

ДОДАТОК А

Програмний код реалізації системи очищення СП

```
int pumpPin1 = 6;
int pumpPin2 = 5;
int motorWheelPin1 = 8;
int motorWheelPin2 = 7;
int motorBrushPin1 = 3;
int motorBrushPin2 = 4;
int trigPin1 = 12;
int echoPin1 = 13;
int trigPin2 = 10;
int echoPin2 = 11;
int powerPin = 2;

bool movingForward = true;
int sonarTriggerCount = 0;
bool systemActive = false;
bool initialStart = false;
bool allowRestart = false;

void setup() {
  pinMode(pumpPin1, OUTPUT);
  pinMode(pumpPin2, OUTPUT);
  pinMode(motorWheelPin1, OUTPUT);
  pinMode(motorWheelPin2, OUTPUT);
  pinMode(motorBrushPin1, OUTPUT);
  pinMode(motorBrushPin2, OUTPUT);
  pinMode(trigPin1, OUTPUT);
  pinMode(echoPin1, INPUT);
```

```
pinMode(trigPin2, OUTPUT);
pinMode(echoPin2, INPUT);
pinMode(powerPin, INPUT_PULLUP);
}

void loop() {
  checkPowerSwitch();
  if (systemActive) {
    runSystem();
  } else {
    stopAll();
    if (allowRestart) {
      checkForRestart();
    }
  }
}

void checkPowerSwitch() {
  static bool lastSwitchState = digitalRead(powerPin);
  bool currentSwitchState = digitalRead(powerPin);

  if (currentSwitchState != lastSwitchState) {
    delay(50);
    currentSwitchState = digitalRead(powerPin);
    if (currentSwitchState != lastSwitchState) {
      if (!systemActive && allowRestart) {
        systemActive = true;
        resetSystem();
      } else {
        systemActive = !systemActive;
      }
    }
  }
}
```

```
    }  
    lastSwitchState = currentSwitchState;  
  }  
}  
}
```

```
void checkForRestart() {  
  if (digitalRead(powerPin) == LOW) {  
    delay(50);  
    if (digitalRead(powerPin) == LOW) {  
      systemActive = true;  
      resetSystem();  
    }  
  }  
}
```

```
void resetSystem() {  
  initialStart = false;  
  movingForward = true;  
  sonarTriggerCount = 0;  
  allowRestart = false;  
}
```

```
void runSystem() {  
  if (!initialStart) {  
    startBrushMotorOnly();  
    initialStart = true;  
    delay(3000);  
  }  
}
```

```
if (sonarTriggerCount < 6) {
    moveBrushAndWheel();
    pumpSequence();
    checkDistanceAndAdjust(trigPin1, echoPin1);
    checkDistanceAndAdjust(trigPin2, echoPin2);
} else {
    stopAll();
    allowRestart = true;
}
}

void startBrushMotorOnly() {
    digitalWrite(motorBrushPin1, HIGH);
    digitalWrite(motorBrushPin2, LOW);
}

void moveBrushAndWheel() {
    digitalWrite(motorWheelPin1, movingForward ? HIGH : LOW);
    digitalWrite(motorWheelPin2, movingForward ? LOW : HIGH);
    digitalWrite(motorBrushPin1, movingForward ? HIGH : LOW);
    digitalWrite(motorBrushPin2, movingForward ? LOW : HIGH);
}

void pumpSequence() {
    digitalWrite(pumpPin1, LOW);
    digitalWrite(pumpPin2, HIGH);
    delay(650);
    digitalWrite(pumpPin1, LOW);
    digitalWrite(pumpPin2, LOW);
    delay(5000);
}
```

```
}
```

```
void stopAll() {  
    digitalWrite(motorWheelPin1, LOW);  
    digitalWrite(motorWheelPin2, LOW);  
    digitalWrite(motorBrushPin1, LOW);  
    digitalWrite(motorBrushPin2, LOW);  
    digitalWrite(pumpPin1, LOW);  
    digitalWrite(pumpPin2, LOW);  
}
```

```
void checkDistanceAndAdjust(int trigPin, int echoPin) {  
    digitalWrite(trigPin, LOW);  
    delayMicroseconds(2);  
    digitalWrite(trigPin, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(trigPin, LOW);  
    long duration = pulseIn(echoPin, HIGH);  
    long distance = duration / 58.2;  
    if (distance > 50) {  
        stopAll();  
        delay(3000);  
        movingForward = !movingForward;  
        sonarTriggerCount++;  
        Serial.print("Distance: ");  
        Serial.print(distance);  
    }  
}
```


