

Лабораторная работа №6.

Управление поворотами робота-паука. Боковые движения.

Цель:

Изучить функции `left()`, `right()`, а также функции `right_line()` и `left_line()` для движения боком

Теоретическая часть

Принципы поворота

- Разница в амплитуде движения левых/правых лап
- Координация осевых и подъемных сервоприводов
- Балансировка центра тяжести при развороте

Принцип бокового движения:

- Синхронное смещение всех лап в одну сторону
- Координация осевых и подъемных сервоприводов
- Поддержание стабильного центра тяжести

Разбор функций поворота

Функция `left()`:

```
void left() {  
    // Инициализация переменных  
    int B = 90, D = 90, F = 90, H = 90;
```

```

// Фаза 1: Подготовка левых лап
for(int i = 90; i <= 150; i += 4) {
    servoA.write(i);          // Левая передняя вперед
    servoB.write(B -= 2);     // Плавный подъем
    servoE.write(i);          // Левая задняя вперед
    servoF.write(F -= 2);
    delay(5);
}

// Фаза 2: Коррекция правых лап
for(int i = 90; i <= 150; i += 4) {
    servoC.write(180-i);      // Правая средняя назад
    servoD.write(D += 2);     // Плавное опускание
    servoG.write(180-i);      // Правая задняя назад
    servoH.write(H += 2);
    delay(5);
}

// Возврат в нейтральное положение
center_servos();
}

```

Принцип работы:

- ⚙️ Левые лапы делают шаг вперед
- ⚙️ Правые лапы отводятся назад
- ⚙️ Синхронизированное движение создает

Разбор функций бокового движения

Функция `right_line()`:

```

void right_line() {
    // Инициализация переменных положения
    int B = 90, D = 90, F = 90, H = 90;
}

```

```
// Фаза 1: Исходное смещение
for(int i = 90; i <= 110; i += 2) {
    // Правые лапы
    servoA.write(i);           // Передняя
    servoB.write(B += 2);      // Подъем с шагом 2°

    servoC.write(i);           // Средняя
    servoD.write(D += 2);

    // Левые лапы (зеркально)
    servoE.write(180-i);       // Задняя
    servoF.write(F += 2);

    servoG.write(180-i);       // Средняя
    servoH.write(H += 2);

    delay(del*3); // Увеличенная задержка для стабильности
}

// Фаза 2: Перенос веса
for(int i = 110; i <= 130; i += 2) {
    servoA.write(i);
    servoB.write(B -= 2); // Плавное опускание

    // ... аналогично для других сервоприводов
    delay(del*3);
}

// Дополнительные фазы возврата...
}
```

Особенности:

Синхронное движение правых лап вперед

Зеркальное движение левых лап назад

Увеличенная задержка для устойчивости

Практическая часть

Задание 1. Напишите функцию `right()`

Ключевые отличия от `left()`:

- ✦ Инвертированные углы для передних/задних лап
- ✦ Обратное направление изменения вспомогательных сервоприводов

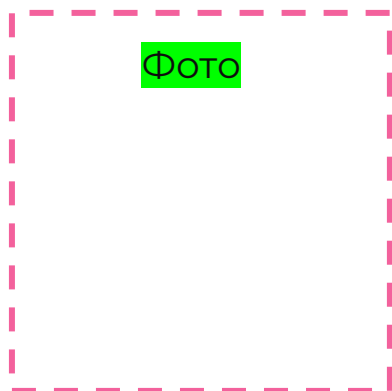
Задание 2. Напишите функцию `left_line()`

Ключевые отличия от `right_line()`:

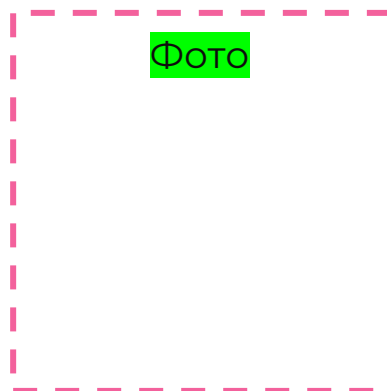
- ✦ Инвертированные углы для всех сервоприводов
- ✦ Обратное направление изменения подъемных сервов

Задание 3: "Квадратная траектория"

Реализуйте движение
по квадрату.
Поворачивайте на углах



Двигайтесь по квадрату
без поворотов



Задание 4: Пройдите трассу с обходом препятствий

Создайте полосу препятствий из подручных средств. Пройдите ее разными способами движения.

Задание 5: "Разминка"

Сделай по два приседа-выпада в каждую сторону в таком порядке:

Вправо (два раза)

Влево (два раза)

Вперёд (два раза)

Назад (два раза)

Контрольные вопросы



1. Почему в `left()` используется `servoC.write(180-i)`?
2. Как влияет изменение `angle_step` на плавность поворота?
3. Каким образом можно реализовать поворот на произвольный угол?
4. Почему важно синхронизировать движение передних и задних лап?
5. Как использовать `forwardMirror()` для улучшения поворотов?