

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»



Кафедра РТТ

Методичні вказівки
до самостійної роботи
з дисципліни
«Вбудовані біомедичні системи та бездротові сенсорні мережі»
для студентів спеціальності
172 «Телекомунікації та радіотехніка»
освітньої програми «Телемедичні та біомедичні системи»
всіх форм навчання

Частина 1.

2019

Методичні вказівки до самостійної роботи з дисципліни «Вбудовані біомедичні системи та бездротові сенсорні мережі» для студентів спеціальності 172 «Телекомунікації та радіотехніка» освітньої програми «Телемедичні та біомедичні системи» всіх форм навчання. Частина 1. / Укл.: А.В. Пархоменко, О.М. Гладкова. – Запоріжжя: НУ «Запорізька політехніка», 2019. – 53 с.

Укладачі: А.В. Пархоменко, к.т.н., доцент каф. ПЗ
О.М. Гладкова, к.т.н., доцент каф. РТТ

Рецензент: Г.М. Шило, д.т.н., зав.каф. ІТЕЗ

Відповідальний
за випуск: зав.каф. радіотехніки та телекомунікацій
С.В. Морщавка, к.т.н., доцент.

Затверджено
на засіданні
кафедри РТТ
Протокол № 1 від
“11” вересня 2019 р.

Ухвалено
на засіданні НМК
факультету РЕТ
Протокол № 2 від
“19” вересня 2019 р.

ЗМІСТ

САМОСТІЙНА РОБОТА № 1 СИМУЛЯЦІЯ РОБОТИ ARDUINO ЗА ДОПОМОГОЮ WEB-ЗАСТОСУНКУ TINKERCAD	4
САМОСТІЙНА РОБОТА № 2 СТВОРЕННЯ ПРОЄКТУ СИСТЕМИ ІНДИКАЦІЇ.....	14
САМОСТІЙНА РОБОТА № 3 ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ РОБОТИ З АКТУАТОРАМИ.....	20
САМОСТІЙНА РОБОТА № 4 ДОСЛІДЖЕННЯ ТА ВИКОРИСТАННЯ СЕРВОДВИГУНІВ	26
САМОСТІЙНА РОБОТА № 5 ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ РОБОТИ З РЕЛЕ.....	37
САМОСТІЙНА РОБОТА № 6 РОБОТА З БІБЛІОТЕКОЮ ARDUINO. ПРОГРАМУВАННЯ LCD ДИСПЛЕЯ	48

САМОСТІЙНА РОБОТА № 1

СИМУЛЯЦІЯ РОБОТИ ARDUINO ЗА ДОПОМОГОЮ WEB-ЗАСТОСУНКУ TINKERCAD

Мета роботи: ознайомитися з основними елементами плати Arduino Mega та web-застосунком віртуальної імітації роботи платформи Arduino, розробити прості проекти мерехтіння світлодіодів з використанням можливостей web-застосунку та реальної платформи.

1. Теоретичні відомості

1.1 Arduino

Arduino – це електронний конструктор і зручна платформа швидкої розробки електронних пристроїв для новачків і професіоналів. Переваги платформи: зручна у використанні, простота мови програмування, а також відкрита архітектура і програмний код. Пристрій програмується через USB без використання програматорів.

Пристрої на базі Arduino можуть отримувати інформацію про навколишнє середовище за допомогою різних датчиків, а також можуть управляти різними виконавчими пристроями.

Існує декілька версій платформ Arduino. В цьому курсі буде розглядатися платформа Arduino Uno (рис. 1.1).

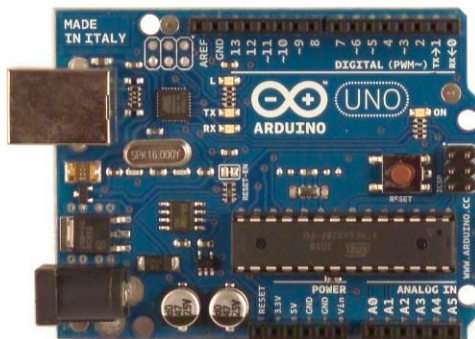


Рисунок 1.1 – Зовнішній вигляд Arduino Uno

Arduino Uno базується на мікроконтролері Atmel ATmega328. Мікроконтролер на платі програмується за допомогою мови Arduino (заснований на мові Wiring) і середовища розробки Arduino (заснована на середовищі Processing). Проекти пристроїв, засновані на Arduino, можуть працювати самостійно, або ж взаємодіяти з програмним забезпеченням на комп'ютері (напр.: Flash, Processing, MaxMSP). Плати можуть бути зібрані користувачем самостійно або куплені в зборі. Програмне забезпечення доступне для безкоштовного скачування. Вихідні креслення схем (файли CAD) є загальнодоступними, користувачі можуть застосовувати їх на свій розсуд.

Платформа має 14 цифрових вхід/виходів (6 з яких можуть використовуватися як виходи ШІМ), 6 аналогових входів, кварцовий генератор 16 МГц, роз'єм USB, силовий роз'єм, роз'єм ICSP і кнопку перезавантаження. Кожен з 14 цифрових виводів Uno може налаштований як вхід або вихід, використовуючи функції `pinMode ()`, `digitalWrite ()`, і `digitalRead ()`. Кожен вивід має навантажувальний резистор (за замовчуванням відключений) 20-50 кОм і може пропускати до 40 мА. Для роботи необхідно підключити платформу до комп'ютера за допомогою кабелю USB, або подати живлення за допомогою адаптера AC/DC або батареї.

Платформа може працювати при зовнішньому живленні від 6 В до 20 В. При напрузі живлення нижче 7 В, вивід 5V може видавати менше 5 В, при цьому платформа може працювати нестабільно. При використанні напруги вище 12 В регулятор напруги може перегрітися і пошкодити плату. Рекомендований діапазон від 7 В до 12 В.

Деякі виводи мають особливі функції:

- VIN. Вхід використовується для подачі живлення від зовнішнього джерела (в відсутність 5 В від роз'єму USB або іншого регульованого джерела живлення). Подача напруги живлення відбувається через даний вивід.

- 5V. Регульоване джерело напруги, що використовується для живлення мікроконтролера і компонентів на платі. Живлення може подаватися від виводу VIN через регулятор напруги, або від роз'єму USB, або іншого регульованого джерела напруги 5 В.

- 3V3. Напруга на виводі 3.3 В генерується вбудованим регулятором на платі. Максимальне споживання струму 50 мА.

- GND. Виводи заземлення.
 - Послідовна шина: 0 (RX) і 1 (TX). Виводи використовуються для отримання (RX) і передачі (TX) даних TTL. Дані виводи підключені до відповідних роз'ємів мікросхеми послідовної шини ATmega8U2 USB-to-TTL.
 - Зовнішнє переривання: 2 і 3 Дані висновки можуть бути налаштовані на виклик переривання або на молодшому значенні, або на передньому чи задньому фронті, або при зміні значення. Детальна інформація знаходиться в описі функції `attachInterrupt ()`.
 - ШІМ: 3, 5, 6, 9, 10, і 11. Будь-який з виводів забезпечує ШІМ з роздільною здатністю 8 біт за допомогою функції `analogWrite ()`.
 - SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). За допомогою даних виводів здійснюється зв'язок SPI, для чого використовується бібліотека SPI.
 - LED: 13. Вбудований світлодіод, підключений до цифрового виводу 13. Якщо значення на виводі має високий потенціал, то світлодіод горить.
- На платформі Uno встановлені 6 аналогових входів (позначених як A0 .. A5), кожен роздільною здатністю 10 біт (тобто може приймати 1024 різних значення). Стандартно виводи мають діапазон вимірювання до 5 В відносно землі, проте є можливість змінити верхню межу за допомогою виводу AREF і функції `analogReference ()`. Деякі виводи мають додаткові функції:
- I2C: 4 (SDA) і 5 (SCL). За допомогою виводів здійснюється зв'язок I2C (TWI), для створення якої використовується бібліотека Wire.
- Додаткова пара виводів платформи:
- AREF. Опорна напруга для аналогових входів. Використовується з функцією `analogReference ()`.
 - Reset. Низький рівень сигналу на виводі перезавантажує мікроконтролер. Звичайно застосовується для підключення кнопки перезавантаження на платі розширення, що закриває доступ до кнопки на самій платі Arduino.

1.2 Основні функції для роботи зі світлодіодами

Для роботи зі світлодіодом необхідно знати і вміти володіти такими функціями і константами:

- оператор `setup()`;
- оператор `loop()`;
- функція `pinMode()`
- функція `digitalWrite()`;
- функція `delay()`;
- константи `OUTPUT`, `HIGH`, `LOW`.

Далі наведений код програми найпростішого прикладу мерехтіння вбудованим у плату Arduino світлодіодом, який підключено до 13 виводу:

```
void setup()
{ pinMode(13, OUTPUT);
}
```

Ця функція виконується на початку роботи програми (після запуску мікроконтролеру). Тобто послідовно виконується кожна команда, яка знаходиться між фігурними скобками цієї функції. Наприкінці кожної строки необхідно поставити символ закінчення команди “;”. Тут функція `setup` містить одну єдину команду – `pinMode(13, OUTPUT)`. Ця команда налаштовує 13 порт Arduino, як вивід. Порт № 13 знаходиться на верхній колодці портів Arduino.

