярен, защото е евтин, компактен и лесен за управление с микроконтролери. Изисква сравнително малко пинове и често използва интерфейси като I2C или паралелна връзка.



*Фигура 3*

**2.4 Стъпкови мотори**

Стъпковите мотори са вид електрически мотори, които се въртят на стъпки вместо да се въртят плавно както при обикновените мотори. Всеки импулс, който моторът получава, го завърта с определен ъгъл което позволява изключително прецизен контрол на позицията. Това ги прави много подходящи за приложения, където е важно точното позициониране като 3D принтери, CNC машини, роботи и различни автоматизирани механизми.

Съществуват различни видове стъпкови мотори като униполярни и биполярни които се различават по начина на управление на намотките. Стъпковите мотори могат да работят в различни режими – пълностъпков, половин стъпка или микростъпка – в зависимост от нужната прецизност и плавност на движението.

Тъй като тези мотори не се нуждаят от сензори за позиция, те са лесни за управление с микроконтролери като Arduino и често се използват в любителски и професионални проекти. Все пак изискват подходящ драйвер за да се подават точните импулси в правилната последователност и със съответния ток.



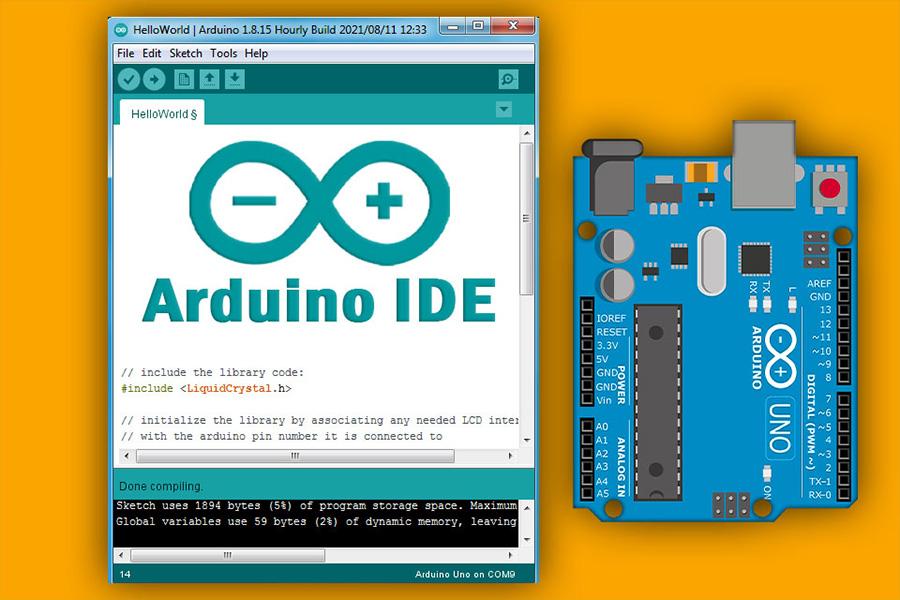
*Фигура 4*

**2.5 Arduino IDE**

Arduino IDE е софтуерна среда за писане, компилиране и качване на програми върху платки от семейството на Arduino. IDE означава Integrated Development Environment, което означава интегрирана среда за разработка. Тя предоставя удобно място където потребителите могат да пишат своя код на езика C или C++, да го проверят за грешки и да го качат директно върху микроконтролера чрез USB кабел.

Arduino IDE е създаден с цел да бъде лесен за използване дори от начинаещи. Интерфейсът е семпъл и интуитивен, включва основни бутони за компилиране и качване, както и сериен монитор за наблюдение на данни от платката в реално време. Освен това има вградена библиотека с примери които помагат на потребителите да започнат бързо и да експериментират с различни сензори и компоненти.

Едно от най-силните качества на Arduino IDE е неговата отвореност и възможността да се разширява чрез допълнителни библиотеки и поддръжка на различни платки, включително не само оригинални Arduino, но и съвместими устройства. Това превръща средата в мощен инструмент за разработване на всевъзможни електронни проекти, от най-простите до доста сложни системи за автоматизация, управление и комуникация.



