

ЛАБОРАТОРНА РОБОТА № 6 SERVO MOTOR

Мета роботи: ознайомитись з особливістю будови та різновидами серводвигунів, та отримати навички роботи Arduino з серводвигунами.

1. Теоретичні відомості

Сервопривід – привід з управлінням через негативний зворотній зв'язок, що дозволяє точно керувати параметрами руху. Сервопривід з мотором (серводвигун), призначений для приведення в рух пристроїв керування через поворот вихідного валу.

Сімейство серводвигунів різноманітне, його можна розподілити на декілька груп (рис.6.1). Відмінні особливості обумовлені наступними факторами:

- конструкція двигунів (статор, ротор);
- необхідні системи регулювання;
- система зворотного зв'язку (датчики).

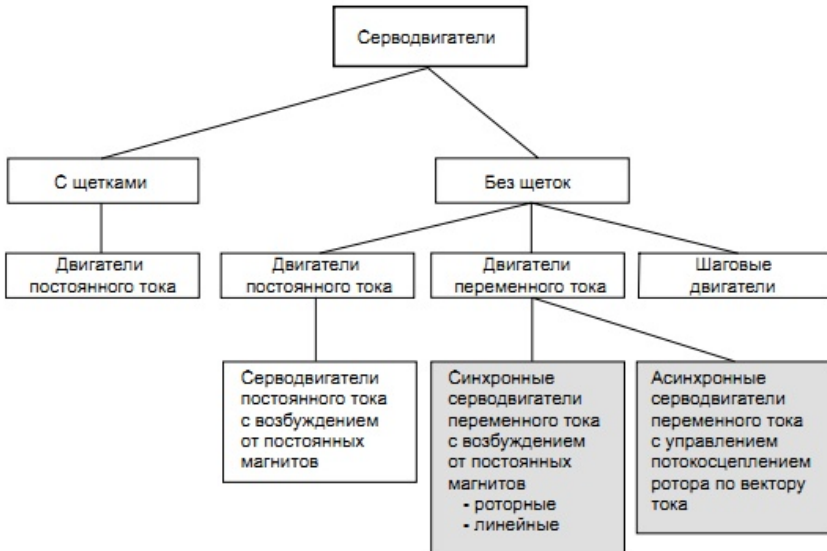


Рисунок 6.1 – Огляд серводвигунів

Будова сервоприводу для радіокерованих моделей зображена на рисунку 6.2. Він складається з електродвигуна, укладеного в один корпус з редуктором і керуючою електронікою, яка найчастіше складається з потенціометра зворотного зв'язку та плати управління. Серводвигун використовує три приводи для роботи. Плюсовий привід для живлення, зазвичай 4.8В-6В, мінусовий привід та сигнальний привід. Керуючий сигнал передає інформацію щодо положення вихідного валу. Вал пов'язаний з потенціометром, який визначає його положення. Контролер згідно опору потенціометра і значенням керуючого сигналу визначає, в який бік потрібно обератит мотор, щоб отримати потрібна положення вихідного валу.



Рисунок 6.2 – Конструкція сервоприводу

Виводи сервоприводу наступні:

Три проводи: чорний(коричневий), червоний, білий(жовтий) – відповідно, земля, живлення та керуючий сигнал (ШІМ) .

Керування сервопривідом відбувається шляхом подачі на нього прямокутних імпульсів по сигнальному виводу з частотою біля 50Гц, амплітудою не менше 3.3В (часто не менше 4.8В), шириною стандартно від 1000 до 2000мкс, що відповідає стандартним крайнім положенням (див. рис. 6.3). Зазвичай реальний діапазон може бути трохи ширше (наприклад, 900-2100мкс), але це вже залежить конкретно від виробника сервоприводів.