Після функції `setup` виконується функція `loop`.

```
void loop()
{
digitalWrite(13, HIGH); // вмикаємо світлодіод
delay(1000); // чекаємо секунду
digitalWrite(13, LOW); // вимикаємо світлодіод
delay(1000); // чекаємо секунду
}
```

На відміну від `setup`, функція `loop` постійно повторюється – як тільки послідовно виконані всі команди в скобках, функція

запускається знову. Функція loop для цього прикладу складається з чотирьох команд:

На порт 13 подається напруга (5 В) – світлодіод вмикається.

Затримка до виконання наступної команди 1000 мілісекунд (одну секунду)

Порт 13 з'єднується з землею – світлодіод вимикається.

Ще одна затримка на 1 секунду.

Після виконання усіх чотирьох команд, знову виконується перша команда (включення світлодіоду) и так продовжується до тих пір, поки Arduino включена або поки не буде натиснута кнопка RESET.

1.3 Симуляція роботи світлодіоду та Arduino в Tinkercad

Сервіс Tinkercad компанії Autodesk дає можливість створити електронні схеми. Вебзастосунок має підтримку платформи Arduino та дозволяє редагувати код, а також дає можливість у візуальному режимі будувати схем і виконувати інтерактивну імітацію їх роботи у реальному часі. Працювати над схемами можна спільно з іншими людьми, використовуючи бібліотеку компонентів.

Для роботи з сервісом Tinkercad перейдіть за посиланням <https://www.tinkercad.com/>.

Після проходження реєстрації в Tinkercad ви опинитесь на головній сторінці (рис. 1.2). Домашня сторінка показує список схем, які ви створили, а також нещодавно створені/редаговані.

Щоб створити нову схему клікніть на кнопку «+New», яка знаходиться у верхньому меню. Після, натисніть кнопку "New Electronics Lab" на правій панелі вибору. Далі вам буде представлено сторінка для побудови свого проекту з основними функціями та інструментами для роботи зі схемою та програмою. Веб-інтерфейс програмного забезпечення, представлений на рисунку 1.3, складається з таких основних блоків:

1. робоча область створення схеми
2. панель керування проектом
3. меню вибору відображення схеми (Breadboard View, Schematics View, PCB View)

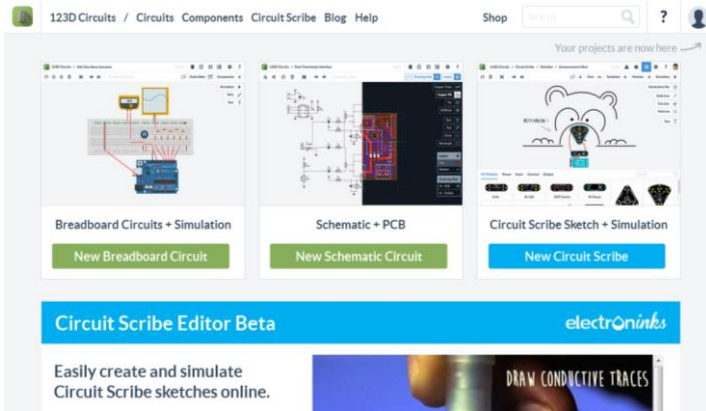


Рисунок 1.2 – Головна сторінка Tinkercad

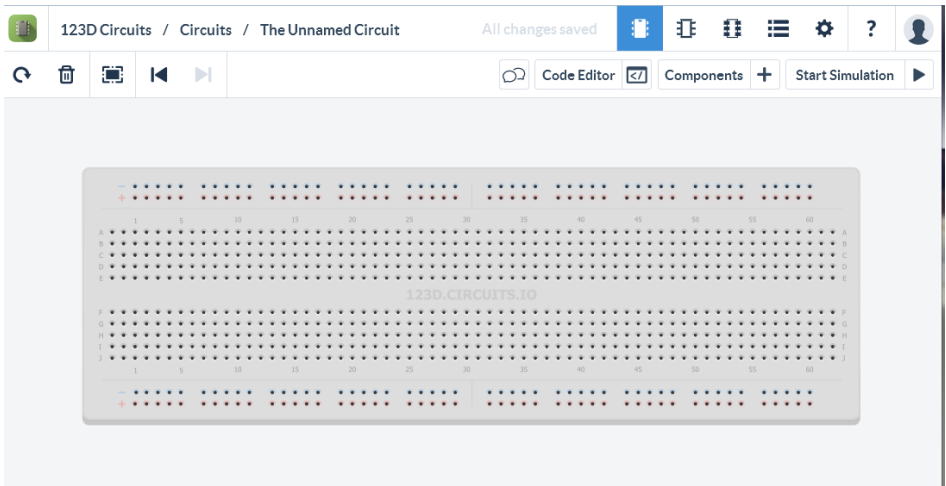


Рисунок 1.3 – Інтерфейс розробки проєкту схеми

Для розуміння роботи в середовищі Tinkercad, розробимо простий проєкт миготіння світлодіода. Створюємо схему за прикладом як зображено на рисунку 1.4.

Клікаємо на кнопку Components яка знаходиться на панелі керування проєктом праворуч. У вікні компонентів яке відкрилося знаходимо плату Arduino Uno, обираємо її і розміщуємо її на робочій поверхні. Оскільки даний стимулятор не має в базі платформи Arduino

Mega2560 ми обираємо Arduino Uno, робота з цими платами Arduino ідентична. Далі обираємо LED (світлодіод) і розміщуємо її на Bredbord. Відкривається вікно налаштування параметрів світлодіода, де можемо задати його ім'я та колір. Після встановлюємо резистор і також задаємо його параметри: ім'я та опір 100 Ом. Як всі компоненти встановлені, з'єднуємо їх.

