

ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

Харківський національний університет радіоелектроніки
Кафедра ЕОМ



Система моніторингу характеристик руху мобільного колісного робота на базі датчику одометру

Кваліфікаційна робота
Другий (магістерський) рівень

Автор:
Жуков С.В.
Студ.гр.СПм-21-2

Керівник:
КАРГІН А.О.

Актуальність



► Сьогодні мобільні колісні роботи знаходять широке застосування в різних галузях, таких як промисловість, медицина, автомобільна та військова промисловість, логістика та інші. У зв'язку з цим, на ринку існує велика кількість аналогів роботів з різними характеристиками та можливостями. У загальному, роботи відіграють важливу роль в автоматизації процесів та полегшенні праці людей в різних сферах діяльності.

Актуальність мобільних колісних роботів в сучасному світі

- Гнучкість і маневреність
- Автономність
- Робота у важких умовах
- Використання у виробництві
- Медичні застосування

Постанова задачі

Мета дослідження:

Виявлення можливості використання датчика одометру для руху мобільного колісного роботу. Проаналізувати апаратну базу для виконання розробки та визначення необхідних компонентів для її реалізації. Проведення аналізу технологій для вирішення поставленої задачі, визначення переваг та недоліків різних методів.

Для досягнення поставленої мети потрібно вирішити наступні завдання:

1. Дослідити можливості датчика одометру та його аналогів.
2. Провести огляд наукових джерел та існуючих систем моніторингу руху мобільних роботів, що використовують датчики одометру.
3. Визначити та описати параметри руху роботу.
4. Провести опис компонентів реалізації.



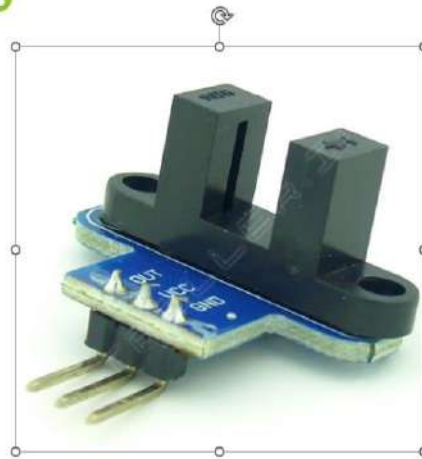
NURE
Національний університет
радіоелектроніки

Існуючі мобільні колісні роботи



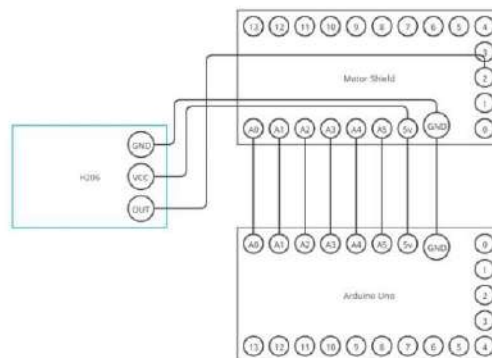
NURE
Національний університет
радіоелектроніки

Датчики одометру



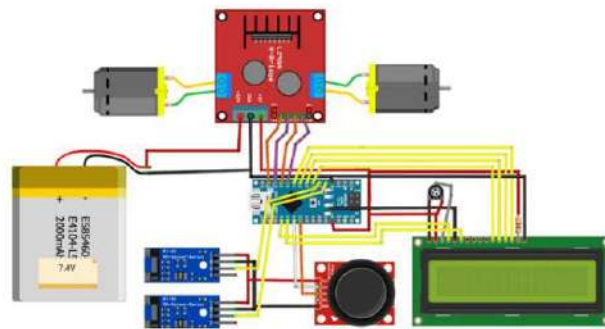
NURE
Національний університет
радіоелектроніки

