Міністерство освіти і науки України

**Прикарпатський національний університет**

**імені В.Стефаника**

*Факультет математики та інформатики*

*Кафедра інформаційних технологій*

*Програмування вбудованих систем*

Лабораторна робота № 5 - 6

Тема: «Вивчення та реалізація алгоритмів керування колісним роботом »

Варіант : **9**

Виконав: **Рижкін О. К.**

Група ІПЗ-23

Дата: 16 Травня 2024р.

Викладач: Лазарович І.М.

Івано-Франківськ – 2024

**Мета:** Вивчити принципи роботи ПІД-регулятора та його складових компонент

**Завдання з лабораторної 1:**

Нам потрібно розробити трьохколісного робота з такими параметрами:

1. Варіант – 9.

2. Максимальна швидкість (км/год) – 1.

3. Кількість коліс на платформі – 3.

4. Час автономної роботи (год) – 2.

5.Максимальне корисне навантаження – 700гр.

5. Рух робота – По чорній лінії з контролем швидкості.

**5.1 Завдання для виконання**

Використовуючи теоретичний матеріал лекцій 5-6 для індивідуального завдання з розробки колісного робота (згідно лаб.1) розробити програму керування роботом з використанням:

1. Пропорційного керування;

2. Пропорційно-диференційного керування;

3. Пропорційно-інтегрального керування;

4. ПІД-регулювання.

При розробці програми врахувати наступну схему підключення:

1. Датчики (згідно варіанту): піни А0, A1, A2, А3, 2;

2. Bluetooth SoftwareSerial – піни 10 – RX, 11 – TX

3. Ходові двигуни з можливістю ШІМ-регулювання - піни 5 (ліве колесо, EN), 6 (праве колесо, EN), сигнал вперед лівий - 7, назад лівий - 8, сигнал вперед правий - 3, назад правий - 4;

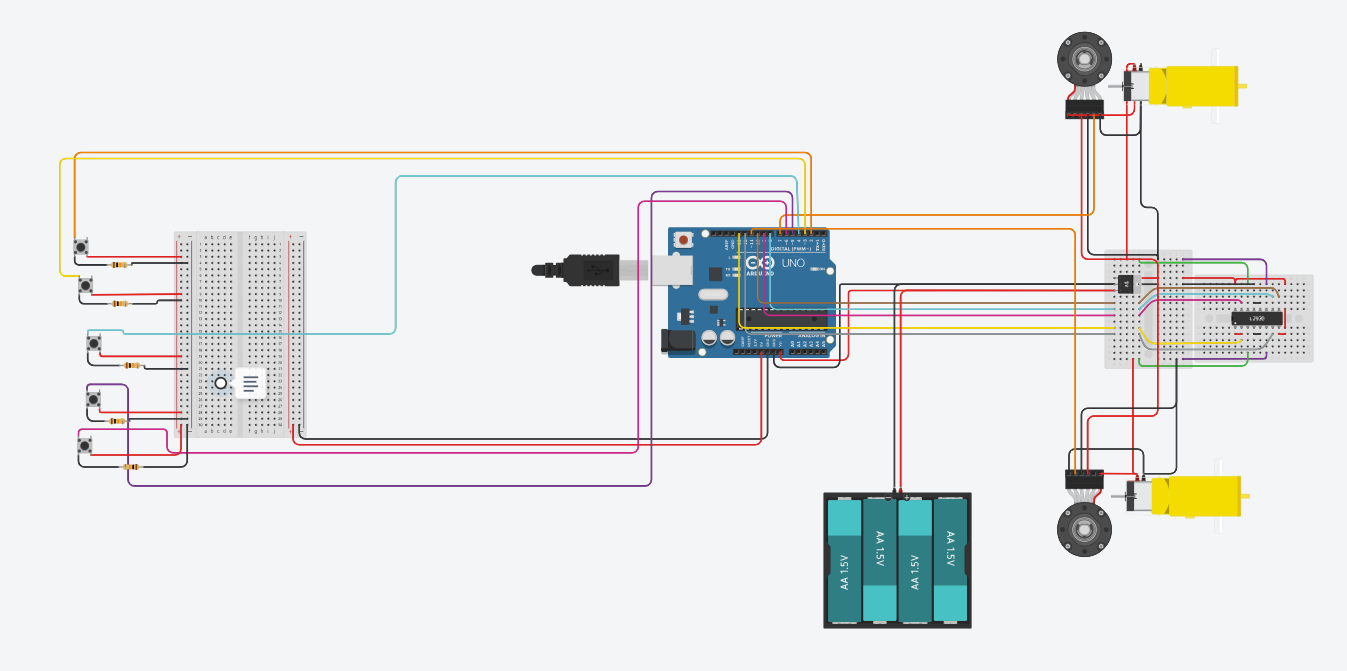
4. Контроль швидкості (за потреби) – піни 12 (ліве колесо), 13 (праве колесо);

