**Технічне завдання**

**На розробку системи контролю веб сервера** **ng-deck**

**Зміст**

[1.Загальні відомості](#_l07zz8u6xw1q)

[1.1. Об'єкт розробки](#_af73va7sjte0)

[1.2. Область застосування](#_6ysu4ywn04u0)

[1.3. Призначення розробки](#_2xdkqqgh6h0d)

[1.4. Умовне позначення і шифр](#_qz8shkepkxqz)

[1.5. Визначення](#_wiqtyp1amye1)

[1.6. Перелік умовних скорочень](#_8v0sz623u0fd)

[2. Призначення та цілі розробки системи](#_5a1emv1aqu2d)

[2.1. Передумови необхідності ng-deck](#_kro7wtlodoon)

[2.2. Мета створення ng-deck](#_7yiuhe8t3omm)

[2.3. Очікувані результати](#_1o62afw31dsq)

[2.4. Огляд існуючих рішень](#_otb9g4w6bw1m)

[3. Основні вимоги до системи](#_tr7dhblxcuy3)

[3.1. Загальні вимоги до системи](#_bv7xl2xrgqe6)

[3.2. Вимоги до апаратної частини](#_xx725afq5bun)

[3.3. Вимоги до програмної частини](#_n3av2f45qtgc)

[3.4. Нефункціональні вимоги до системи](#_44sdj727nqtb)

[3.5. Функціональні вимоги до системи](#_403aegdmxjyc)

[4. Додаткові параметри Платформи](#_1dnrrkc2oeo2)

[4.1. Вимоги до надійності](#_dee8qtp3pdsk)

[4.2. Вимоги до складу і параметрів технічних засобів](#_iq25iyz994j5)

[4.3. Вимоги до інформаційної та програмної сумісності](#_i8zsp8yy15e0)

[4.4. Додаткові вимоги](#_gtdx1h3bh988)

# 

# **1.Загальні відомості**

## **1.1. Об'єкт розробки**

Система контролю веб сервера “ng-deck” призначена для контролю сервера розробника при написанні Angular проектів.

## **1.2. Область застосування**

Робота з webpack dev-server при розробці Angular проектів. Призначений для розробників, для використання вдома або в офісі.

## **1.3. Призначення розробки**

Система призначена для запуску / перезавантаження / зупинки dev сервера при розробці Angular проектів.

## **1.4. Умовне позначення і шифр**

Повне найменування: Система контролю web-dev сервера “ng-deck”

Умовне позначення (шифр): ng-deck

## **1.5. Визначення**

В цьому документі використовуються такі визначення:

*Веб-застосунок* - сайт

*dev сервер* - локальний сервер який запускається при розробці для тестування і відлагодження Angular проекту, існує лише на комп'ютері на якому його було запущено. Показує сайт в режимі реального часу. Оновлюється при будь якій зміні файлів проекту.

*Angular* - фреймворк, призначений для спрощення написання великих веб застосунків, використовує webpack.

*Webpack* - утиліта яка збирає всі файли проекту в один формат для запуску у web-dev сервері

*Arduino* - плата з мікроконтролером і портами вводу-виводу, призначена для розробки апаратних засобів які взаємодіють з кнопками, дисплеями, клавіатурами та іншими пристроями. Іншими словами - малий комп’ютер.

*Сервіс демон* - утиліта яка працює у фоновому режимі без прямої взаємодії з користувачем.

*Послідовний порт* - інтерфейс передачі даних між пристроями. У сучасних комп'ютерах його роль виконує USB.

## **1.6. Перелік умовних скорочень**

В цьому документі використовуються такі позначення і скорочення:

COM порт - послідовний порт

# 

# **2. Призначення та цілі розробки системи**

## **2.1. Передумови необхідності ng-deck**

Angular використовує webpack dev server, який запускається при розробці, і при зміні файлів проекту він автоматично підхоплює зміни і на гарячу оновлює веб сервер. Наприклад така помилка як 'XXX is not a known element' означає що такого елемента не існує, або неправильно вказані імпорти / експорти, хоча якщо все добре прописано, але ця помилка появляється це вже дивно(таке стається досить часто при створенні нових файлів, і є добре відомою проблемою Angular) . Єдиним рішенням є зайти в термінал, зупинити сервер і знову його запустити щоб він правильно підхопив усі файли для збірки.

## **2.2. Мета створення ng-deck**

Метою створення системи є підвищення комфорту під час розробки та досягнення наступних цілей:

* Автоматизація рутинних задач розробника
* Підвищення швидкості розробки
* Подолання проблеми неправильного підхоплення нових файлів

## **2.3. Очікувані результати**

Результатом має стати система, яка:

* Взаємодіє з комп'ютером по СОМ порту
* Шукає Angular проекти у вказаних директоріях
* Відображає список проектів на дисплеї Arduino
* Дозволяє вибрати проект зі списку(активний проект)
* Дозволяє запустити активний проект з Arduino
* Дозволяє зупинити активний проект з Arduino
* Дозволяє перезапустити активний проект з Arduino
* Відображає додаткову інформацію про активний проект(статус, порт)
* Має просте налаштування

## **2.4. Огляд існуючих рішень**

Єдиним готовим засобом який може частково виконувати функції ng-deck є stream deck - клавіатура з програмованими кнопками. Цей пристрій призначений для ведення прямих трансляцій і дозволяє налаштувати різні дії на натискання кнопок. Його можна би було налаштувати для запуску Angular проекту по натисканні кнопки, але він не може відстежувати стан та перезапускати проект, оскільки не має такого функціоналу. Також серед його недоліків - прив'язаність до Windows та висока ціна(від 400$).