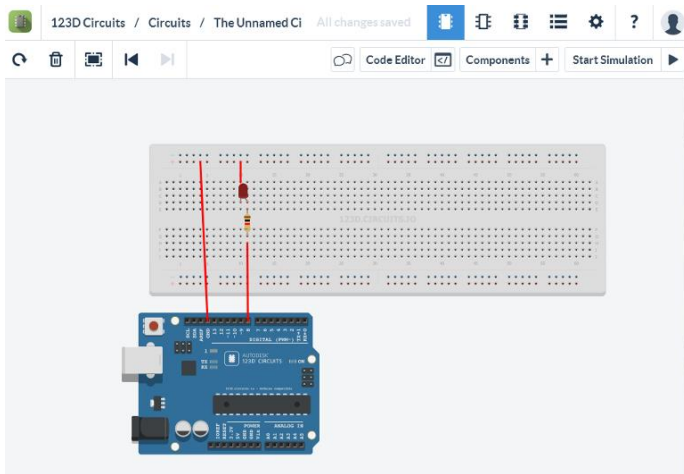


Рисунок 1.4 – Приклад схеми

Викличемо вікно редагування коду клікнувши на кнопку «Code Editor» яка знаходиться праворуч на панелі керування проєктом. У редактор копіюємо код:

```
int led = 8;

void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```

Компілюємо і завантажуюмо код клікнувши на кнопку «Apload&Run».

2. Завдання до самостійної роботи

2.1 Розробіть у Tinkercad проєкт підключення світлодіода. За допомогою мультиметра зробіть вимірювання напруги, яку дає плата Arduino Uno, встановіть відповідний резистор для обраного світлодіода.

2.2 Розробіть проєкт мерехтіння 6 світлодіодів використовуючи цикл FOR.

2.3 Розробіть у Tinkercad будь-який проєкт з підключенням осцилографа.

3. Робота з платою Arduino

3.1 Виконайте підключення плати Arduino до комп'ютера та напишіть програму мерехтіння світлодіоду, який встановлений за замовчанням на 13 виході плати. Для цього:

3.2 Запустіть на комп'ютері середовище розробки Arduino

3.3 Написати код.

3.4 Підключіть плату Arduino через USB до ПК.

3.5 Натисніть кнопку Verify и переконайтесь, що у нижній частині вікна з'явився надпис Done Compiling. Це значить, що у написаній програмі не знайдено помилок.

3.6 Виберіть у Tools->Board ваш тип плати. Перевірте, чи правильно обрано USB-порт в Tools->Serial port. Після натисніть на кнопку Upload.

3.7 Якщо внизу з'явився надпис “Done uploading” – процес запису прошов успішно.

3.8 Підключіть 3 світлодіод (послідовно красний. Жовтий і зелений) до плати Arduino. Написати код згідно з алгоритмом представленим на рисунку 1.5.

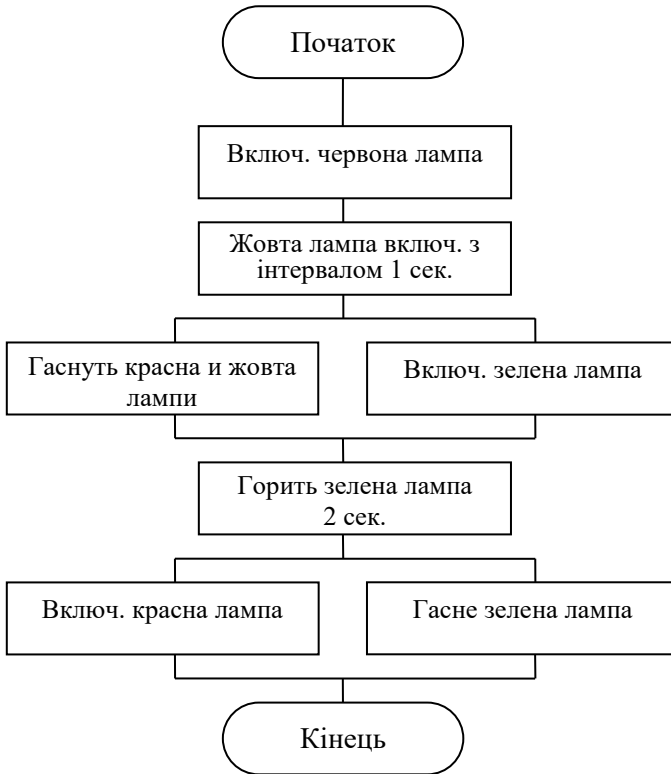


Рисунок 1.5 – Схема алгоритму роботи системи індикації

4. Зміст звіту

- 4.1 Тема та мета роботи
- 4.2 Три схеми проєкту: компонентна, принципова, друкована.
- 4.3 Код програми
- 4.4 Висновки з роботи

5. Контрольні запитання

5.1 У якому режимі здійснюється макетне моделювання в сервісі Tinkercad?

5.2 До якого моменту буде працювати плата Arduino?

5.3 До якого виводу підключено вбудований світлодіод на платформі Arduino Uno?

5.4 Які оператори середовища Arduino треба використовувати, щоб створити проект керування світлодіодами?

5.5 Що таке RX та TX на платі Arduino?

5.6 Скільки усього входів/виходів має платформа Arduino Uno? Назвіть основні з них.

САМОСТІЙНА РОБОТА № 2

СТВОРЕННЯ ПРОЄКТУ СИСТЕМИ ІНДИКАЦІЇ

Мета роботи: ознайомитись з середовищами Proteus VSM та Atmel Studio 6.2. Навчитися створювати проєкт віртуальної симуляції роботи світлодіода. Ознайомитись з різновидами функцій та команд для створення прототипу системи індикації.

1. Теоретичні відомості

1.1 Створення проєкту програми для мікроконтролера у Atmel Studio 6.2

Atmel Studio – інтегроване середовище розробки (IDE) для розробки 8-ми і 32-х бітних додатків від компанії Atmel, що працює в операційних системах Windows NT/2000/XP/Vista/7. Atmel Studio містить асемблер і симулятор, що дозволяє відстежити виконання програми. Atmel Studio містить в собі менеджер проєктів, редактор вихідного коду, інструменти віртуальної симуляції та внутрішньосхемного налагодження, дозволяє писати програми на асемблері або на C / C ++.

Після того як ви встановите *Atmel Studio 6.2* на свій комп'ютер, створіть новий проєкт:

1. Відкрийте Atmel Studio 6.2 та виберіть **New Project**. У діалоговому вікні New Project, виберіть **GCC C Executable Project** як шаблон (рис.2.1). Введіть ім'я проєкту і вкажіть місце де він буде зберігатися. Назвемо наш проєкт "BlinkLED". Зніміть прапорець **Create directory for solution** для спрощення структури каталогів проєкту. Клікніть **OK**.

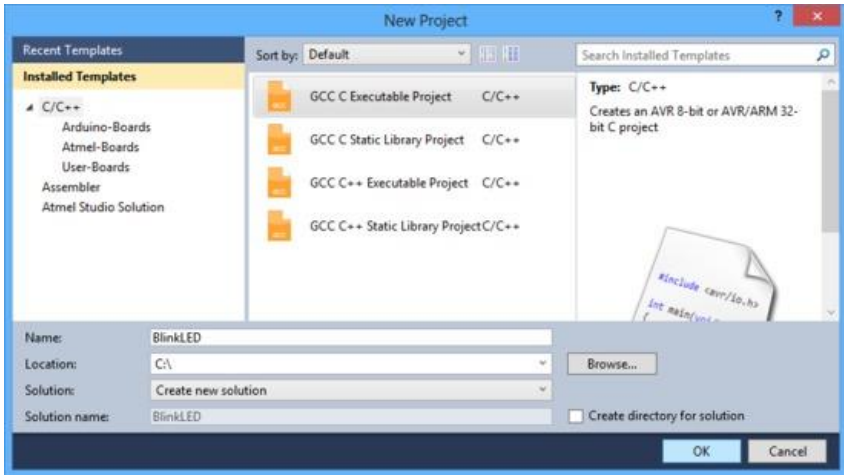


Рисунок 2.1 – Діалогове вікно New Project в Atmel Studio 6

2. У вікні Device Selection (рис. 2.2), оберіть ім'я пристрою AVR яке необхідно. Клікніть **ОК** для створення нового проєкту.

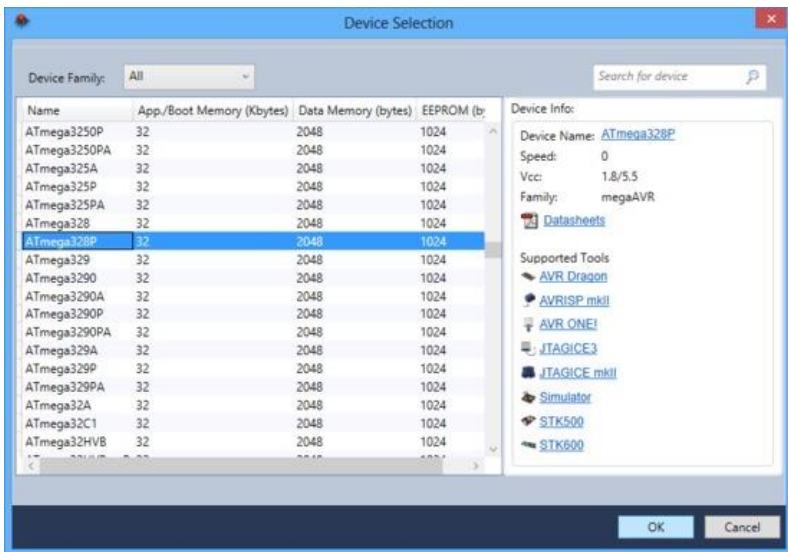


Рисунок 2.2 – Діалогове вікно Device Selection в Atmel Studio 6

3. Напишіть код у BlinkLED.cpp приведений нижче:

```
#define F_CPU 8000000      // частота AVR в Гц,
                           // використовується для util/delay.h
#include <avr/io.h>
#include <util/delay.h>