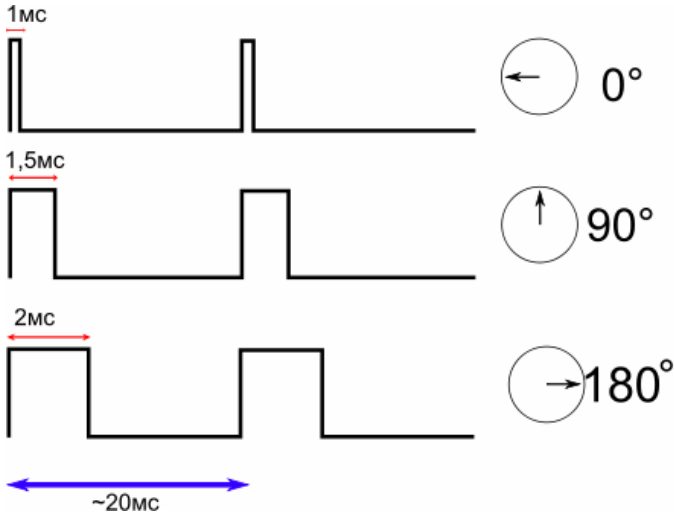


Рисунок 6.3 – Керуючі сигнали

2. Виконання роботи

2.1 Створення сервоприводу у Proteus

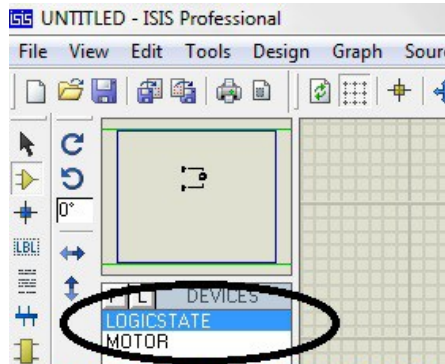
Ми перевіримо, як спроектувати сервопривід у Proteus ISIS, а також спроектуємо керуючий привід постійного струму за допомогою логічних елементів. З початку ми створимо двигун постійного струму, який присутній в Proteus і дуже простий у використанні. Ми будемо керувати їм шляхом подачі напруги на його обидві сторони, тобто прямим методом.

Двигун постійного струму являє собою простий двигун, який потребує різної полярності на своїх двох кінцях. Якщо ця полярність в прямому напрямку, то двигун постійного струму рухається в одному напрямку, і якщо ми змінимо полярність то двигун буде рухатися в протилежному напрямку. Отже, створимо DC Motor Drive Circuit в Proteus ISIS.

Простий DC Motor Drive Circuit в Proteus ISIS

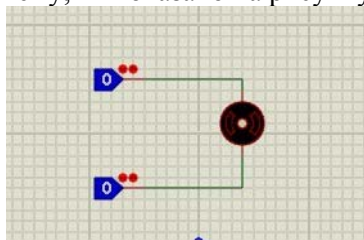
Виберіть компоненти з бібліотеки Proteus:

1. Motor
2. Logic State



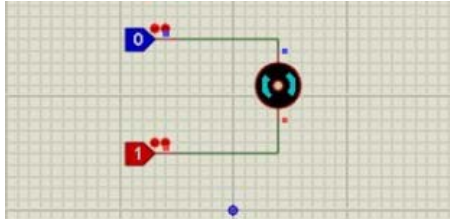
Logic State має два стани 1 і 0. Коли 0 це означає 0В і коли 1 - 5V.

Тепер розробіть схему, як показано на рисунку нижче:



Як бачимо на схемі додано два логічні стан на обох сторонах двигуна. Напрямок двигуна буде залежати від цієї логіки. Таким чином, їх буде всього чотири стани:

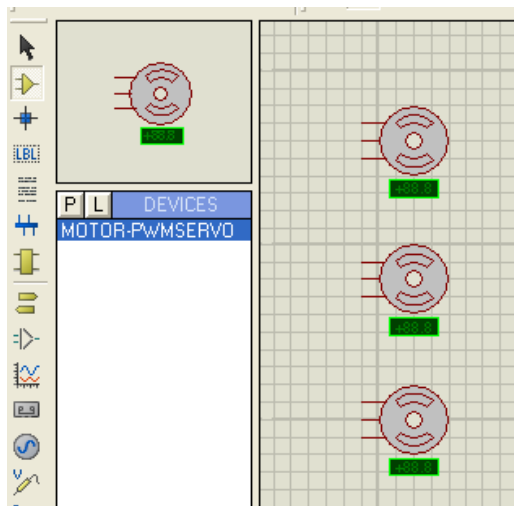
1. Коли обидва стани 0, двигун не буде рухатися і залишатися нерухомими.
2. Коли обидва стани знаходяться на 1, також двигун не буде рухатися і залишатися нерухомими.
3. Двигун буде рухатися за годинниковою стрілкою, коли верхній стан 1, а нижній 0.
4. Двигун буде рухатися проти годинникової стрілки, коли верхній стан 0, а нижній 1.



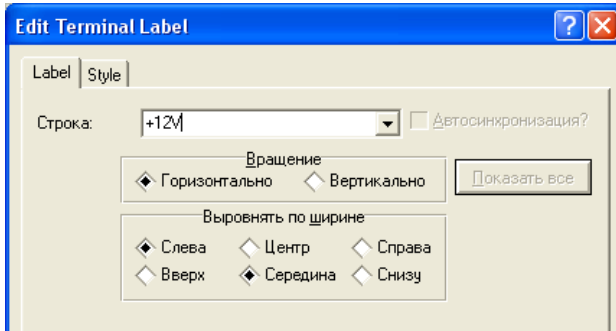
Керування сервоприводом у Proteus ISIS за допомогою ШІМ

Для керування сервоприводом нам потрібно формувати ШІМ з частотою 50 Герц. При цьому для положення “0” довжина імпульсу повинна складати 1000 мікросекунд, а для положення “максимум” – 2000 мікросекунд. Середнє положення – 1500 мікросекунд (рис. 6.3). Розглянемо це конкретно на прикладі.