ДОДАТОК В

Схема електрична принципова розроблюваної системи

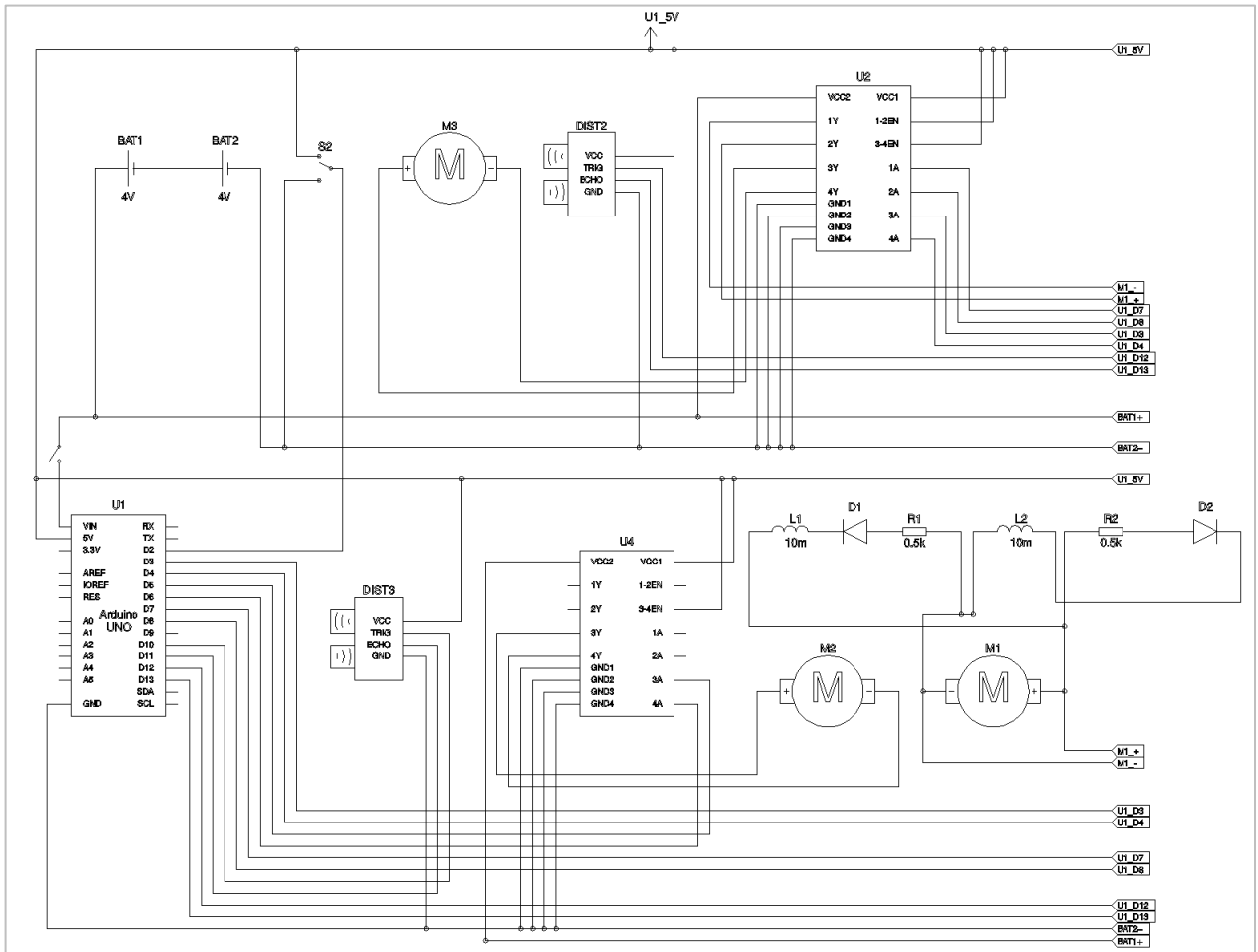


Рисунок В.1 – Зображення ескізу електричної принципової схеми розроблюваної системи

ДОДАТОК Г

Демонстраційний матеріал

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Кафедра КІТАР

КВАЛІФІКАЦІЙНА РОБОТА

На тему: Розробка системи автоматизації для технологічного процесу очищення
поверхні сонячної панелі від забруднень

Виконав:
студент групи АКТАКІТ-20-1
Гайдук Ілля Михайлович

Керівник:
доцент кафедри КІТАР
Жарікова Ірина Володимирівна

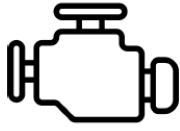
1

Актуальність роботи

- Розвиток альтернативних джерел енергії
- ↓
- Розвиток автоматизованих очищувальних систем
- ↓
- Досягнення максимальної ефективності очищення поверхонь сонячних панелей