5. Сервопривід руля (за потреби) – пін 9.

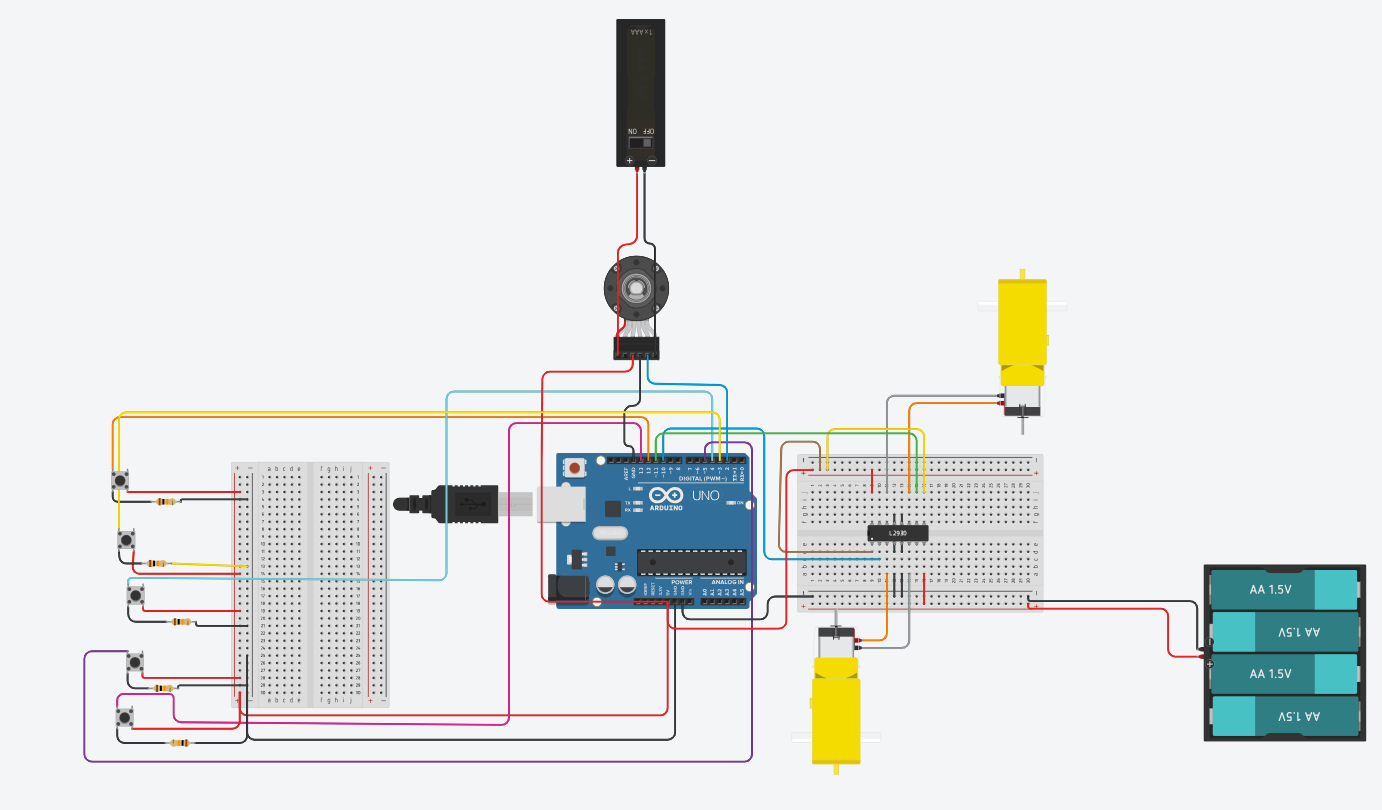
6. Гіроскоп/акселерометр MPU-6050 (за потреби) – піни А4(SDA) і А5 (SCL)