*Фигура 5*

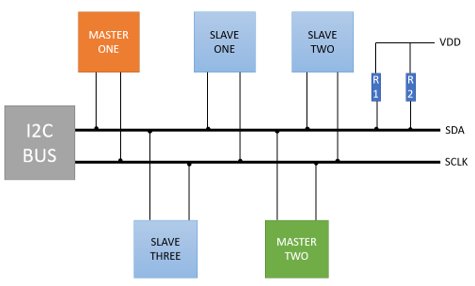
**2.6 I2C**

I2C комуникацията е сериен протокол за предаване на данни между микроконтролери и периферни устройства като сензори дисплеи памети и други компоненти. Името I2C идва от Inter-Integrated Circuit и представлява двупроводна комуникация при която се използват само две линии – една за данни и една за синхронизация. Линията за данни се нарича SDA а линията за синхронизация SCL. Тези две линии се споделят между всички устройства свързани в комуникацията което позволява използване на една и съща шина за множество компоненти.

Комуникацията се осъществява между едно главно устройство което обикновено е микроконтролер като Arduino и едно или повече подчинени устройства. Главното устройство контролира темпото на предаване чрез SCL линията и определя кога и с кое подчинено устройство ще комуникира. Всяко подчинено устройство има уникален адрес по който бива разпознавано.

Едно от основните предимства на I2C е простотата на свързване и малкият брой необходими пинове. Това прави I2C особено подходящ за системи с ограничени хардуерни ресурси. Също така позволява свързване на множество устройства към една и съща шина което опростява схемата и кабелите. Протоколът е сравнително бавен в сравнение с други като SPI но за много приложения като температурни сензори или LCD дисплеи скоростта е напълно достатъчна.

I2C се използва широко в електрониката и е добре поддържан от различни микроконтролери включително Arduino. С помощта на специални библиотеки и функции комуникацията се настройва лесно и позволява ефективен обмен на данни между устройства без сложен хардуер.



*Фигура 6*

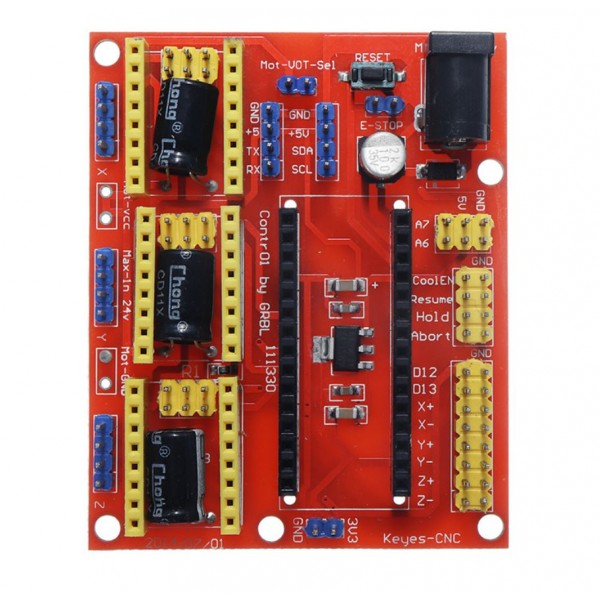
**2.7 CNC shield V4**

CNC Shield V4 е платка за разширение, предназначена за управление на стъпкови мотори чрез микроконтролер, обикновено Arduino Nano. Тя е специално създадена за проекти като CNC машини, 3D принтери и гравиращи устройства, където се изисква прецизно движение по няколко оси.

Тази платка позволява лесно свързване на до четири стъпкови мотора чрез драйвери от типа A4988 или DRV8825, които се поставят директно върху щита. Всяка ос – обикновено X, Y, Z и допълнителна A ос – има свои пинове за управление на посока и стъпка, като сигналите се подават от Arduino. Това улеснява програмирането и контрола на движенията чрез специализиран софтуер като GRBL, който се качва на Arduino-то.

CNC Shield V4 има и пинове за свързване на крайни изключватели (endstops), които определят начални и крайните позиции на осите, както и захранващи терминали за подаване на по-високо напрежение към моторите, обикновено 12V или 24V. Освен това версия 4 е съвместима с Arduino Nano, което я прави по-компактна в сравнение с по-старите версии, които се използват с Arduino Uno.

