

Лабораторная работа №5.

Движение четырёхлапого робота-паука

Цель:

Освоить синхронизацию сервоприводов для реализации походки "крест-накрест". Реализовать функции движения вперед/назад и поворотов с сохранением устойчивости робота.

Теоретическая часть

Конструкция четырёхлапого паука

Каждая лапа управляет двумя сервоприводами:

Подъём работает по вертикали (например, от 90° до 60°)

Движение — по горизонтали (например, от 90° до 120° или 60°).

```
servoA.attach(2); //передняя левая (1) вправо-влево  
servoB.attach(3); // передняя левая (1) вверх-вниз  
servoC.attach(4); // задняя левая (2) вправо-влево  
servoD.attach(5); // задняя левая (2) вверх-вниз  
servoE.attach(6); // задняя правая (3) вправо-влево  
servoF.attach(7); // задняя правая (3) вверх-вниз  
servoG.attach(8); // передняя правая (4) вправо-влево  
servoH.attach(9); // передняя правая (4) вверх-вниз
```

Алгоритм ходьбы

Походка крест-накрест:

Робот двигает лапами диагональными парами (1-3 и 2-4), чтобы сохранять устойчивость:

Фаза 1: Шаг

- 1 Лапы **1** и **3** поднимаются → переносятся вперёд → опускаются, т.е описывают верхний полукруг.
- 2 Лапы **2** и **4** в это время отталкиваются назад: опускаются → переносятся назад → поднимаются, т.е. описывают нижний полукруг.

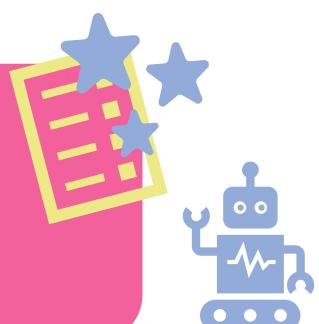
Фаза 2: Возвращение на свои места

- 1 Лапы **1** и **3** опускаются → переносятся назад → поднимаются, т.е. описывают нижний полукруг.
- 2 Лапы **2** и **4** поднимаются → переносятся вперёд → опускаются, т.е описывают верхний полукруг.

Каждая фаза включает:

Таким образом робот остаётся устойчивым.

- 1.Подъём лапы
- 2.Перенос лапы вперёд
- 3.Опускание лапы
- 4.Отталкивание назад

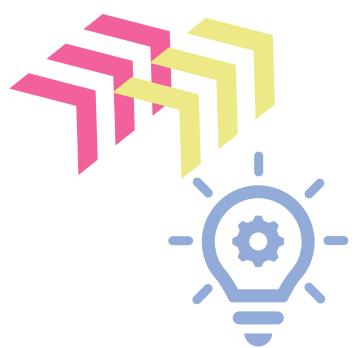


Движение вперед/назад:

Функция `forward()`

Реализация основного алгоритма движения вперед через:

- ⚙️ Поочередное перемещение пар лап
- ⚙️ Плавное перераспределение веса
- ⚙️ Циклическое повторение фаз движения



```
void forward() { // Шаг вперед
    // Начальные углы вспомогательных серво лапок
    int B = 90;
    int D = 90;
    int F = 90;
    int H = 90;

    // Этап 1:
    // Лапы 1 и 3 поднимаются и двигаются вперед (угол i с 90 до
    110)
    // Лапы 2 и 4 опускаются и двигаются назад, вспомогательные
    сервы постепенно поднимаются
    for (int i = 90; i <= 110; i += 2) {
        servoA.write(i);           // Лапа 1 - подъем и движение вперед
        B += 1;
        servoB.write(B);          // Вспомогательное движение лапы 1
        delay(5);
        servoC.write(180 - i);    // Лапа 2 - опускание и движение назад
        (зеркально)
        D += 1;
        servoD.write(D);          // Вспомогательное движение лапы 2
        delay(5);
        servoE.write(180 - i);    // Лапа 3 - подъем и движение вперед
        (зеркально)
        F += 1;
        servoF.write(F);          // Вспомогательное движение лапы 3
        delay(5);
        servoG.write(i);          // Лапа 4 - опускание и движение назад
        H += 1;
        servoH.write(H);          // Вспомогательное движение лапы 4
        delay(5);
    }

    // Этап 2:
    // Лапы 1 и 3 продолжают движение вперед и опускание (i от 110
    до 130),
    // вспомогательные серво постепенно опускаются
```

```
for (int i = 110; i <= 130; i += 2) {  
    servoA.write(i);  
    B -= 1;  
    servoB.write(B);  
    delay(5);  
    servoC.write(180 - i);  
    D -= 1;  
    servoD.write(D);  
    delay(5);  
    servoE.write(180 - i);  
    F -= 1;  
    servoF.write(F);  
    delay(5);  
    servoG.write(i);  
    H -= 1;  
    servoH.write(H);  
    delay(5);  
}  
}
```

```
// Этап 3:  
// Лапы 1 и 3 возвращаются назад, i от 130 до 110,  
// вспомогательные серво продолжают опускать лапы  
for (int i = 130; i >= 110; i -= 2) {  
    servoA.write(i);  
    B -= 1;  
    servoB.write(B);  
    delay(5);  
    servoC.write(180 - i);  
    D -= 1;  
    servoD.write(D);  
    delay(5);  
    servoE.write(180 - i);  
    F -= 1;  
    servoF.write(F);  
    delay(5);  
    servoG.write(i);  
    H -= 1;  
    servoH.write(H);  
    delay(5);  
}  
}
```

```
// Этап 4:  
// Лапы 1 и 3 опускаются на исходные позиции, i от 110 до 90,  
// вспомогательные серво поднимают лапы обратно к начальным  
углам  
for (int i = 110; i >= 90; i -= 2) {  
    servoA.write(i);  
    B += 1;  
    servoB.write(B);  
    delay(5);  
    servoC.write(180 - i);  
    D += 1;  
    servoD.write(D);  
    delay(5);  
    servoE.write(180 - i);  
    F += 1;  
    servoF.write(F);  
    delay(5);  
    servoG.write(i);  
    H += 1;  
    servoH.write(H);  
    delay(5);  
}  
}
```

Как это работает?

1 Фазированное движение:

Лапы движутся в противофазе для баланса

Передние и задние лапы работают зеркально

2 Управление сервоприводами:

Основные сервы (A,C,E,G) - задают направление

Вспомогательные (B,D,F,H) - контролируют высоту

3

Параметры:

Шаг изменения угла: 2°

Базовая задержка: 5мс

Проблема отклонения при движении

Во время ходьбы робот-паук может отклоняться в сторону из-за:

- ★ Механических погрешностей сборки
- ★ Неидентичных характеристик сервоприводов
- ★ Неравномерного распределения веса

Проявление проблемы:

Постепенное смещение вправо из-за:

- ★ Накопления механических погрешностей
- ★ Преимущественной нагрузки на правые сервоприводы

Решение через зеркальные движения

Принцип коррекции:

Чередование стандартной и зеркальной походки

Компенсация систематических ошибок

Статистическое выравнивание траектории

Зеркальный шаг (`forwardMirror()`)

Теперь в фазе 1 вперед шагают лапы 2 и 4, а отталкиваются лапы 1 и 3. И, соответственно, в фазе 2 пары лап двигаются наоборот.

Компенсирующие факторы:



- Инверсия направлений движения
- Обратное распределение нагрузок
- Симметричное накопление ошибок

Функции `back()` и `backMirror()`

Обратные варианты движения:

- `back()` - инверсия `forward()`
- `backMirror()` - инверсия `forwardMirror()`

Основные изменения

- ★ Обратный порядок фаз
- ★ Инвертированные направления движения сервоприводов
- ★ Измененные пределы углов

! Поднимающаяся пара лап в фазе 1 шагает назад, соответственно, отталкивающаяся – двигается вперед

Практическая часть

Задание 1. Напишите функцию `forwardMirror()`

Задание 2. Напишите функцию `back()`

Задание 3. Напишите функцию `backMirror()`

Задание 4. Крест-накрест

Реализуйте походку крест-накрест вперед и назад с компенсацией смещений.

Контрольные вопросы



- | 1. В чем преимущество походки "крест-накрест"?
- | 2. Почему для одной лапы нужно два сервопривода?
- | 3. Какую функцию выполняют вспомогательные сервоприводы (B, D, F, H)?
- | 4. Почему робот отклоняется в сторону при движении?
- | 5. Как работает компенсация смещения с помощью forwardMirror()?