Забезпечити можливість передавання (перемикання) способу керування і параметрів регулювання через Bluetooth-модуль HC-05 (через SoftwareSerial). Для передавання способу керування передбачити такі (власні) команди: pControl, pdControl, piControl, pidControl, або аналогічні. Для передавання параметрів регулювання можна використати такі команди: Kp=8.5, або Kd=0.01, або Ki=0.5, чи аналогічні. Після отримання цих параметрів їх потрібно зберегти в EEPROM (і відповідно зчитати їх в функції setup). Також передбачити команди для початку і завершення руху робота (start i stop).

**Схема робота з лабораторної 2:**



**Відредактована схема з лабораторної 5 -6:**

****

**https://www.tinkercad.com/things/l4ZNbMiqCJ3-copy-of-lab2/editel?sharecode=znkkWmstEYCuhokdy\_dZQ5h23gHHbOx2cc9WilVV7UY**

#include <EEPROM.h>

#define RIGHT\_MOTOR 10

#define LEFT\_MOTOR 11

#define encoderPin 2

//затримака перевірки відхилення і зміни курсу

#define per 200

unsigned long currentMillis;

unsigned long previousMillis = 0;

int diff = 0;

int speed = 65;

float rpm = 0;

boolean isStop = false;

int desiredRpm = 150; // швидкість якої хочемо досягнути

float kp = 10; // Пропорційний коефіцієнт

float kpS = 1; // Для швидкості

float kd = 5; // Диференціальна

float ki = 2; // Інтегральна

int controlMode = 1;// вид контролю 1-3

String inputString = ""; // Рядок для зберігання вхідних даних з серійного порту

int prevSpeedErr = 0;

int prevDiffErr = 0;

long integSpeedErr = 0;

long integDiffErr = 0;

volatile unsigned int pulseCount = 0;

int position =0;

// Верхня та нижня межа

int mRound(int n, int b1, int b2) {

if (n > b1) n = b1;

if (n < b2) n = b2;

return n;

}

void runMotors() {

if(!isStop){ //пуск

speed = mRound(speed, 255, 0); // зменшуємо діапазон зачення щоб не ввести незрозумілі значення

analogWrite(LEFT\_MOTOR, mRound(speed - diff, 255, 0));

analogWrite(RIGHT\_MOTOR, mRound(speed + diff, 255, 0));

}

}

void readFromEEPROM() {

if(EEPROM.read(sizeof(float) \* 3 + sizeof(int)\*2)){ // if!=0 was save

EEPROM.get(0, kp);

EEPROM.get(sizeof(float), kd);

EEPROM.get(sizeof(float) \* 2, ki);

EEPROM.get(sizeof(float) \* 3, controlMode);

EEPROM.get(sizeof(float) \* 3 + sizeof(int) , desiredRpm);

Serial.print("kp: "); Serial.println(kp);

Serial.print("kd: "); Serial.println(kd);

Serial.print("ki: "); Serial.println(ki);

Serial.print("Control Mode: "); Serial.println(controlMode);

}

}

void saveToEEPROM() {

EEPROM.put(0, kp);

EEPROM.put(sizeof(float), kd);

EEPROM.put(sizeof(float) \* 2, ki);