2

Етапи створення очищувальної системи



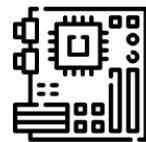
Обрані елементи



3Д-модель



Розрахунки



Макет схеми



Програмний код

3

Основні електронні та механічні компоненти



Arduino Uno



Занурювальний водяний насос



Гумове колесо



Датчик HC-SR04



Двигун JGB37-520-068



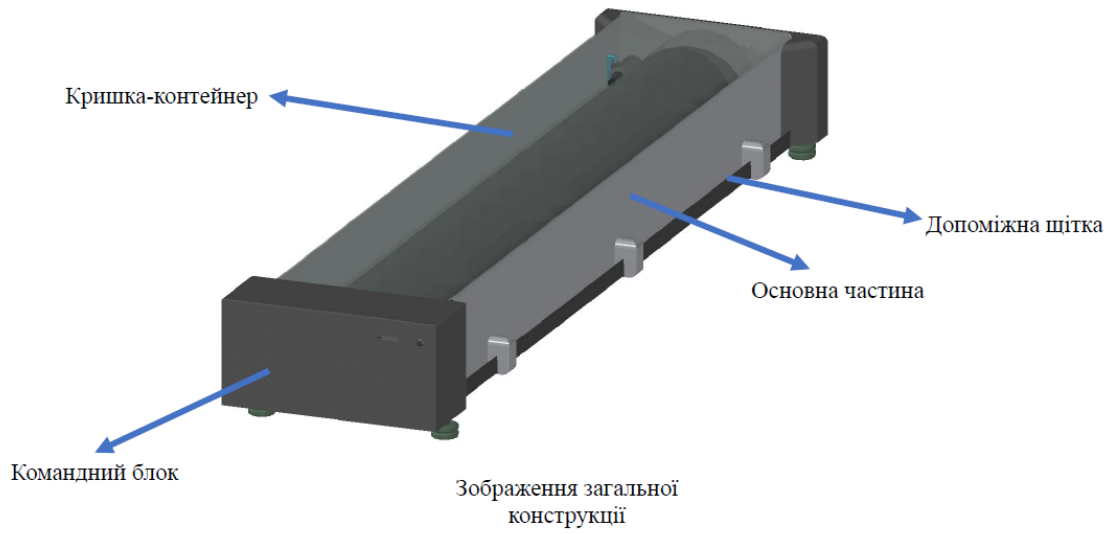
Драйвер DRV8833



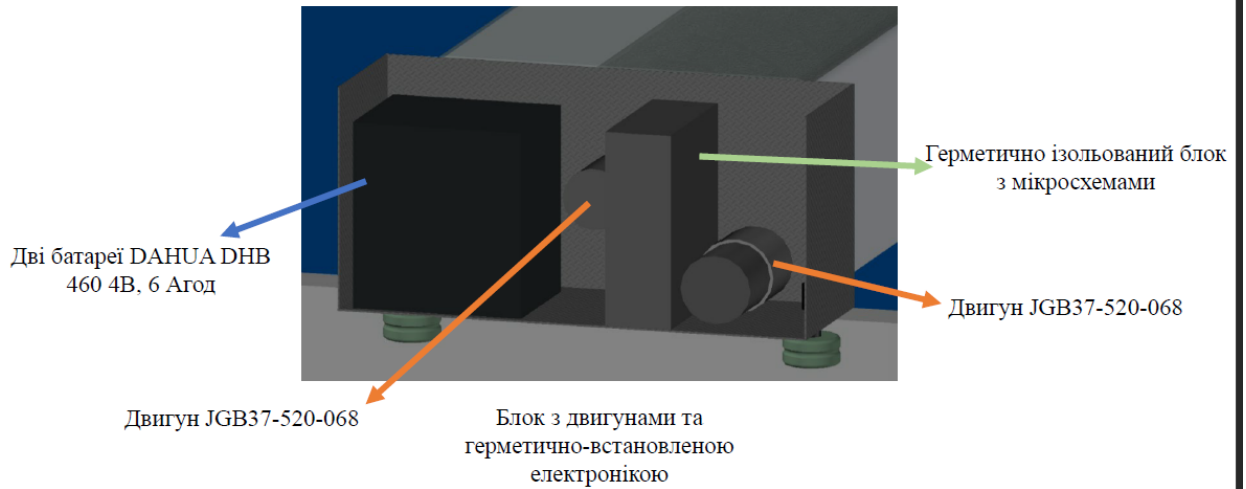
Шітка

4

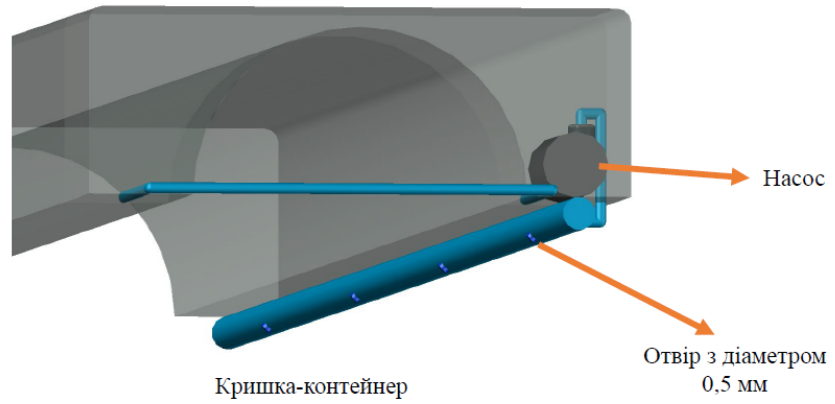
Основна функціональність автоматизованої системи