int main() {
    DDRD |= (1<<DDD1);      // установка LED виходу
    PD1 на вихід
    while (1) {
        PORTD |= (1<<PORTD1); // PD1 включення
        _delay_ms(50);        // затримка на 50 мс
        PORTD &= ~(1<<PORTD1); // PD1 виключення
        _delay_ms(100);       // затримка 100 мс
    }
}
```

4. На панелі інструментів клікніть кнопку **Build Solution** (або натисніть **F7**), щоб скомпілювати код (рис.2.3).

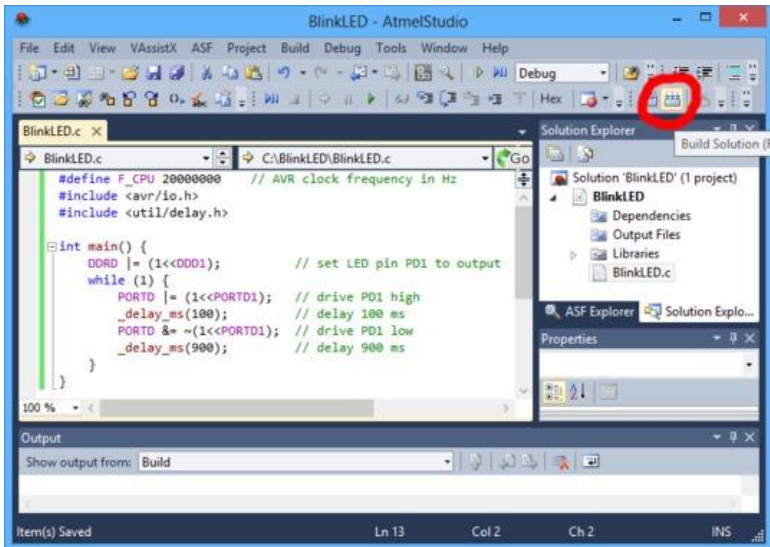


Рисунок 2.3 – Побудова проекту

Після компіляції перейдіть в директорію проєкту і знайдіть файл BlinkLED.hex – це файл, який представляє собою довільні двійкові данні у текстовому вигляді. По історичним причинам є стандартом де-факто при прошивці різноманітних мікросхем. Саме його необхідно буде завантажити у мікроконтролер схеми у Proteus VSM, яка буде побудована далі.

1.2 Симуляція роботи мікроконтролера за допомогою Proteus VSM

На рисунку 2.4 показане розведення виводів мікроконтролера ATmega8. Світлодіод ми будемо підключати до третього виводу мікроконтролера (PD1).

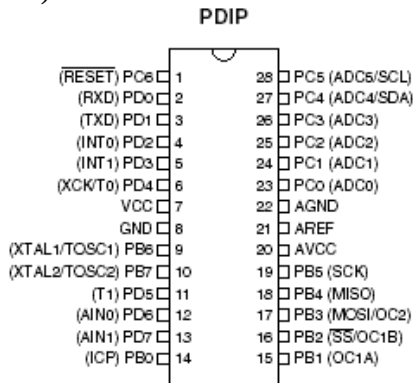


Рисунок 2.4 – Розташування виводів мікроконтролера ATmega8

Створіть новий проєкт у Proteus VSM. І розкресліть схему, як зображено на рисунку 2.5. У менеджері компонентів знайдіть нижче приведені елементи:

1. світлодіод
2. мікроконтролер ATmega328P
3. резистор
4. заземлення

Для того щоб створити елемент «заземлення» (GROUND) перейдіть у панелі інструментів (знаходиться вертикально зліва) на вкладку Terminals Mode ().

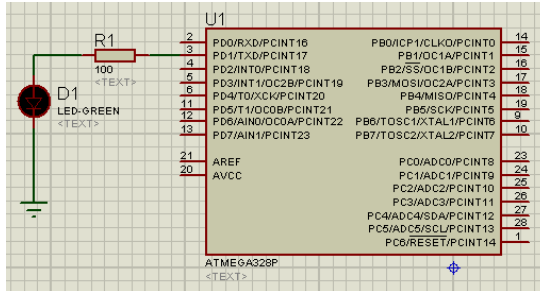


Рисунок 2.5 – Схема підключення

Після з'єднання елементів у електричне коло, двійним кліком клікніть на зображені мікроконтролера. Відкриється вікно Edit Component (рис. 2.6), в якому у полі Program File вказуємо шлях до файлу BlinkLED.hex. також перевірте щоб була встановлена частота мікроконтролера така сама як і у програмному коді, тобто 8МГц.

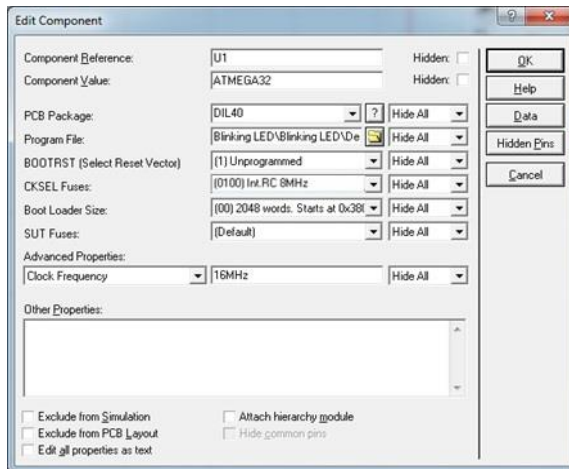
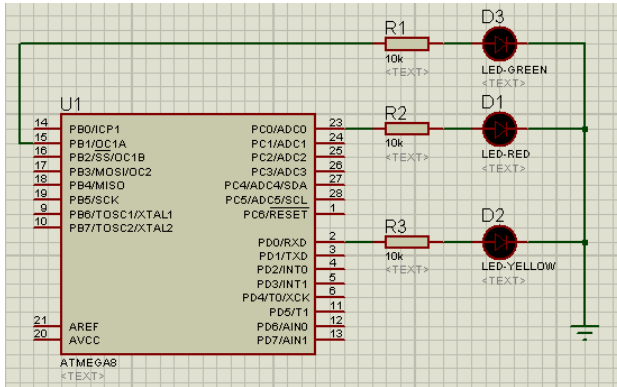


Рисунок 2.6 – Вікно Edit Component

Після на панелі керування симуляцією клікаємо на Play, і якщо все правильно було зроблено, споглядаємо роботу схеми.

2. Завдання до самостійної роботи

3.9 Розробіть програму і виконати симуляцію мерехтіння трьох різних світлодіодів підключених до мікроконтролера ATmega8. Схема підключення у середовищі Proteus VSM приведена нижче:



3. Зміст звіту

- 3.1. Тема та мета роботи
- 3.2. Схема електрична-принципова
- 3.3. Код програми
- 3.4. Висновки з роботи

4. Контрольні запитання

- 4.1 Що таке Нех-файл?
- 4.2 Назвіть послідовність етапів створення нового проєкту в середовищі Atmel Studio
- 4.3 Яке сімейство мікроконтролерів використовувалося при розробці схеми у цій роботі?
- 4.4 При розробці керуючої програми, у яких одиницях вимірювання задається затримка виконання команди мікроконтролера?

САМОСТІЙНА РОБОТА № 3 ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ РОБОТИ З АКТУАТОРАМИ

Мета роботи: ознайомитись з основними принципами роботи та підключення RGB світлодіоду, кнопки та змінного резистора до плати Arduino.

1. Теоретичні відомості

Розглянемо роботу RGB світлодіоду з платою Arduino. Для управління кольором світлодіоду буде використовуватись функція `analogWrite()`. Якщо задіяти на платі контакти, з відміткою «~», ми можемо регулювати напругу, яке подається на відповідний світлодіод.

RGB світлодіод має 4 виводи (рис.3.1).

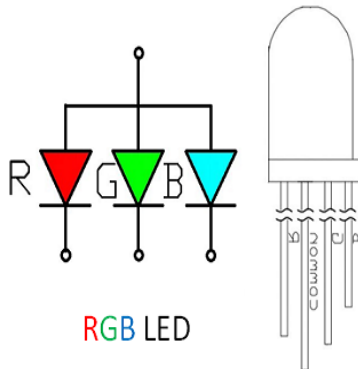


Рисунок 3.1 – RGB світлодіод

На перший погляд, RGB (Red, Green, Blue) світлодіод виглядає як звичайний регулярний світлодіод, однак всередині RGB насправді 3 світлодіоди: один червоний, один зелений і один синій.

3 виводи, кожний окремо, підключено до позитивного виводу кожного одиночного світлодіоду усередині RGB світлодіоду та один з виводів підключається до негативного, який є загальним для виводу

усіх трьох світлодіодів. До загального негативного виводу підключається заземлення (GROUND).

Шляхом контролю яркості кожного індивідуального світлодіоду можна отримувати різні кольори, це схоже на змішування трьох кольорів фарби на палітрі. Жорсткий спосіб змінювати колір RGB світлодіоду – це використовувати різне значення резисторів.

Розглянемо невеликий проект з RGB світлодіодом. Для кожного світлодіода потрібен відповідний резистор на 270 Ом, щоб запобігти можливості протікання занадто великих токів. Ці резистори встановлюються в коло між катодами (червоний, зелений и синій) та керуючими пінами на Arduino (рис. 3.2).

Якщо використовується RGB світлодіод с загальним анодом, замість загального катода, тоді загальний пін світлодіода підключається до піна +5 V замість піна gnd.

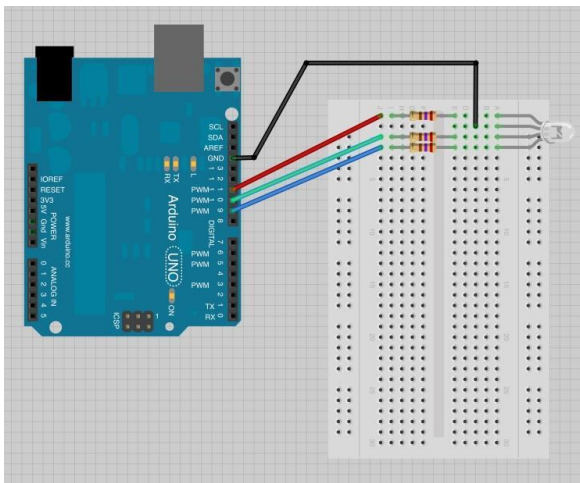


Рисунок 3.2 – Схема відключення

Але більш зручний спосіб – це використання змінного резистора.

Змінний резистор (або потенціометр) – це змінний резистор з регульованим опором. Потенціометри використовуються в робототехніці як регулятори різних параметрів – гучність звуку, потужності, напруги тощо.

Потенціометр має три контакти (рис. 3.3). Середній контакт йде на аналоговий вихід. Зовнішні контакти йдуть до піна із живленням и до землі.



Рисунок 3.3 – Зображення потенціометра

Для детального розуміння роботи потенціометра розглянемо приклад його роботи.

На рисунку 3.4 зображена схема підключення потенціометра до плати. У цьому випадку потенціометр підключений між землею і +5 V потенціалами (оскільки плата Arduino живиться +5 V), а движок сполучений з каналом аналого-цифрового перетворювача мікроконтролера. У такому разі можна регулювати вихідну напругу потенціометра в межах від 0 до 5 V.

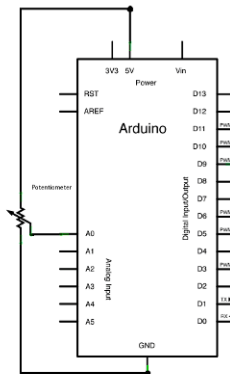


Рисунок 3.4 – Схема підключення потенціометра до плати Arduino

Код програми для роботи потенціометра:

```
void setup() {
  Serial.begin(9600);
}
```

```
void loop() {
    int sensorValue = analogRead(A0);  // отримуємо
    поточне значення
    Serial.println(sensorValue, DEC); // виводимо
    результат на монітор
}
```

При ініціалізації встановлюємо потрібну швидкість зв'язку: **Serial.begin(9600);**. Далі в циклі ми постійно зчитуємо дані, що поступають з потенціометра за допомогою методу `analogRead()`. Оскільки значення знаходитимуться в діапазоні від 0 до 1023, можемо використовувати тип `int` для змінної `sensorValue`.

Отриманий результат будемо виводиться у вікно послідовного монітора в десятковому форматі.

Далі ми розглянемо роботу з кнопкою. В одній з попередніх робіт вже було розглянуте загальне визначення кнопки.

На рисунку 3.5 зображена схема підключення кнопки до піна плати Arduino.

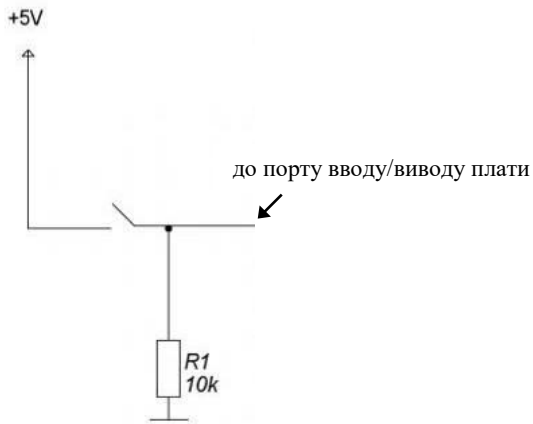


Рисунок3.5 – Схема підключення кнопки

Пін плати на вході повинен мати стан 0 або 1. Коли стан «підвішений», тобто не визначений, на вході будуть збиратися різні зовнішні наводки (статичні, електричні, електромагнітні), що буде призводити до хибного спрацювання кнопки. Щоб підвести пін до 0 або 1 використовують підтягуючі резистори. Вони бувають двох

видів, верхньої або нижньої підтяжки. Верхні підключаються до плюса живлення, а нижні до мінуса.

Після підключення схеми переходимо до написання програми. Необхідно пін, до котрого буде підключатися кнопка, ініціалізувати на вхід за допомогою команди:

```
pinMode(buttonPin, INPUT); //де buttonPin – це номер піна до якого підключена кнопка
```

Також необхідно об'явити змінну в яку буде записуватися статус кнопки.

2. Завдання до самостійної роботи

2.1 Виконайте підключення до 5 порту звичайного світлодіода. Підключіть до 6 порту кнопку. Напишіть програму, щоб при натисканні кнопки світлодіод, підключений до 5 порту, вмикався, а вбудований світлодіод до 13 порту вимикався.

2.2 При повороті ручки потенціометра змінювати яскравість світлодіоду. Схема підключення зображена на рисунку 3.6.

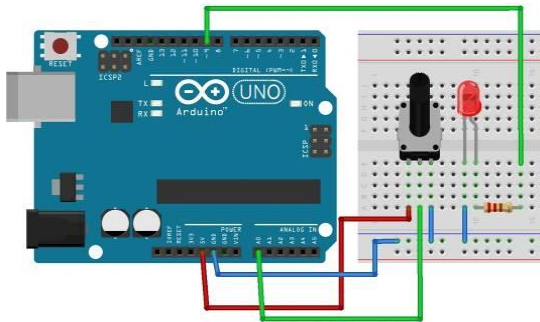


Рисунок 3.6 – Схема підключення

2.3 Підключіть RGB світлодіод до плати та по чергово змінюйте значення опору резисторів.

3. Зміст звіту

3.1. Тема та мета роботи

3.2. Схема електрична-принципова

3.3. Код програми

3.4. Висновки з роботи

4. Контрольні запитання

4.1 Як працює потенціометр?

4.2 До яких виводів підключається потенціометр в схемі керування на платформі Arduino

4.3 Які виводи має RGB світлодіод?

4.4 Які функції Arduino використовуються для управління кольором RGB світлодіоду?

4.5 З якою метою в схемі підключення кнопки до Arduino Uno використовуються підтягуючі резистори?

САМОСТІЙНА РОБОТА № 4

ДОСЛІДЖЕННЯ ТА ВИКОРИСТАННЯ СЕРВОДВИГУНІВ

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.4.1). Відмінні особливості обумовлені наступними факторами:

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

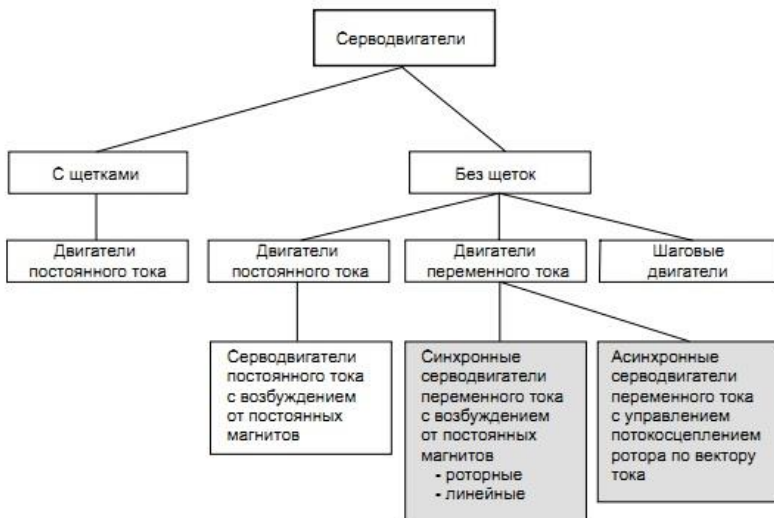


Рисунок 4.1 – Огляд серводвигунів

Будова сервоприводу для радіокерованих моделей зображена на рисунку 4.2. 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 4.2 – 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. 4.3). Зазвичай реальний діапазон може бути трохи ширше (наприклад, 900-2100мкс), але це вже залежить конкретно від виробника сервоприводів.

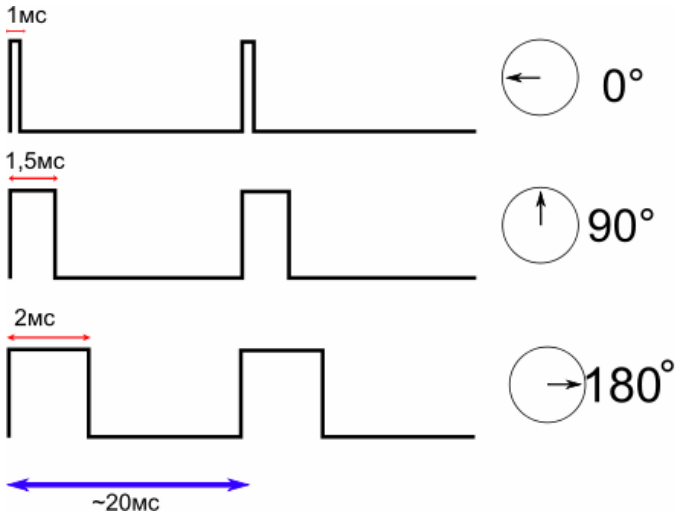


Рисунок 4.3 – Керуючі сигнали

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: Motor, Logic State (рис.4.4).

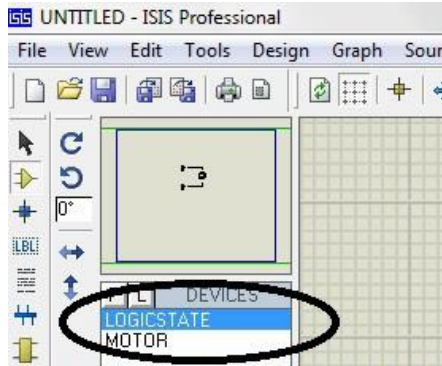
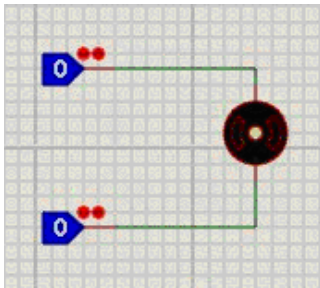
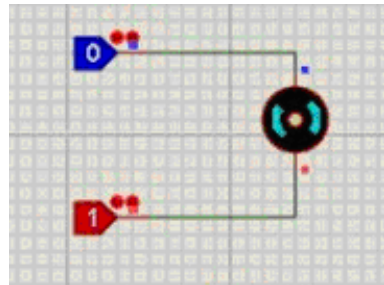


Рисунок 4.4 - Вибір компонентів з бібліотеки Proteus

Logic State має два стани 1 і 0. Коли 0 це означає 0В і коли 1 - 5V. Тепер розробіть схему, як показано на рис. 4.5.



а



б

Рисунок 4.5 – Схема з різними станами для серводвигуна

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 проєкт. У бібліотеці елементів знаходимо сервопривід та розміщуємо так як зображено на рис. 4.6. На ці три сервоприводи ми будемо відповідно надсилати сигнали різної довжини: 1мс, 1.5мс та 2 мс.

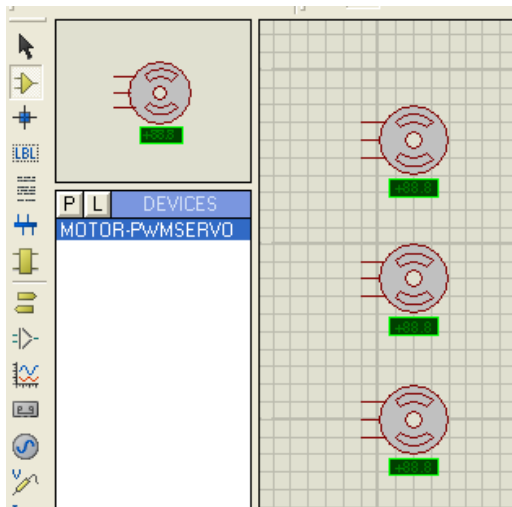


Рисунок 4.6 – Сервоприводи для проєкту

Встановлюємо елемент живлення «POWER» та у його властивостях задаємо живлення «+12V» (рис. 4.7).

Після встановлення заземлення електрична схема буде виглядати так як зображено на рис. 4.8.

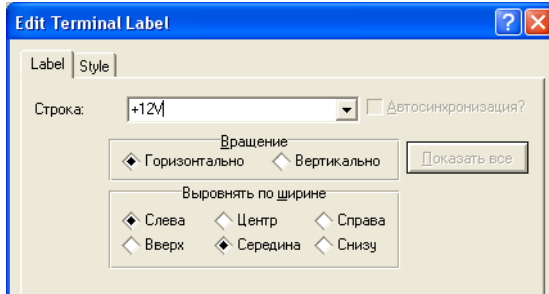


Рисунок 4.7 – Встановлення параметру елемента живлення

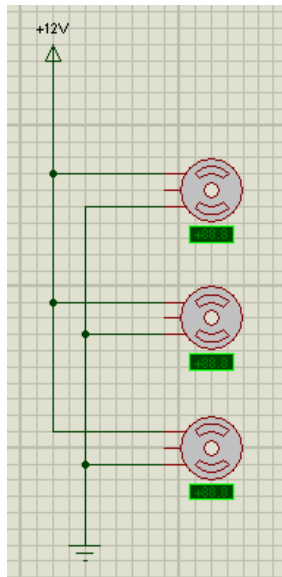


Рисунок 4.8 – Схема підключення сервоприводів

Для генерування сигналів потрібної частоти та довжини встановимо елементи, які будуть генерувати три різні сигнали на кожен з сервоприводів. Перейдіть у лівій панелі інструментів у режим Generator Mode та виберіть компонент PULSE. Встановіть три компоненти PULSE до схеми та з'єднайте їх як зображено на рис. 4.9.

Для першого зверху генератора імпульсу встановить параметр довжини (ширина) імпульсу 1мс та частота 50Гц (рис. 4.10).

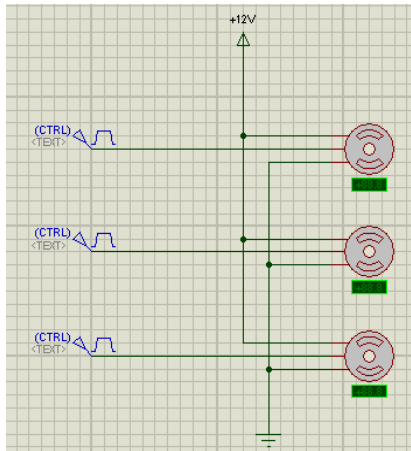


Рисунок 4.9 – Схема підключення генераторів сигналу

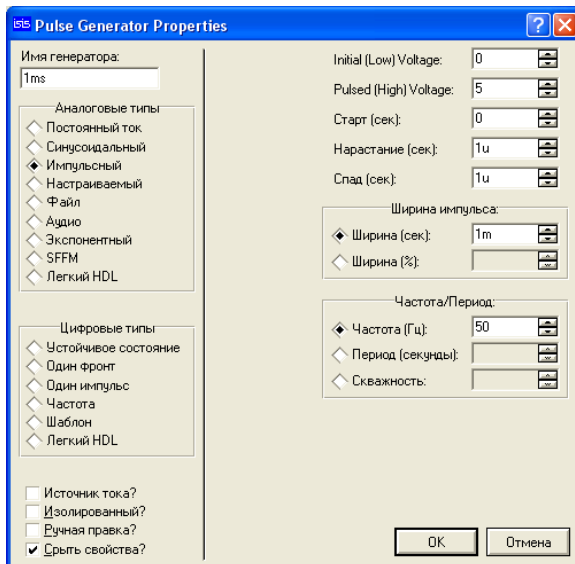


Рисунок 4.10 – Встановлення параметрів сигналу

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 рис. 4.11.

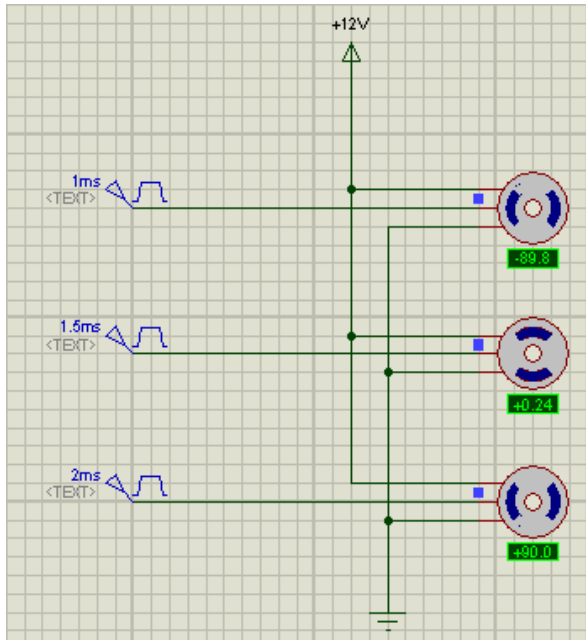


Рисунок 4.11 – Результат симуляції

2.2 Створення проєкту з сервоприводом на Arduino

Для розглядання прикладу керування положенням сервоприводу за допомогою Arduino необхідна плата та сервопривід. Схема підключення зображена на рисунку 4.12. 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.

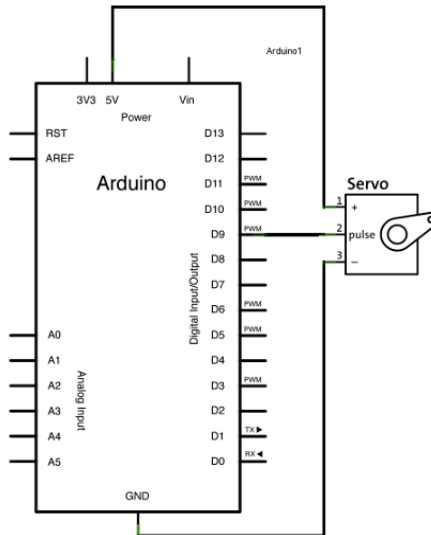


Рисунок 4.12 – Схема підключення сервоприводу до Arduino

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
  {
    // с кроком в 1 градус
    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 самостійної 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.

3.3 Проаналізувати проєкт, представлений на <https://arduino-kit.ru/product/modul-xbee-s1-xb24-awi-001>. Розглянути можливості організації управління на основі Модуля XBee S1 XB24-AWI-001 та серводвигунів.

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 на платформі Arduino?
- 5.5 Назвіть основні команди Arduino для керування сервоприводом?

САМОСТІЙНА РОБОТА № 5

ДОСЛІДЖЕННЯ ОСОБЛИВОСТЕЙ РОБОТИ З РЕЛЕ

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

1. Теоретичні відомості

Реле є найкращим способом управління будь-яким двигуном. У цій роботі ми будемо розбиратися як контролювати реле в Proteus ISIS. Реле є одним з ключових компонентів будь-якого електронного або електричного кола і, як правило, є проблемою при проєктуванні.