Основната идея зад CNC Shield V4 е да осигури лесен и надежден начин за изграждане на системи с прецизен контрол върху движение, като същевременно се намалява сложността при окабеляване и свързване. Това го прави предпочитан избор за любители и създатели на машини с цифрово управление.



*Фигура 7*

**2.8 Ротационен енкодер**

Ротационният енкодер е електромеханично устройство, което превръща ъгловото движение или позиция на ос в електрически сигнал. Най-често се използва за определяне на посоката и броя на завъртанията на вал или копче. Той работи като въртящ се превключвател, но за разлика от потенциометъра, енкодерът няма краен ход – може да се върти безкрайно в двете посоки.

Има два основни вида ротационни енкодери – **инкрементални** и **абсолютни**. При инкременталните се изпращат импулси всеки път когато валът се завърти, а посоката на въртене се определя по реда на импулсите от двата изхода (обикновено означени като A и B). При абсолютните енкодери всеки ъгъл на завъртане има уникален цифров код, така че винаги може да се определи точната позиция.

Най-често използваният в проекти с Arduino е инкременталният енкодер. Той се използва в интерфейси като цифрови потенциометри, менюта, контролери за мотор и други приложения, където е нужно да се отчита въртене с прецизност. Често има и вграден бутон, който реагира на натискане при завъртане, което позволява още по-интерактивен контрол.

Енкодерите са много полезни, защото предлагат дълъг живот, висока точност и възможност за безкрайно въртене без износване на позицията. Лесно се интегрират в проекти с микроконтролери, като се използват прекъсвания или четене на импулси за определяне на движението.



*Фигура 8*

**2.9 3D принтиране**

3D принтирането е процес, при който се създават триизмерни обекти чрез добавяне на материал слой по слой. Нарича се още адитивно производство, защото за разлика от традиционните методи, при които материал се изрязва или отнема (например с фреза), тук материалът се добавя точно там, където е нужен. Най-често се използват пластмаси като PLA или ABS, но има и принтери, които работят с метал, смола, бетон и дори шоколад.

Процесът започва със създаване на 3D модел на компютър чрез специализиран софтуер като Tinkercad, Fusion 360 или Blender. След това моделът се "нарязва" на тънки хоризонтални слоеве със слайсинг програма като Cura или PrusaSlicer. Резултатът е G-код – инструкции, които казват на принтера къде да се движи и колко материал да подаде във всеки момент.

По време на самото принтиране дюзата на принтера нагрява пластмасовия нишков материал и го отпечатва слой по слой, докато се изгради желаният обект. Процесът може да отнеме от няколко минути до няколко часа или дори дни в зависимост от размера и сложността на модела.

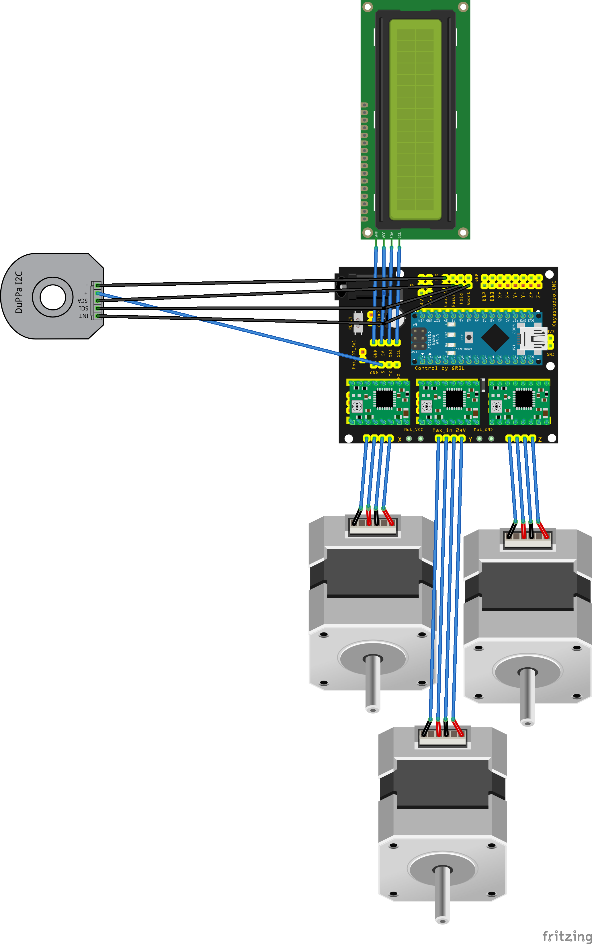
3D принтирането намира приложение в много области – от прототипиране и инженерство до медицина, мода и изкуство. С него могат да се създават както функционални части, така и декоративни елементи. Особено популярно е сред любители и малки производители, защото позволява бързо и евтино създаване на уникални или персонализирани обекти.



