

Практическое занятие №6

Тема: «Подключение и программирование сервоприводов»

Цель работы: приобрести практические навыки по подключению и программированию сервоприводов на платформе Arduino.

Последовательность выполнения работы:

- Изучить теоретические сведения, приведенные в практическом занятии.
- Сделать монтажную и принципиальную схему в программе Fritzing. (напоминание: принципиальная схема формируется автоматически после создания монтажной).
- Собрать схемы желательно на макетной плате, иначе при отсутствии набора Arduino в web-приложениях (<https://wokwi.com/projects/new/arduino-uno> или при отсутствии необходимых компонентов на <https://www.tinkercad.com/>) для приведенных примеров.
- Запрограммировать микроконтроллер согласно тексту, указанному в примере.
- Выполнить задание для самостоятельной работы.

Содержание отчета:

- 1) Название практического занятия, его цель.
- 2) Ход работы: выполнение задания, фото или скриншоты собранной схемы, написанный программный код, шрифтом Courier New, 12 кегль, одинарный отступ без абзацев.
- 3) Контрольные вопросы.
- 4) Задание самостоятельной работы при наличии.
- 5) Вывод о проделанной работе.
- 6) Файл Fritzing с принципиальной и монтажной схемой.

Оценивание

Оценка «3»: правильно оформленный отчет, правильно выполненные задания, подтверждение этому и ответы на контрольные вопросы (1-3, 5 пункты).

Оценка «4»: правильно оформленный отчет, правильно выполненные задания, подтверждение этому, ответы на контрольные вопросы и выполненное задание в самостоятельной работе (при наличии) (1-5 пункты).

Оценка «5»: правильно оформленный отчет, правильно выполненные задания, подтверждение этому, ответы на контрольные вопросы, выполненное задание в самостоятельной работе (при наличии) и файлы с монтажной и принципиальной схемой Fritzing задания практического занятия и самостоятельной работы (при наличии) (1-6 пункты).

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Сервопривод (сервомотор) — это электромеханическое устройство, предназначенное для точного управления положением, скоростью и ускорением вала. Он широко используется в робототехнике, станках с ЧПУ, авиамоделировании, промышленной автоматизации и других областях, где требуется высокая точность движения.

Устройство сервопривода:

Сервопривод состоит из нескольких ключевых компонентов:

Электродвигатель:

- Постоянного тока (DC) – простые и недорогие, но менее точные.
- Бесколлекторные (BLDC) – более эффективные, долговечные, с высоким КПД.
- Шаговые двигатели – обеспечивают точное позиционирование без обратной связи.

Управляющая электроника сервопривода:

- Контроллер (драйвер) – сравнивает заданное положение с текущим и корректирует движение.
- ШИМ-модулятор – регулирует мощность, подаваемую на двигатель.

Редуктор (в некоторых моделях) понижает скорость вращения и увеличивает крутящий момент, или может быть выполнен из металла (для высоких нагрузок) или пластика (для легких применений).

Принцип работы

1. На вход сервопривода подается управляющий сигнал (обычно ШИМ или цифровой протокол).
2. Контроллер сравнивает текущее положение (с датчика) с заданным.
3. Если есть рассогласование, двигатель вращается в нужном направлении до достижения требуемого угла.
4. Обратная связь обеспечивает точное позиционирование.

Типы сервоприводов

По типу управления:

- Аналоговые – управляются ШИМ-сигналом (например, 1–2 мс импульс при 50 Гц).
- Цифровые – имеют встроенный процессор, быстрее реагируют, точнее.
- Программируемые – настраиваются через ПО (например, сервоприводы Dynamixel).

По назначению:

- Позиционные – работают в режиме углового позиционирования (обычно 0–180°).
- Непрерывного вращения – могут вращаться на 360° (как мотор с редуктором).
- Линейные – преобразуют вращение в линейное перемещение.

По мощности:

- Стандартные (5–6 В, 1–5 кг·см) – для роботов, моделей.
- Силовые (12–48 В, 10–50 кг·см) – промышленные применения.
- Микро- и наносервоприводы – для миниатюрных устройств.

Основные характеристики

Параметр	Описание
Напряжение питания	Обычно 4.8–7.4 В (для модельных), 12–48 В (промышленные)
Крутящий момент	Измеряется в кг·см или Н·м (например, 5 кг·см при 6 В)
Скорость вращения	Время поворота на 60° (например, 0.1–0.2 с)
Угол поворота	90°, 180°, 270° или непрерывное вращение
Тип редуктора	Пластиковый, металлический (латунь, титан)
Интерфейс управления	PWM (ШИМ), RS-485, CAN, EtherCAT

Области применения

- Робототехника – управление конечностями, захватами.
- Авиамоделирование – рулевые машинки в самолетах и вертолетах.
- Промышленность – станки ЧПУ, конвейеры, автоматические линии.
- Медицина – хирургические роботы, протезирование.
- Умный дом – автоматические шторы, системы вентиляции.

Как определить подходящий сервопривод для задач?