Схема з'єднання компонентів на базі датчика H206



NURE
Національний університет
радіоелектроніки

Фізична схема мобільного роботу



Методи програмного управління

Жорстке програмне управління	Гнучке програмне управління	Гнучке ситуаційне програмне управління
<i>вперед</i>	<i>впередK1 S=120</i>	<i>впередK1 R=f,S=50,V=1</i>
<i>затримка $t_{4, \omega, 3}$</i>	<i>праворуч $\gamma=90^\circ$</i>	<i>праворуч $\gamma=90^\circ$</i>
<i>вимк_вперед</i>	<i>впередK1 S=500</i>	<i>впередK2 R=h,S=50,V=2</i>
<i>праворуч</i>	<i>ліворуч $\gamma=90^\circ$</i>	<i>ліворуч $\gamma=90^\circ$</i>
<i>затримка t_{90°</i>	<i>впередK1 S=120</i>	<i>впередK1 R=k,S=50,V=1</i>
<i>вимк_праворуч</i>	<i>стоп</i>	<i>стоп</i>
<i>вперед</i>		
<i>затримка $t_{1, 3}$</i>		
<i>вимк_вперед</i>		
<i>ліворуч</i>		
<i>затримка t_{90°</i>		
<i>вперед</i>		
<i>затримка $t_{4, \delta, 1}$</i>		
<i>вимк_вперед</i>		
<i>стоп</i>		



Методи керування переміщенням роботи

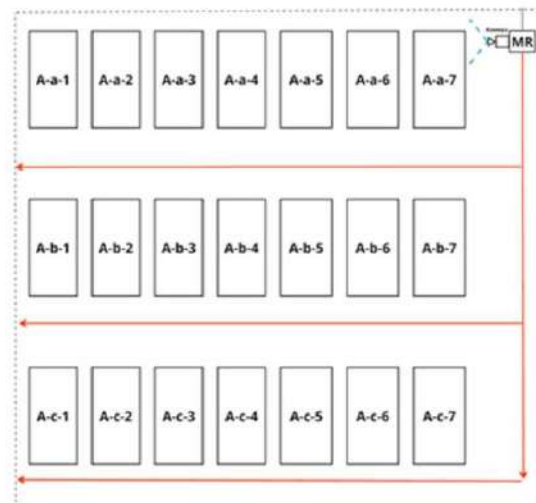
Основні методи управління включають в себе:

- жорстке програмне управління;
- гнучке програмне управління;
- гнучке ситуаційне програмне управління.

Для реалізації цих управляючих програм було використано середовище MatLab.

Кожен метод демонструє очікувану поведінку робота, здатність виконувати різні рухи та змінювати параметри руху залежно від ситуації.

Схема руху роботи



ВИСНОВКИ

- ▶ У порівнянні з існуючими аналогами, система моніторингу характеристик руху мобільного колісного роботу на базі датчика одометру має свої переваги, такі як висока точність вимірювань, незалежно від ходу.
- ▶ Була проаналізована апаратна база для виконання розробки та визначено необхідні компоненти для її реалізації. Також було проведено аналіз технологій для вирішення поставленої задачі, визначено переваги та недоліки різних методів збору, обробки та аналізу даних.
- ▶ У ході аналізу методологічного підґрунтя було визначено, що для успішної реалізації системи моніторингу потрібно враховувати фізичні властивості руху мобільного колісного роботу та використовувати різні методи обробки та аналізу даних для отримання точної та повної інформації про рух.
- ▶ Система моніторингу характеристик руху мобільного колісного роботу на базі датчика одометру є важливою технологією для вимірювання та аналізу рухових параметрів роботів. Для успішної реалізації системи потрібно враховувати апаратну базу, технології збору, обробки та аналізу даних, а також фізичні властивості руху мобільного колісного роботу. Правильне використання цієї системи може допомогти в підвищенні продуктивності та ефективності роботів, також покращенні безпеки праці та економії ресурсів.



ДОДАТОК Б

КОД СИСТЕМИ

```

#define LM_pos 9          // left motor
#define LM_neg 8          // left motor
#define RM_pos 10         // right motor
#define RM_neg 11        // right motor
#define joyX A2
#define joyY A3
#include <LiquidCrystal.h>
const int rs = 14, en = 15, d4 = 4, d5 = 5, d6 = 6, d7 = 7;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
int left_intr = 0;
int right_intr = 0;
int angle = 0;
/*Hardware details*/
float radius_of_wheel = 0.033;
volatile byte rotation;
float timetaken,rpm,dttime;
float v;
int distance;
unsigned long pevertime;
void setup()
{
  rotation = rpm = pevertime = 0; //Initialize all variable to
  zero
  Serial.begin(9600);
  lcd.begin(16, 2); //Initialise 16*2 LCD
  lcd.print("Vechile Monitor");
  lcd.setCursor(0, 1);
  lcd.print("-CircuitDigest ");
  delay(2000);
  lcd.clear();
  lcd.print("Lt:      Rt:      ");
  lcd.setCursor(0, 1);
  lcd.print("S:      D:  A:      ");
  pinMode(LM_pos, OUTPUT);
  pinMode(LM_neg, OUTPUT);
  pinMode(RM_pos, OUTPUT);
  pinMode(RM_neg, OUTPUT);
  digitalWrite(LM_neg, LOW);
  digitalWrite(RM_neg, LOW);

  attachInterrupt(digitalPinToInterrupt(2), Left_ISR, CHANGE);
  attachInterrupt(digitalPinToInterrupt(3), Right_ISR, CHANGE);
}
void loop()
{
  int xValue = analogRead(joyX);