*Фигура 9*

# ГЛАВА 3. ПРАКТИЧЕСКА РАЗРАБОТКА

**1. Разработка на проект във виртуална среда**



*Фигура 10*

2**. Код с примерни стойности**

LCD дисплей:

-Адрес: 0x27

-Размер: 16 колони, 2 реда

#include <LiquidCrystal\_I2C.h>

#include <AccelStepper.h>

#include <EEPROM.h>

Използването на EEPROM позволява запазване на последно избрания ден, дори при рестарт на устройството.

LiquidCrystal\_I2C lcd(0x27, 16, 2);

AccelStepper dispenserZ(AccelStepper::DRIVER, 7, 4);

AccelStepper dispenserX(AccelStepper::DRIVER, 5, 2);

AccelStepper dispenserY(AccelStepper::DRIVER, 6, 3);

Стъпкови мотори (тип DRIVER):

X ос: DIR (5), STEP (2)

Y ос: DIR (6), STEP (3)

Z ос: DIR (7), STEP (4)

Управляващ бутон:

-Свързан към аналогов пин A3 (използва се като цифров вход).

-Състоянието се следи чрез digitalRead().

const int buttonPin = A3;

unsigned long buttonPressTime = 0;

bool buttonHeld = false;

const int stepsPerSlot = 2000;

int currentDay = 0;

String days[7] = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"};

void setup() {

Serial.begin(9600);

pinMode(8, OUTPUT);

digitalWrite(8, LOW);

Начална инициализация (setup)

Задават се максимални скорости и ускорения за моторите.

Инициализира се LCD и се включва подсветката.

Задава се пин 8 като изход и се държи в LOW (вероятно за управление на релета или индикатор).

Чете се последно запазения ден от EEPROM.

Ако данните са невалидни, се нулира към понеделник.

dispenserX.setMaxSpeed(500); dispenserX.setAcceleration(300);

dispenserY.setMaxSpeed(500); dispenserY.setAcceleration(300);

dispenserZ.setMaxSpeed(500); dispenserZ.setAcceleration(300);

lcd.init();

lcd.backlight();

pinMode(buttonPin, INPUT\_PULLUP);

currentDay = EEPROM.read(0);

if (currentDay < 0 || currentDay > 6) currentDay = 0;

LCD показва стартовия екран:

lcd.setCursor(0, 0);

lcd.print("Dispenser Ready");

lcd.setCursor(0, 1);

lcd.print("Day: ");

lcd.print(days[currentDay]);

}

Главен цикъл (loop)

Работа на моторите:

Използва се .run() за всички мотори, което позволява непрекъснато движение (неблокиращо).

void loop() {

dispenserX.run();

dispenserY.run();

dispenserZ.run()

Управление с бутон:

При засичане на натискане (LOW) → записва началния момент и отбелязва, че бутонът е задържан.

if (digitalRead(buttonPin) == LOW && !buttonHeld) {

buttonPressTime = millis();

buttonHeld = true;

}

if (digitalRead(buttonPin) == HIGH && buttonHeld) {

unsigned long duration = millis() - buttonPressTime;

buttonHeld = false;

Когато бутонът се освободи (HIGH) и е бил натиснат:

-Изчислява се продължителността.

-Ако е над 800 ms → стартира дозиране.

-Ако е по-кратко → сменя деня (инкрементира и записва в EEPROM).

if (duration > 800) {

dispenseToday();

} else {

currentDay = (currentDay + 1) % 7;

EEPROM.write(0, currentDay);

Функция за дозиране: dispenseToday()

Целта е да позиционира главата за дозиране спрямо текущия ден.