Спочатку змодельємо реле в простій схемі, в якій при запуску моделювання, реле автоматично активується, а після цього будемо керувати реле за допомогою логіки, тобто, коли ви подаєте + 5V до нього, то реле буде активуватися і коли ви даєте Gnd, то він буде знеструмлений. Крім того, розглянемо, що таке реле і як його використовувати?

Для управління різними виконавчими пристроями, комутації ланцюгів, управління приладами в електроніці активно застосовуються електромагнітні реле (рис.5.1).



Рисунок 5.1 – Зовнішній вигляд електромагнітного реле

Будова реле досить проста. Його основою є котушка, що складається з великої кількості витків ізолюваного проводу. Всередину котушки встановлюється стрижень з м'якого заліза. В результаті виходить електромагніт. Також в конструкції реле присутній ярмі. Він закріплений на пружному контакті. Сам же пружний контакт закріплений на ярмі. Разом зі стрижнем і ярмом ярмо утворює магнітопровід.

Якщо котушку підключити до джерела струму, то магнітне поле, що утворилося намагнічує сердечник. Він в свою чергу притягує якір. Якір укріплений на пружному контакті. Далі пружний контакт замикається з іншим нерухомим контактом. Залежно від конструкції реле, якір може по-різному механічно управляти контактами.

Внутрішня будова електромагнітного реле. У більшості випадків реле монтується в захисному корпусі. Він може бути як металевим, так і пластмасовим. Подивімося на те, що знаходиться всередині цього реле (рис.5.2).

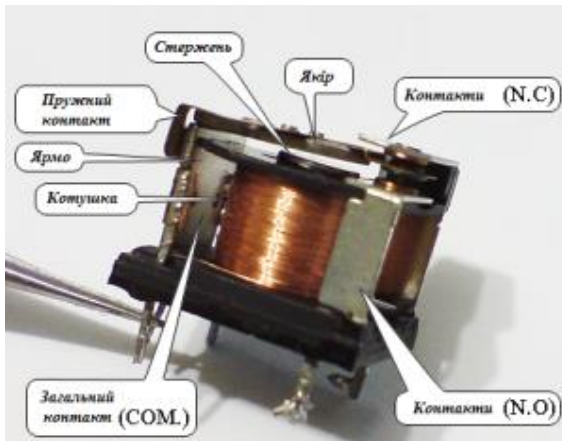


Рисунок 5.2 – Внутрішня будова реле

Це реле без захисного корпусу. Як бачимо, реле має котушку, стрижень, що пружний контакт, на якому закріплений якір, а також виконавчі контакти.

На принципових схемах електромагнітне реле позначається наступним чином (рис.5.3).

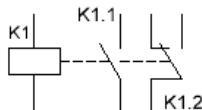


Рисунок 5.3 – Принципова схема електромагнітного реле

Умовне позначення реле на схемі складається наче з двох частин. Одна частина (**K1**) – це умовне позначення електромагнітної котушки. Вона позначається у вигляді прямокутника з двома виводами. Друга частина (**K1.1**; **K1.2**) – це групи контактів, якими управляє реле. Залежно від своєї складності реле може мати досить велику кількість комутованих контактів. Вони розбиваються на групи. Як бачимо, на позначенні зображені дві групи контактів (**K1.1** і **K1.2**).

Як тільки ми замкнемо керуючий ланцюг вимикачем **SA1**, струм від батареї живлення **G1** надійде на реле **K1**. Реле спрацює, і його контакти **K1.1** замкнуть виконавчий ланцюг. На навантаження надійде напруга живлення від батареї **G2** і лампа **HL1** засвітиться. Якщо розімкнути ланцюг вимикачем **SA1**, то з реле **K1** буде знята напруга живлення та контакти реле **K1.1** знову розімкнуться і лампа **HL1** вимкнеться.

Комутовані контакти реле можуть мати своє конструктивне виконання. Так, наприклад, розрізняють нормально-розімкнуті контакти, нормально-замкнуті контакти та контакти на перемикання (перекидні). Розберемося з цим детальніше.

Нормально розімкнуті контакти – це контакти реле, які знаходяться в розімкнутому стані до тих пір, поки через котушку реле не потече струм. Говорячи простіше, коли реле вимкнено, контакти теж розімкнуті. На схемах реле з нормально-роз'єднаними контактами позначається як на рисунку 5.4.

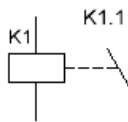


Рисунок 5.4 – Схема реле з нормально розімкнутими контактами

Нормально замкнуті контакти – це контакти реле, що знаходяться в замкнутому стані, поки через котушку реле не почне текти струм. Таким чином, виходить, що при вимкненому реле контакти замкнуті. Такі контакти на схемах зображують як на рисунку 5.5.

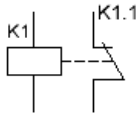


Рисунок 5.5 – Схема реле з нормально зімкнутими контактами

Перекидні контакти – це комбінація з нормально-замкнутих і нормально-розімкнутих контактів. У перекидних контактів є спільний провід, який переключається з одного контакту на інший (рис.5.6).

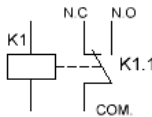


Рисунок 5.6 – Схема реле з перекидними контактами

Сучасні широко розповсюдженні реле, як правило, мають перекидні контакти, але можуть зустрічатися і реле, які мають у своєму складі тільки нормально-розімкнуті контакти.

У імпортних реле нормально-розімкнуті контакти реле позначаються скороченням **N.O.** А нормально-замкнуті контакти **N.C.** Загальний контакт реле має скорочення **COM.** (від слова *common* – «загальний»).

Параметри електромагнітних реле. Як правило, розміри самих реле дозволяють наносити на корпус їх основні параметри. Як приклад, розглянемо імпортне реле Bestar BS-115C. На його корпусі нанесені такі написи, як на рис. 5.7.



Рисунок 5.7 – Зовнішній вигляд реле

COIL 12VDC – це номінальна напруга спрацьовування реле (12V). Оскільки це реле постійного струму, то вказано скорочене позначення постійної напруги (скорочення DC позначає постійний струм/напруга). Англійське слово **COIL** перекладається як «котушка», «соленоїд». Воно вказує на те, що скорочення 12VDC має відношення до котушки реле.

Далі на реле вказані електричні параметри його контактів. Зрозуміло, що потужність контактів реле може бути різною. Це залежить як від габаритних розмірів контактів, так і від використовуваних матеріалів. При підключенні навантаження до контактів реле потрібно знати потужність, на яку вони розраховані. Якщо навантаження споживає потужність більше тієї, на яку розраховані контакти реле, то вони будуть нагріватися, іскрити, «залипати». Природно, це призведе до швидкого виходу з ладу контактів реле.