```

```

int yValue = analogRead(joyY);

int acceleration = map (xValue, 500, 0, 0, 200);
if (xValue<500)
{
analogWrite(LM_pos, acceleration);
analogWrite(RM_pos, acceleration);
}
else
{
analogWrite(LM_pos, 0);
analogWrite(RM_pos, 0);
}
if (yValue>550)
analogWrite(RM_pos, 80);
if (yValue<500)
analogWrite(LM_pos, 100);
if(millis()-dtime>500)
{
rpm= v = 0;
dtime=millis();
}
v = radius_of_wheel * rpm * 0.104;
distance = (2*3.141*radius_of_wheel) * (left_intr/40);
int angle_left = (left_intr % 360) * (90/80);
int angle_right = (right_intr % 360) * (90/80);
angle = angle_right - angle_left;
lcd.setCursor(3, 0); lcd.print(" "); lcd.setCursor(3, 0);
lcd.print(left_intr);
lcd.setCursor(11, 0); lcd.print(" "); lcd.setCursor(11,
0);lcd.print(right_intr);
lcd.setCursor(2, 1); lcd.print(" "); lcd.setCursor(2,
1);lcd.print(v);
lcd.setCursor(9, 1); lcd.print(" "); lcd.setCursor(9,
1);lcd.print(distance);
lcd.setCursor(13, 1); lcd.print(" "); lcd.setCursor(13,
1);lcd.print(angle);
delay(100);
lcd.setCursor(3, 0); lcd.print(" "); lcd.setCursor(3, 0);
lcd.print(left_intr);
lcd.setCursor(11, 0); lcd.print(" "); lcd.setCursor(11,
0);lcd.print(right_intr);
lcd.setCursor(2, 1); lcd.print(" "); lcd.setCursor(2,
1);lcd.print(v);

}
void Left_ISR()
{
left_intr++;delay(10);
}
void Right_ISR()
{

```

```
right_intr++; delay(10);

rotation++;
dtime=millis();
if(rotation>=40)
{
    timetaken = millis()-pevtime;
    rpm=(1000/timetaken)*60;
    pevtime = millis();
    rotation=0;
}
}
```

ДОДАТОК В

РЕАЛІЗАЦІЯ ДОДАТКОВИХ ДАТЧИКІВ

В.1 Фунція виведення

```

void setup()
{
  rotation = rpm = pevtime = 0; //Initialize all variable to
zero
  Serial.begin(9600);
  lcd.begin(16, 2); //Initialise 16*2 LCD
  lcd.print("Bot Monitor"); //Intro Message line 1
  lcd.setCursor(0, 1);
  lcd.print("-CircuitDigest "); //Intro Message line 2
  delay(2000);
  lcd.clear();
  lcd.print("Lt:      Rt:      ");
  lcd.setCursor(0, 1);
  lcd.print("S:      D:  A:      ");

  pinMode(LM_pos, OUTPUT);
  pinMode(LM_neg, OUTPUT);

  pinMode(RM_pos, OUTPUT);
  pinMode(RM_neg, OUTPUT);

  digitalWrite(LM_neg, LOW);
  digitalWrite(RM_neg, LOW);

  attachInterrupt(digitalPinToInterrupt(2), Left_ISR, CHANGE);
//Left_ISR is called when left wheel sensor is triggered
  attachInterrupt(digitalPinToInterrupt(3), Right_ISR,
CHANGE); //Right_ISR is called when right wheel sensor is
triggered
}

```

В.2 Цикл реалізації напрямку

1.2

```

int xValue = analogRead(joyX);
int yValue = analogRead(joyY);

int acceleration = map (xValue, 500, 0, 0, 200);

if (xValue<500)

```

```
{
analogWrite(LM_pos, acceleration);
analogWrite(RM_pos, acceleration);
}

else
{
analogWrite(LM_pos, 0);
analogWrite(RM_pos, 0);
}

if (yValue>550)
analogWrite(RM_pos, 80);

if (yValue<500)
analogWrite(LM_pos, 100)
```