Етапи:

Показва съобщение на дисплея:

lcd.clear()

lcd.setCursor(0, 0);

lcd.print("Select Day:");

lcd.setCursor(0, 1);

lcd.print(days[currentDay]);

}

delay(200);

}

}

void dispenseToday() {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Dispensing:");

lcd.setCursor(0, 1);

lcd.print(days[currentDay]);

Изчислява целева позиция:

long targetPosition = currentDay \* stepsPerSlot;

Подава команда към всички мотори:

dispenserX.moveTo(targetPosition);

dispenserY.moveTo(targetPosition);

dispenserZ.moveTo(targetPosition);

Изчаква мотори да достигнат позицията:

while (dispenserX.distanceToGo() != 0 ||

dispenserY.distanceToGo() != 0 ||

dispenserZ.distanceToGo() != 0) {

dispenserX.run();

dispenserY.run();

dispenserZ.run();

}

Изписва съобщение „Готово“:

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Done!");

lcd.setCursor(0, 1);

lcd.print("Next: ");

lcd.print(days[currentDay]);

## delay(1000);

## }

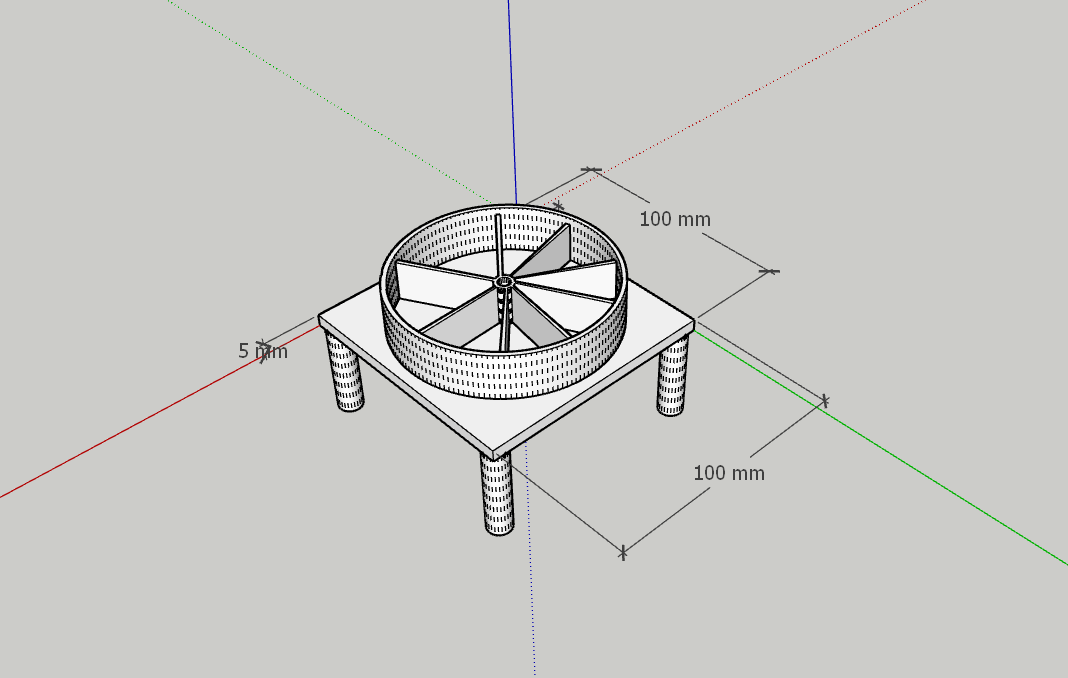
Този код реализира интуитивен и надежден седмичен дозатор, използващ минимум интерфейс (бутон + дисплей), и е подходящ за системи за лекарствено или хранително дозиране. Кодът е структуриран, използва асинхронно управление на мотори и предоставя добра основа за разширения като:

-Автоматична смяна на деня чрез RTC модул.