# 

# **3. Основні вимоги до системи**

## **3.1. Загальні вимоги до системи**

Система повинна складатися із двох частин

* Arduino з дисплеєм, енкодером і двома кнопками яка спілкується з комп'ютером по COM порту.
* Сервіс демон який слухає COM порт і виконує дії в залежності від отриманої комнади.

## **3.2. Вимоги до апаратної частини**

В основі апаратної частини Arduino nano, які підключається до комп'ютера по СОМ порту та отримує з нього інформацію та передає команди. Arduino повинна відображати список проектів, обирати проект зі списку, відображати статус, надсилати запити до сервіса про запуск проекту, зупинку проекту, перезапуск проекту. Також апаратна частина повинна мати 2 кнопки та енкодер.

Список проектів повинен відображатися на рідкокристалічному дисплеї 1602. За допомогою энкодера відбувається навігація по списку (вправо - вниз по списку, вліво - вгору по списку). По натисканні на енкодер вибраний проект стає активним і може бути запущений натисканням першої кнопки. Друге натискання тієї ж кнопки повинно перезавантажити проект, про що необхідно повідомити на рідкокристалічному дисплеї. Друга кнопка повинна зупиняти активний проект

## **3.3. Вимоги до програмної частини**

Програмна частина проекту це сервіс який слухає СОМ порт та виконує дії в залежності від отриманих команд чи подій. Програма повинна мати файл конфігурації, який дозволяє вказати директорії, в яких система повинна шукати Angular проекти та можливість додати аргументи для запуску. Сервіс повинен розрізняти валідні Angular проекти поміж інших файлів та директорій.

При отриманні команди від мікроконтролера про запуск проекту система повинна запускати обраний проект у новому потоці для можливості подальшої комунікації з Arduino.

Сервіс повинен зберігати стан апаратної частини на випадок втрати живлення мікроконтролера або випадкового від'єднання кабеля.

## **3.4. Нефункціональні вимоги до системи**

**Платформа повинна забезпечувати виконання наступних нефункціональних вимог апаратної частини:**

* Зручний та зрозумілий інтерфейс
* Відновлення стану після втрати живлення
* Швидкодія
* Відмовостійкість
* Коректне опрацювання натискання кнопок
* Перехід в режим енергозбереження при неактивності
* Захист від некоректних дій

**Платформа повинна забезпечувати виконання наступних нефункціональних вимог програмної частини:**

* Зручний та зрозумілий конфігураційний файл
* Реагувати на створення нових проектів
* Просте встановлення та налаштування
* Низьке споживання системних ресурсів в режимі очікування

## **3.5. Функціональні вимоги до системи**

**Платформа повинна забезпечувати виконання наступних функціональних вимог апаратної частини:**

* Відображення списку проектів - Arduino повинна відображати список проектів отриманих по СОМ порту на дисплеї.
* Вибір проекту - За допомогою энкодера повинна бути можливість обрати активний проект
* Надіслати команду сервісу про запуск проекту - По натисканні кнопки Arduino повинна відправити сигнал сервісу по СОМ порту про запуск активного проекту
* Надіслати команду сервісу про зупинку проекту - По натисканні кнопки Arduino повинна відправити сигнал сервісу по СОМ порту про зупинку активного проекту.
* Надіслати команду сервісу про перезапуск проекту - По натисканні кнопки Arduino повинна відправити сигнал сервісу по СОМ порту про перезапуск активного проекту.
* Відображення статусу - коли вибраний проект запущено на дисплеї повинна відображатись інформація про статус активного проекту та порт на якому запущено сервер.

**Платформа повинна забезпечувати виконання наступних функціональних вимог програмної частини:**

* Пошук проектів за вказаними у конфігураційному файлі шляхами - у конфігураційному файлі потрібно вказати шляхи до директорій за якими система повинна стежити та шукати в них проекти для передачі списку доступних проектів по СОМ порту.
* Запуск проекту - для цього потрібно з кореня директорії проекту виконати команду ng serve що запустить dev сервер.
* Зупинка проекту - для цього потрібно знайти pid активного проекту. Після того як pid знайдено можна зупинити процес.
* Перезапуск проекту - зупинка і запуск активного проекту.

# **4. Додаткові параметри Платформи**

## **4.1. Вимоги до надійності**

* Апаратна система повинна зберігати свій стан у сервісі та повертатися до у випадку втрати живлення.

## **4.2. Вимоги до складу і параметрів технічних засобів**

* Програмне забезпечення повинно функціонувати на IBM-сумісних персональних комп'ютерах.
* Наявність вільної пам'яті для встановлення програми
* Наявність вільного послідовного порту до якого можна підключити пристрій

## **4.3. Вимоги до інформаційної та програмної сумісності**

* Програмне забезпечення повинно функціонувати на операційних системах сімейства GNU / Linux
* Програмне забезпечення повинне працювати з Angular 9 i вище
* Наявність JVM сумісної з версією сервіса
* Надання сервісу прав доступу до папок з проектами

## **4.4. Додаткові вимоги**

* Зібрати виконавчий JAR файл