1. Определите напряжение питания (5В, 12В, 24В).
2. Выберите крутящий момент (чем больше нагрузка, тем выше момент).
3. Учитывайте скорость (быстрые сервоприводы дороже).
4. Решите, нужен ли аналоговый или цифровой вариант.
5. Проверьте тип редуктора (пластик дешевле, металл долговечнее).

ЗАДАНИЕ

Собрать схему:

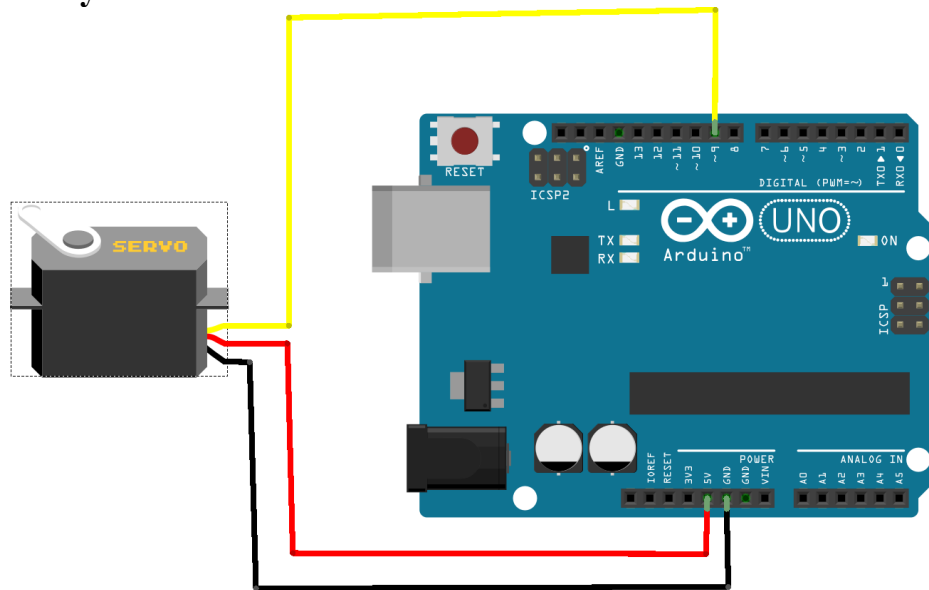


Рисунок 1 – Монтажная схема к заданию

Код программы 1

```
int servopin = 9 ;// define the digital interface to connect the servo servo signal line 9
int myangle ;// define the angle variables
int pulselwidth ;// define variable pulse width
int val;
void servopulse (int servopin, int myangle) // define a pulse function
{
    pulselwidth = (myangle * 11) +500 ;// the angle value into a pulse width 500-2480
    digitalWrite (servopin, HIGH) ;// the servo interface level to high
    delayMicroseconds (pulselwidth) ;// number of microseconds delay pulse width value
    digitalWrite (servopin, LOW) ;// the servo interface level to Low
    delay (20-pulselwidth/1000);
}
void setup ()
{
    pinMode (servopin, OUTPUT) ;// set servo interface output interface
    Serial.begin (9600) ;// connect to the serial port, the baud rate is 9600
    Serial.println ("servo = o_seral_simple ready");
}
void loop () // will be 0-9's 0-180 number into perspective, and let the number of times
the corresponding LED flashes
{
    val = Serial.read () ;// read the value of the serial port
    if (val> '0'&& val <= '9')
    {
        val = val-'0'; // will be converted to numeric variables characteristic quantities
        val = val * (180/9) ;// will figure into perspective
        Serial.print ("moving servo to");
        Serial.print (val, DEC);
        Serial.println ();
        for (int i = 0; i <= 50; i++) // give enough time to let it go to the steering
angle specified
        {
            servopulse (servopin, val) ;// reference pulse function
        }
    }
}
```

Код программы 2

#include <Servo.h> // define header files, there is one thing to note, you can directly click on the menu bar at the Arduino software Sketch > Import library > Servo, Servo function call, you can also directly enter the #include <Servo.h>, But at the input to the attention of the #include and <Servo.h> should be a space between, otherwise a compile-time error.

Servo myservo ;// define a variable name servos

void setup ()

{

myservo.attach (9) ;// define Servo Interface (9,10 all OK, shortcomings can only control 2)

}

void loop ()

{

myservo.write (90) ;// set the steering angle of rotation

}

САМОСТОЯТЕЛЬНАЯ РАБОТА

Ответьте на контрольные вопросы:

1. Какие основные компоненты входят в состав сервопривода?
2. Чем отличается аналоговый сервопривод от цифрового?
3. Какой тип двигателя чаще всего используется в сервоприводах и почему?
4. Какие типы сервоприводов существуют по характеру движения?
5. В чем разница между позиционным сервоприводом и сервоприводом непрерывного вращения?
6. Какие параметры сервопривода указывают на его мощность и скорость работы?
7. Где чаще всего применяются сервоприводы? Приведите примеры.
8. Как правильно выбрать сервопривод для конкретной задачи? На какие параметры обратить внимание?
9. Какой вид программы управления из задания более предпочтителен: 1 или 2 программный код?