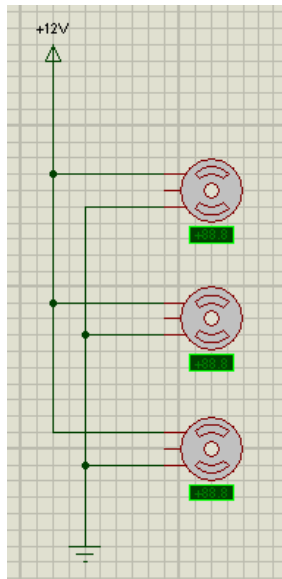
Відкриваємо Proteus ISIS та створюємо новий проект. У бібліотеці елементів знаходимо сервопривід та розміщуємо так як зображено на рисунку нижче. На ці три сервоприводи ми будемо відповідно надсилати сигнали різної довжини: 1мс, 1.5мс та 2 мс.



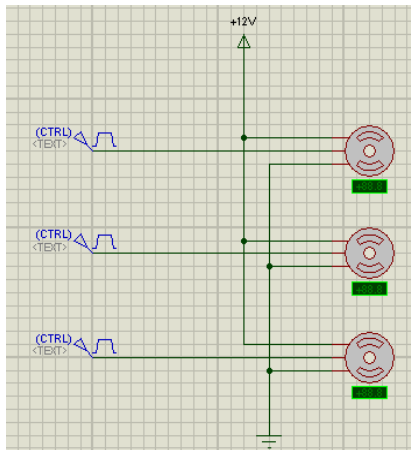
Встановлюємо елемент живлення «POWER» та у його властивостях задаємо живлення «+12V»



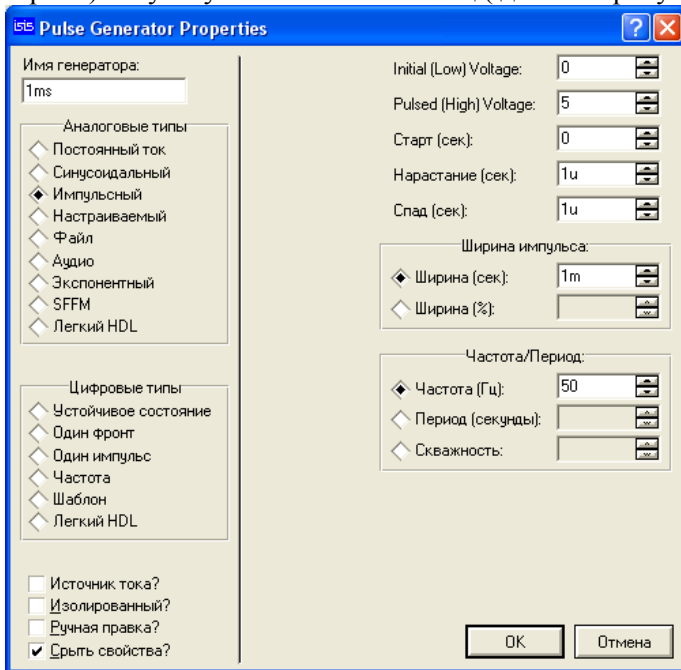
Після встановлення заземлення електрична схема буде виглядати так як зображено на рисунку нижче.



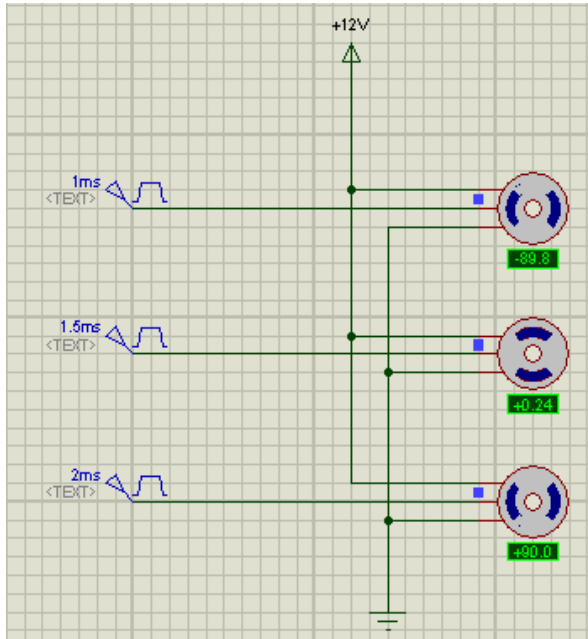
Для генерування сигналів потрібної частоти та довжини встановимо елементи, які будуть генерувати три різні сигнали на кожен з сервоприводів. Перейдіть у лівій панелі інструментів режим Generator Mode та виберіть компонент PULSE. Встановіть три компоненти PULSE до схеми та з'єднайте їх як зображено на рисунку нижче.