Опис командної частини

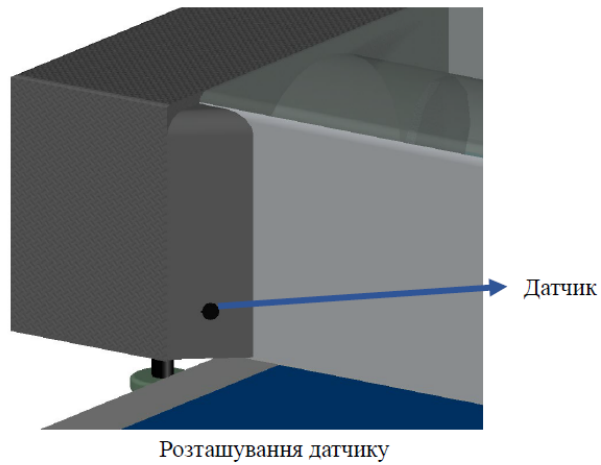


Опис кришки-контейнеру



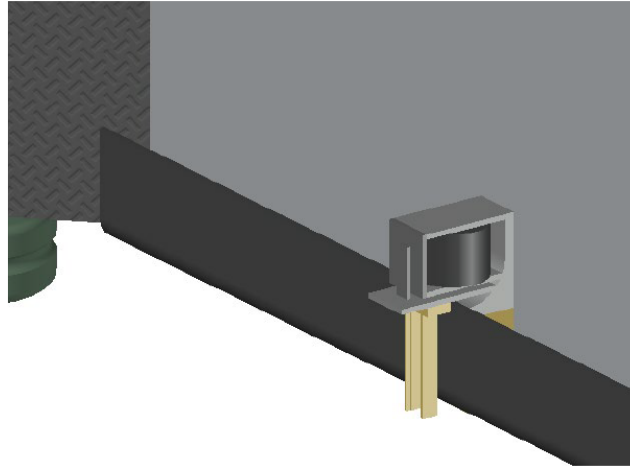
7

Опис розташування та принцип роботи датчика



8

Опис допоміжних щіток



Механізм регулювання
положенням допоміжної
щітки

9

Створення віртуального макету схеми підключення

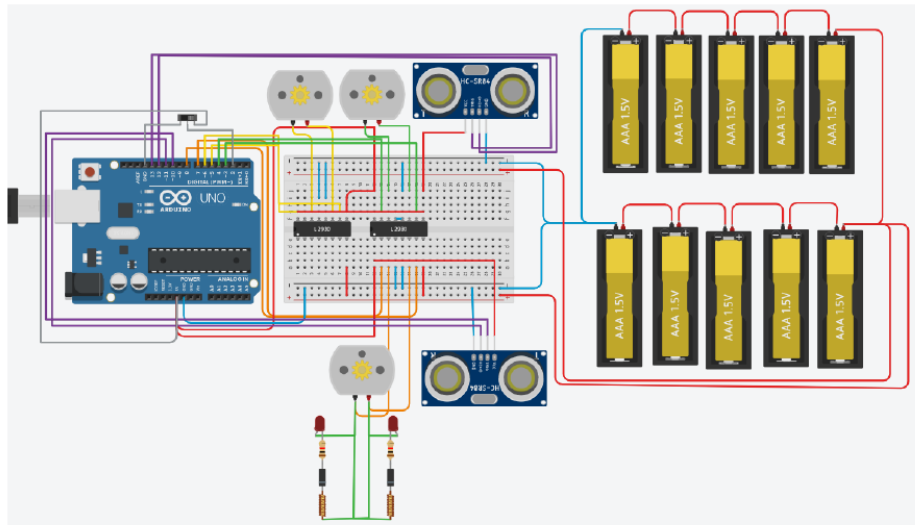
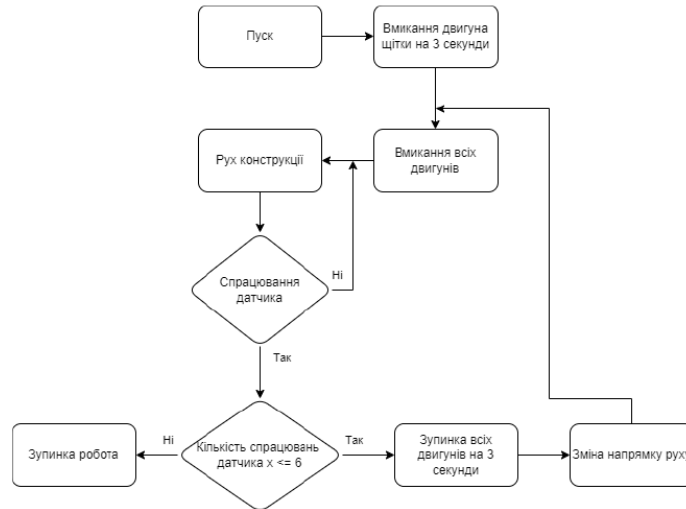


Схема підключення
віртуального макету

10

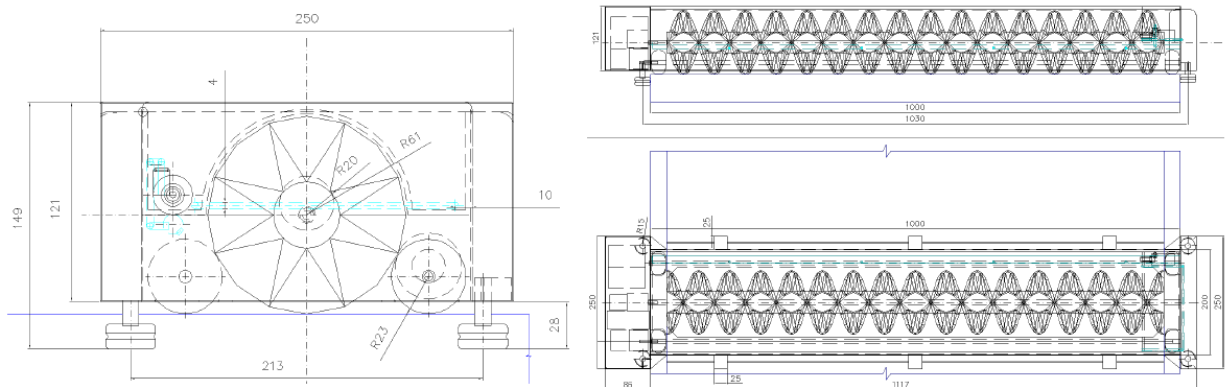
Алгоритм роботи



Алгоритм роботи
автоматизованої системи

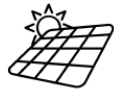
11

Кресленик



12

Майбутні вдосконалення



Інтегровані сонячні панелі



Додаткові датчики



Дистанційне керування

13

Висновки

У кваліфікаційній роботі здійснено аналіз існуючих методів і систем очищення сонячних панелей. Було обрано програмне середовище AutoCAD та розроблено 3D-модель, ескіз кресленика та електричну принципову схему системи автоматизації очищення, а також вибрано необхідні апаратні модулі, матеріали та компоненти.

Проведено необхідні розрахунки супротиву шіток до руху конструкції. Знайдено необхідний крутний момент, за яким обрано оптимальні двигуни. Розраховано необхідну ємність акумулятора для забезпечення тривалості роботи автоматизованої системи на протязі 3 годин без перерв.

Проведено аналіз стійкості механічної частини головного двигуна за критерієм Рауса. Аналіз показав, що всі три умови за критерієм виконуються, тому механічну частину двигуна можна вважати стійкою.

Було обрано програмне середовище TinkerCAD, в якому створено віртуальний макет схеми автоматизованої системи.

Програмна реалізація автоматизованої системи очищення була здійснена з використанням мови програмування C. Розроблено слідкуючу систему, яка основана на двох інфрачервоних датчиків сонару. Створено алгоритм роботи автоматизованої системи.

14