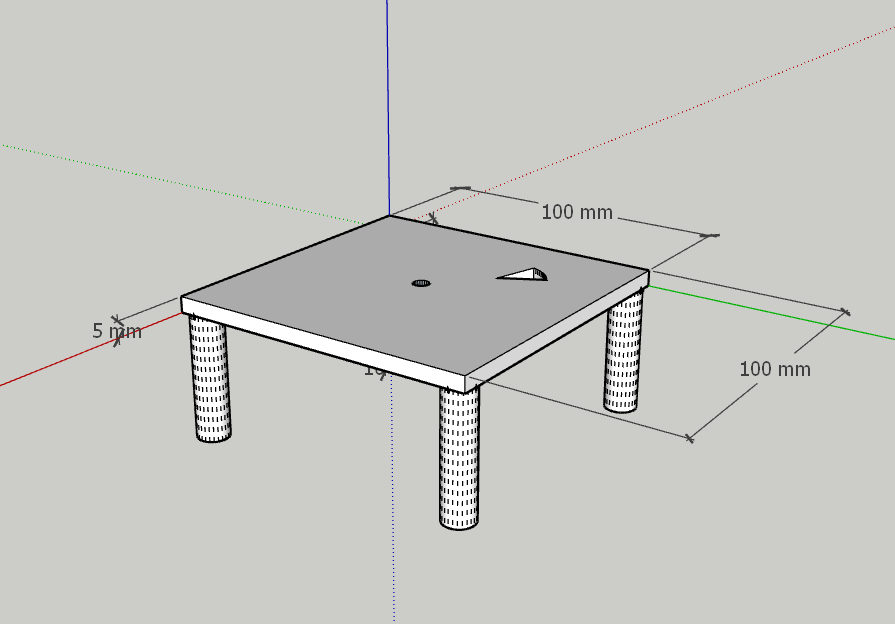
-Уеб или Bluetooth интерфейс за настройка.

-Повече мотори или сензори.

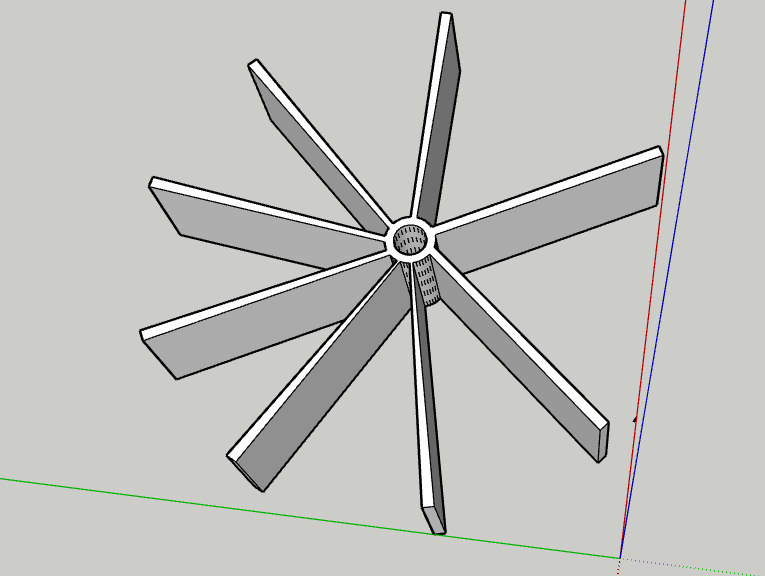
**3.1 3D части на проекта**



*Фигура 11*

**

*Фигура 12*



*Фигура 13*

**3.2 Калибриране и настройки**

Точната работа на дозиращия механизъм зависи от правилната калибрация на стъпковите мотори и задаване на параметри като позиция, брой стъпки и механични ограничения. В този раздел се описва как се извършва калибриране на системата и кои настройки могат да се коригират от потребителя.

Тази стойност определя колко стъпки трябва да направят моторите, за да преминат от един слот (ден) до следващия. Зависи от:

-Типа на стъпковите мотори (напр. 1.8°/стъпка).

-Микростъпките, зададени на драйвера (1/16, 1/8 и т.н.).

-Механичното предаване (зъбни колела, ремъци, винтове).

-Разстоянието между слотовете (в мм или градуси).

Пример: Ако моторът прави 200 стъпки на оборот и използва 1/8 микростъпки, един оборот = 1600 стъпки. Ако разстоянието между слотовете е 1.25 оборота, stepsPerSlot = 1600 × 1.25 = 2000.

Проектът не използва крайни сензори за референтна позиция, затова е важно да се:

-Започва от механично определено начално положение.

-Моторите да се позиционират ръчно в „слот 0“ (понеделник) при стартиране.

-След това Arduino следи позицията на база на командите за движение.

Настройките на моторите се извършват в setup():

Тези стойности се настройват според:

-Теглото на подвижната част.