EEPROM.put(sizeof(float) \* 3, controlMode);

EEPROM.put(sizeof(float) \* 3 + sizeof(int), desiredRpm);

EEPROM.put(sizeof(float) \* 3 + sizeof(int)\*2, true);

}

void stopMotors() {

digitalWrite(RIGHT\_MOTOR, LOW);

digitalWrite(LEFT\_MOTOR, LOW);

isStop = true;

}

void serialEvent() { // коли користувач вводить щось через блютуз\консоль працює ця функція

while (!Serial.available()) {}

inputString = Serial.readString();

if (inputString == "start")isStop = false;

if (inputString == "stop")stopMotors();

if (inputString.startsWith("$")) {

if (inputString.startsWith("$kp=")) kp = inputString.substring(4).toFloat();

if (inputString.startsWith("$kd=")) kd = inputString.substring(4).toFloat();

if (inputString.startsWith("$ki=")) ki = inputString.substring(4).toFloat();

if (inputString.startsWith("$control=")) controlMode = inputString.substring(9).toInt();

if (inputString.startsWith("$rpm=")) desiredRpm = inputString.substring(5).toInt();

saveToEEPROM();

}

}

void proportionalControl() {

diff = kp \* position;

speed = kpS \* (desiredRpm - rpm);

}

void propDifControl() {

diff = kp \* position + kd \* (position - prevDiffErr);

speed = kp \* (desiredRpm - rpm) + kd \* ((desiredRpm - rpm) - prevSpeedErr);

prevDiffErr = position;

prevSpeedErr = desiredRpm - rpm;

}

void propIntegControl() {

diff = kp \* position + ki \* integDiffErr;

speed = kp \* (desiredRpm - rpm) + ki \* integSpeedErr;

integDiffErr = mRound(integDiffErr + position, 300, -300);

integSpeedErr = mRound(integSpeedErr + (desiredRpm - rpm), 300, -300);

}

void pidControl() {

diff = kp \* position + kd \* (position - prevDiffErr) + ki \* integDiffErr;

speed = kp \* (desiredRpm - rpm) + kd \* ((desiredRpm - rpm) - prevSpeedErr) + ki \* integSpeedErr;

prevDiffErr = position;

prevSpeedErr = desiredRpm - rpm;

integDiffErr = mRound(integDiffErr + position, 300, -300);

integSpeedErr = mRound(integSpeedErr + (desiredRpm - rpm), 300, -300);

}

void control() {//вибираємо тип регулювання

switch (controlMode) {

case 1:

proportionalControl();

break;

case 2:

propDifControl();

break;

case 3:

propIntegControl();

break;

case 4:

pidControl();

default:

break;

}

}

int getPos() {

if(digitalRead(5))return -1; // l

if(digitalRead(13))return -2; // l

if(digitalRead(3))return 1; // r

if(digitalRead(12))return 2; // r

if(digitalRead(4))return 0; // m

return position;

}

void countPulse() {

pulseCount++;

}

void setup() {

Serial.begin(9600);

readFromEEPROM();//загрузка коефіцієнтів з памяті

pinMode(3, INPUT);

pinMode(4, INPUT);

pinMode(5, INPUT);

pinMode(12, INPUT);

pinMode(13, INPUT);

pinMode(RIGHT\_MOTOR, OUTPUT);

pinMode(LEFT\_MOTOR, OUTPUT);

runMotors();

pinMode(encoderPin, INPUT\_PULLUP);

attachInterrupt(digitalPinToInterrupt(encoderPin), countPulse, RISING);

}

void loop() {

currentMillis = millis();

if (currentMillis - previousMillis >= per) {

rpm =(float)pulseCount / (float)per \* 60000.0 / 360.0;

Serial.print("RPM: ");

Serial.println(rpm);// швидкість коліс

position = getPos();

Serial.print("pos: ");

Serial.println(position);// відхілення від лінії. діапазон [-2;2]

pulseCount = 0;

control();

runMotors();

previousMillis = currentMillis;

}

}