Для першого зверху генератора імпульсу встановіть параметр довжини (ширина) імпульсу 1мс та частота 50Гц (дивіться рисунок).



Для двох наступних за аналогією з попереднім встановіть довжину імпульсу 1.5мс та 2мс відповідно. Як схема буде завершена, запустіть симуляцію. У результаті симуляції верхній сервопривід повернеться у крайнє ліве положення, середній встановить нульове положення, а нижній повернеться у крайнє праве положення, як зображено на рисунку нижче.



2.2 Створення проекту з сервоприводом на Arduino

Для розглядання прикладу керування положенням сервоприводу за допомогою Arduino необхідна плата та сервопривід. Схема підключення зображена на рисунку 6.4. Як вже було сказано вище серводвигун має 3 приводи: живлення, заземлення, сигнальний. Дріт живлення, як правило, червоний, і повинен бути з'єднаний з піном 5V на платі Arduino. Дріт заземлення, як правило, чорного або коричневого кольору і повинен бути підключений до заземлення на платі Arduino. Сигнальний контакт, як правило, жовтого, білого або помаранчевого кольору, і повинен бути підключений до контакту 9 на платі Arduino.

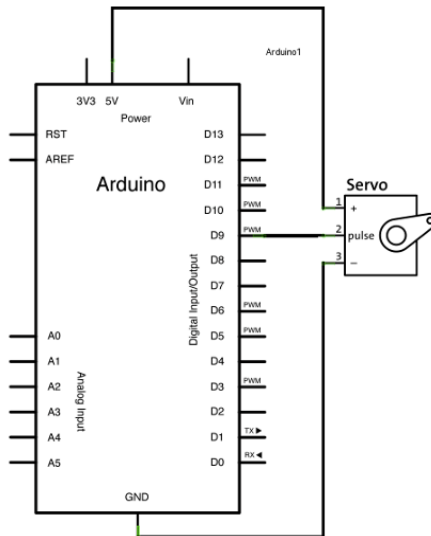


Рисунок 6.4 – Схема підключення сервоприводу до Arduino

Для написання програми у середовищі Arduino є спеціальна бібліотека для роботи з сервоприводом «Servo.h». Далі наведено простий приклад програми роботи з сервоприводом:

```
#include <Servo.h>
```

```
Servo myservo; // створення об'єкта сервоприводу для курування серво; максимум можна створити 8 об'єктів сервоприводів
```

```
int pos = 0; // змінна для збереження положення сервоприводу
```

```
void setup()
```

```
{
```

```
  myservo.attach(9); // призначити серво який підключений до 9 піна до об'єкту сервоприводу
```

```
}
```

```

void loop()
{
  for(pos = 0; pos < 180; pos += 1) // рух від 0 до 180 градусів
  {
    myservo.write(pos);           // дає команду сервоприводу перейти на
    позицію 'pos'
    delay(15);                    // затримка 15мс для того щоб серво досягла
    необхідної позиції
  }
  for(pos = 180; pos >= 1; pos -= 1) // рух від 180 до 0 градусів
  {
    myservo.write(pos);           // дає команду сервоприводу перейти на
    позицію 'pos'
    delay(15);                    // затримка 15мс для того щоб серво досягла
    необхідної позиції
  }
}
}

```

3. Завдання до лабораторної роботи

3.1 Підключити кнопку і сервопривід до плати Arduino. Написати програму, щоб при натисканні кнопки сервопривід рухався за годинниковою стрілкою. При відпусканні кнопки сервопривід зупинявся.

3.2 Підключити кнопку і сервопривід до плати Arduino. Написати програму, щоб при натисканні кнопки сервопривід рухався проти годинникової стрілки. При відпусканні кнопки сервопривід зупинявся.

4. Зміст звіту

- 4.1. Тема та мета роботи
- 4.2. Схема електрична-принципова
- 4.3. Код програми
- 4.4. Висновки з роботи

5. Контрольні запитання

- 5.1 Що таке сервопривід і які різновиди сервоприводів існують?
- 5.2 За якими ознаками відрізняються сервоприводи?
- 5.3 Як відбувається керування сервоприводом керуючими сигналами?
- 5.4 До яких виводів підключається сервопривід до платформи Arduino?
- 5.5 Назвіть основні команди Arduino для керування сервоприводом?