-Желаната скорост на дозиране.

-Стабилността на движение.

Как да тествате:

1. Увеличете постепенно setMaxSpeed() и setAcceleration().
2. Наблюдавайте дали моторът не пропуска стъпки или не трепери.
3. Намалете, ако движението е нестабилно.

Калибрирането и настройките са от съществено значение за точната работа на системата. Те трябва да се изпълняват:

-при първоначално пускане,

-след механични промени,

-при смяна на мотори или драйвери.

Правилната настройка на stepsPerSlot, скоростите и началната позиция гарантират, че дозаторът ще работи надеждно и точно всеки ден от седмицата.

**3.3 Тестване на системата**

Тестването е важен етап от реализацията на проекта и гарантира, че хардуерът и софтуерът функционират правилно и надеждно при реални условия. В този раздел ще бъдат разгледани поетапно методите за проверка на отделните модули и цялостната система.

**3.3.1 Първоначално включване**

Цел: Да се провери дали системата стартира коректно след подаване на захранване.

Очакван резултат:

-LCD дисплеят се включва и показва съобщението

-Няма движение на моторите при стартиране.

-Не се наблюдават необичайни шумове или затопляне на драйверите.

**3.3.2 Тест на LCD дисплея**

Наблюдаване на съобщенията при промяна на деня и при дозиране.

Проверка дали LCD модулът показва правилните стойности, без дефекти.

Очакван резултат:

-Всички редове и колони се визуализират коректно.

-Няма изгорели пиксели или нечетлив текст.

**3.3.3 Тест на бутона**

Метод:

-Кратко натискане променя текущия ден.

-Задържане за >800 ms стартира дозиране.

Очакван резултат:

-При кратко натискане — LCD обновява показания ден.

-При задържане — дисплеят показва „Dispensing“, моторите се движат.

**3.3.4 Тест на стъпковите мотори**

Метод:

-Активиране на дозиране чрез дълго натискане на бутона.

-Наблюдаване на всяка ос (X, Y, Z).

Очакван резултат:

-Моторите се движат синхронно до изчислената позиция.

-Липса на пропускане на стъпки или вибрации.

-Механичните части не се блокират.

Важно: Ако се използват драйвери тип DRV8825/A4988, настрой потенциометъра за ток на мотора, за да избегнеш прегряване или липса на мощност.

**3.3.5 Тест на EEPROM паметта**

Метод:

1. Избери ден.
2. Изключи и включи отново захранването.
3. Провери дали системата „помни“ избрания ден.

Очакван резултат:

-След рестарт системата показва същия ден, който е бил зададен преди изключване.

**3.3.6 Тест на цялостната система**

Метод:

-Симулиране на седмица с ръчно превключване на дни и дозиране.

-Проверка на всички комбинации: понеделник до неделя.

Очакван резултат:

-Всеки ден коректно премества механизмите.

-Всички изходни позиции съвпадат със съответните „слотове“.

-LCD показва точна и разбираема информация.

Тестването потвърждава, че системата е напълно функционална и готова за реална употреба. Препоръчително е да се провежда тест след всяка промяна по софтуера или хардуера. Също така, преди продължителна експлоатация – извършете 7-дневен симулационен тест за максимална сигурност.

# ЗАКЛЮЧЕНИЕ

Проектът за устройство с вградена система за разпределение на таблетки (хапчета) представлява важна стъпка към осигуряване на по-добра и ефективна грижа за здравето на пациентите, особено на тези с хронични заболявания или специални нужди. Разработването на подобно устройство не само решава проблема с грешки при дозиране на медикаменти, но също така повишава безопасността и удобството за потребителите.

Използването на вградени системи в този контекст предоставя значителни предимства, като автоматизация, минимално потребление на енергия, надеждност и висока точност при разпределянето на таблетките. Това позволява на устройството да работи безпроблемно в продължителен период от време, като същевременно гарантира лесно управление и интеграция в ежедневието на потребителя.

Съществуването на функционален и надежден диспенсър за хапчета може да промени значително начина, по който пациентите следват лекарските предписания, минимизирайки рисковете от пропускане на дозировки и небрежност при приема на лекарства. Възможността за добавяне на интелигентни функции като напомняния, дистанционно наблюдение и контрол чрез мобилно приложение добавя още едно ниво на комфорт и сигурност за потребителите.

В заключение, с развитието на технологиите в сферата на вградените системи и медицината, бъдещето на здравните устройства като това за разпределение на хапчета изглежда обещаващо и ще играе съществени роли в подобряване на качеството на живот на потребителите, които разчитат на редовния прием на лекарства. Проектът поставя основите на подобни иновации и показва значителния потенциал на вградените системи за решаване на критични здравословни проблеми.

# ИЗПОЛЗВАНА ЛИТЕРАТУРА

1. https://www.arduino.cc
2. https://3dprintingindustry.com
3. https://all3dp.com
4. https://www.autodesk.com/products/fusion-360
5. https://help.prusa3d.com
6. https://ocw.mit.edu
7. https://www.arduino.cc/reference/en/libraries/encoder/
8. https://learn.adafruit.com/rotary-encoders
9. https://learn.sparkfun.com
10. https://osoyoo.com/2017/04/07/arduino-nano-cnc-shield-v4-0a4988/
11. https://www.circuitbread.com/tutorials/what-is-the-i2c-communication-protocol
12. https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/
13. https://www.tutorialspoint.com/embedded\_systems
14. https://www.geeksforgeeks.org
15. https://en.wikipedia.org/wiki/Stepper\_motor
16. https://www.electronics-tutorials.ws/?s=stepper+motor
17. https://docs.arduino.cc/learn/electronics/lcd-displays/
18. https://www.instructables.com/LCD-With-I2C/
19. https://softuni.bg/blog/cpp-development
20. https://www.guru99.com/bg/cpp-tutorial.html
21. https://en.wikipedia.org/wiki/Arduino\_Nano?utm\_source=chatgpt.com
22. <https://www.blikai.com/blog/featured-products/what-arduino-nano-board-is-and-how-it-works>
23. <https://www.elprocus.com/an-overview-of-arduino-nano-board/>

# ПРИЛОЖЕНИЯ

Създадената автоматизирана дозираща система има широк спектър от приложения в различни области на ежедневието, здравеопазването, индустрията и образованието. В този раздел са представени реални сценарии, в които системата може да бъде използвана ефективно.

## Приложение 1 “ Домашно дозиране на медикаменти ”

Описание:  
Подходяща за възрастни хора, болни с хронични заболявания или хора със затруднено придвижване.

Ползи:

-Напомняне и автоматично подаване на хапчета.

-Намалява риска от пропускане на доза.

-Може да се комбинира с Bluetooth за свързване с близки или медицински лица.

Разширения:

-Добавяне на аларма или звуков сигнал.

-Свързване със смартфон за потвърждение на прием.

**Приложение 2 “** **Употреба в здравни и социални заведения”**

Описание:  
Използване в болници, старчески домове и хосписи за дозиране на лекарства или добавки по график.

Ползи:

-Намаляване на работното натоварване на персонала.

-Автоматизация на рутинни процедури.

-По-добър контрол върху приетите медикаменти.

**Приложение 3 “ Дозиране на химикали в лаборатории”**

Описание:  
Прецизно подаване на реагенти в изследователски или учебни лаборатории.

Ползи:

-Автоматизация на повтарящи се експерименти.

-Повишаване на точността и безопасността.

-Може да се използва с различни модули (помпи, клапани).

**Приложение 4 “ Образователни цели”**

Описание:  
Проектът е отличен пример за използване на стъпкови мотори, дисплеи, EEPROM и механичен контрол в учебни среди.

Ползи:

-Подходящ за STEM обучения.

-Демонстрира основни принципи на автоматизация.

-Може да се използва като база за учебни проекти и състезания.

**Приложение 5 “ Индустриално дозиране на малки обеми”**

Описание:  
Система за дозиране на малки партиди в производство – например аромати, подправки, прахообразни вещества.

Ползи:

-Мини автоматизация в нискобюджетни производства.

-Възможност за мащабиране.

-Прецизност при дозиране в хранителна, козметична или фарма индустрия.

Проектът е изключително гъвкав и лесно адаптируем. С малки промени в механиката и софтуера, може да бъде приложен в множество реални ситуации. Това го прави не само образователно ценно, но и практически приложимо решение с голям потенциал.