Для реле, як правило, зазначаються параметри змінного і постійного струму, які здатні витримати контакти.

Так, наприклад, контакти реле Bestar BS-115C здатні комутувати змінний струм в 12A і напругу 120V. Ці параметри зашифровані в написі **12A 120VAC** (скорочення AC позначає змінний струм).

Також реле здатне комутувати постійний струм силою 10A і напругою 28V. Про це свідчить напис **10A 28VDC**. Це були силові характеристики реле, точніше його контактів.

Споживана потужність реле. Тепер звернемося до потужності, яку споживає реле. Як відомо, потужність постійного струму дорівнює добутку напруги (U) на струм (I): $P = U * I$. Візьмемо значення номінальної напруги спрацьовування (12V) і споживаного струму (30 mA) реле Bestar BS-115C і отримаємо його споживану потужність (англ. – **Power consumption**).

Таким чином, потужність реле Bestar BS-115C становить 360 мільватт (mW).

Є ще один параметр – це чутливість реле. За своєю суттю, це і є потужність споживання реле у включеному стані. Зрозуміло, що реле, якому потрібно менше потужності для спрацьовування, є більш чутливим у порівнянні з тими, які споживають велику потужність. Такий параметр, як чутливість реле, особливо важливий для пристроїв з автономним живленням, так як включене реле витрачає заряд

батареї. Приміром, є два реле з споживаною потужністю 200 mW і 360 mW. Таким чином, реле потужністю 200 mW має більшу чутливість, ніж реле потужністю 360 mW.

2. Виконання роботи

Проста схема підключення Реле в Proteus ISIS.

- Відкрити Proteus ISIS и вибрати компоненти, які зображені на рис. 5.8, з бібліотеки компонентів Proteus.

Встановить та з'єднайте компоненти у схему, яку зображено на рис.5.9.

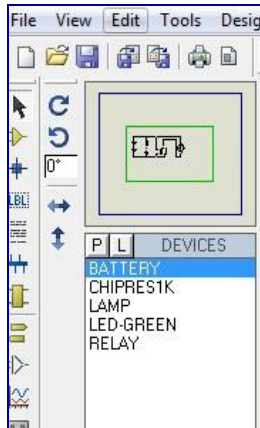


Рисунок 5.8 – Вікно вибору компонентів

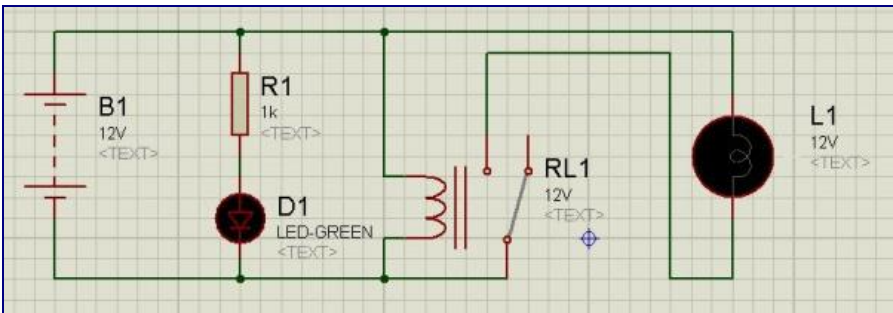


Рисунок 5.9 – З'єднання компонентів в схему.

Коли реле буде під напругою, лампа L1 буде світитись і коли реле знеструмлене, лампа залишиться вимкненою. Світлодіод D1 виконує роботу індикатора, який свідчитиме, що напруга в колі є.

Після розробки схеми, натисніть на кнопку запуску, і якщо все зроблено правильно, то лампа буде світитись як на рис. 5.10.

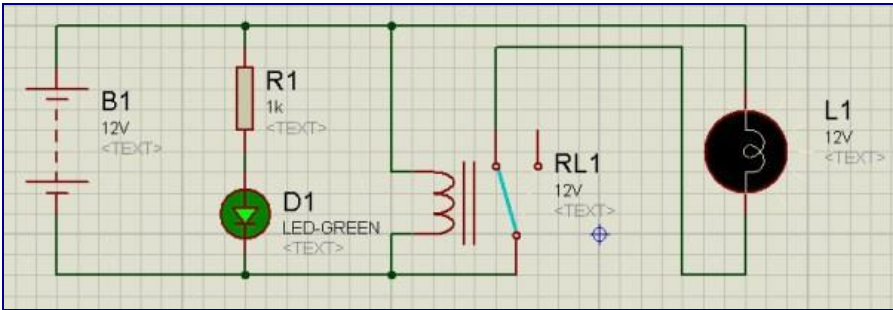


Рисунок 5.10 – Відстежування стану схеми

Отже, тепер ви можете побачити світлодіодний індикатор також загоряється, та бачимо, що реле тепер з'єднаний з другим терміналом і, таким чином замикає ланцюг для лампи і лампи тепер також світиться.

Керування реле за допомогою логіки в Proteus ISIS

У попередній роботі ми бачили просту схему, яка управляється вручну це означає, що для того, щоб включити або виключити його потрібно включити або виключити живлення, але, як правило, потрібно, щоб реле знаходилося під автоматичним контролем якогось мікроконтролера.

Зважаючи на те що мікроконтролери зазвичай працюють від 5V для того, щоб управляти реле 12V з використанням 5V мікроконтролера, ми повинні використовувати транзистор. У цьому випадку, коли ви даєте + 5V реле приводиться в дію, а коли ви даєте GND реле вимикається.

Знайдемо вказані компоненти (рис. 5.11) та спроектуємо схему (рис.5.12).

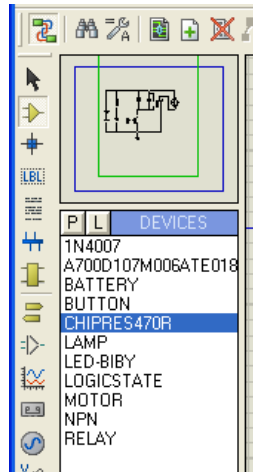


Рисунок 5.11 – Вікно пошуку компонентів

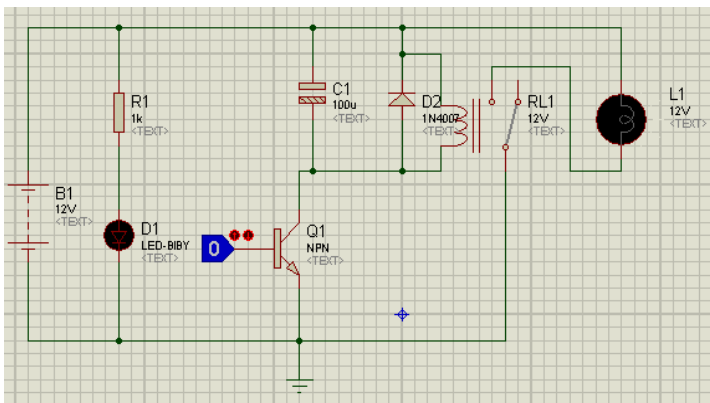


Рисунок 5.12 – Схема для створення

Ми можемо не використовувати мікроконтролер, замість нього будемо використовувати логічний стан, він буде працювати так само, як мікроконтролер. Таким чином, схема дуже схожа на просту схему, яку ми вже робили в попередній роботі. Єдина відмінність тут присутній n-p-n транзистор.

У цьому випадку ми не забезпечуємо подачу живлення безпосередньо до реле, замість цього ми забезпечуємо його за

допомогою цього транзистора. Так що, коли логічний стан дорівнює нулю (це означає землю), транзистор не працюватиме, і живлення не може досягти реле і коли ми робимо логічну 1, що означає + 5V на вході транзистора, то схема реле буде завершена і реле буде запитана.

Запускаємо симуляцію, вимкнений стан реле виглядає як на рис. 5.13. На даному рисунку можемо побачити, що світлодіодний індикатор працює тому, що живлення подається на схему, але лампа все ще вимкнена і на реле також не подається живлення, оскільки логічний стан 0.

Тепер натисніть на логічному стані та задайте логічну одиницю, тобто +5В, включений стан реле показано на рис. 5.14.

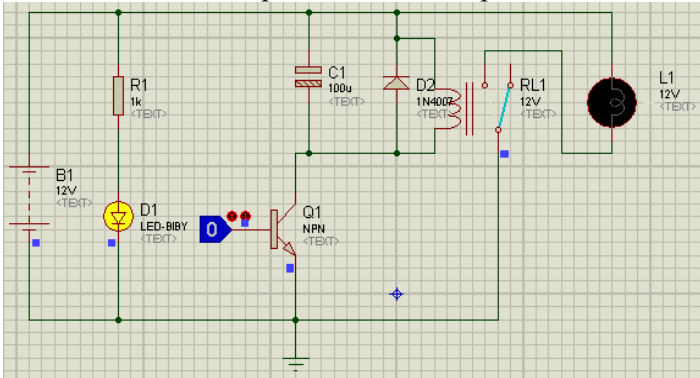


Рисунок 5.13 – Схема з вимкненим реле.

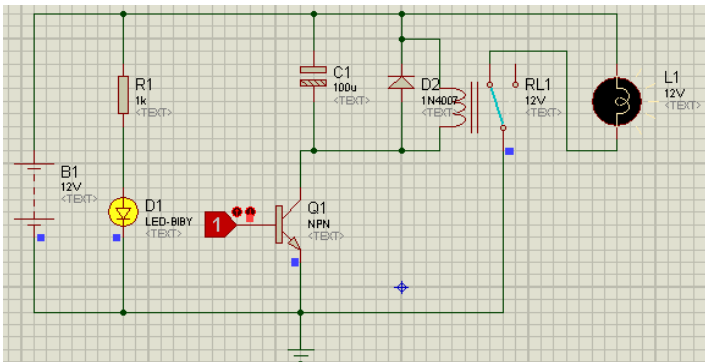


Рисунок 5.14 – Схема з включеним реле.

Робота Arduino з реле

Реле модуль має 3 виводи:

VCC: "+"

GND: "-"

IN: вивід вхідного сигналу

Підключення модулю дуже просте (рис. 5.15):

VCC на + 5 вольт на Arduino.

GND на любий з GND пінов Arduino.

IN на любий з цифрових входів/виходів Arduino.

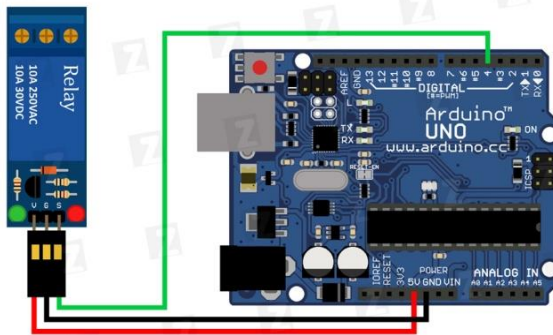


Рисунок 5.15 – Схема підключення до плати Arduino

Переходимо до створення програми. У даному прикладі реле буде включатись и виключатись з інтервалом у 2 секунди.

Приклад програмного коду:

```
// Реле модуль підключено до цифрового виводу 4
int Relay = 4;
void setup()
{
    pinMode(Relay, OUTPUT);
}
void loop()
{
    digitalWrite(Relay, LOW);    // реле вкл.
    delay(2000);
    digitalWrite(Relay, HIGH);  // реле викл.
    delay(2000);
}
```

3. Завдання до самостійної роботи

3.1 Накресліть схему (рис. 5.16) та знайдіть у ній помилки.

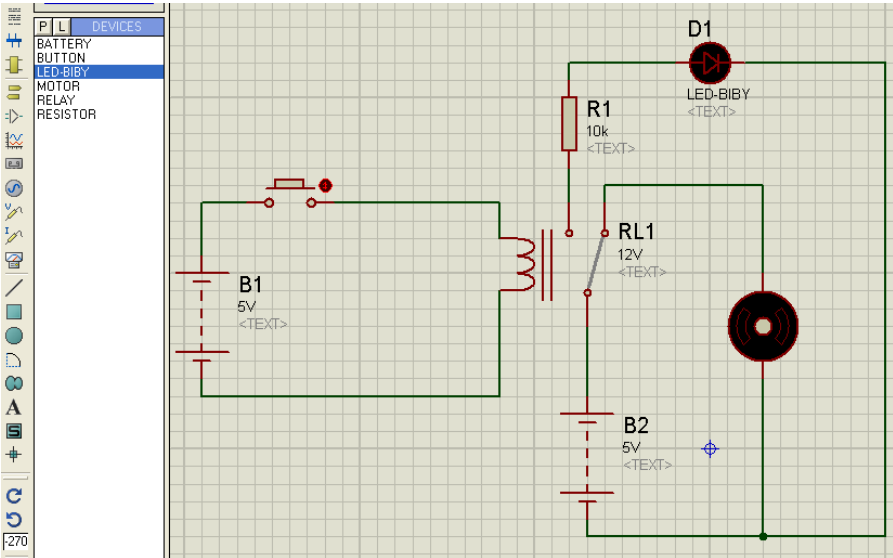


Рисунок 5.16 – Схема до виконання самостійної роботи.

3.2 Створіть проєкт на Arduino з використанням реле та промоделюйте його роботу в середовищі Tinkercad.

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 реле? Яка побудовано реле?
- 5.2 Які бувають комутативні контакти реле?
- 5.3 Принцип роботи біполярного транзистора?

САМОСТІЙНА РОБОТА № 6

РОБОТА З БІБЛІОТЕКОЮ ARDUINO. ПРОГРАМУВАННЯ LCD ДИСПЛЕЯ

Мета роботи: розробити проєкт схеми з підключенням плати Arduino у середовищі Proteus VSM. Ознайомитись з принципами програмування LCD дисплея.

1. Виконання роботи

У папці Arduino Lib знаходяться два бібліотечних файли Arduino (ARDUINO.IDX, ARDUINO.LIB). Вставте ці два файли у бібліотечну папку Proteus, яка знаходиться:

C: -> Program Files (x86) -> Labcenter Electronics -> Proteus 7 Professional -> LIBRARY

Відкриваємо Proteus ISIS та знаходимо у бібліотеці плату Arduino та LCD дисплей LM016L.

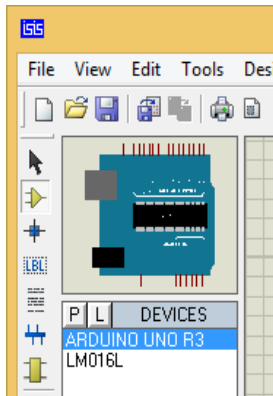


Рисунок 16.1 – Вибір плати з бібліотеки

З'єднуємо виходи Arduino з дисплеєм таким чином:

- LCD RS вихід до 12 виходу Arduino
- LCD Enable до 11
- LCD D4 до 5
- LCD D5 до 4

- LCD D6 до 3
- LCD D7 до 2
- LCD RS до терміналу GROUND.

Далі відкриваємо Arduino IDE.

Для того щоб бачити більш детально куди зберігаються файл з розширенням *.hex, який нам знадобиться для симуляції роботи Arduino у Proteus перейдіть Файл->Налаштування. Та налаштуйте Arduino IDE, щоб показувало докладний звіт при компіляції, поставивши галочку у потрібному місці.

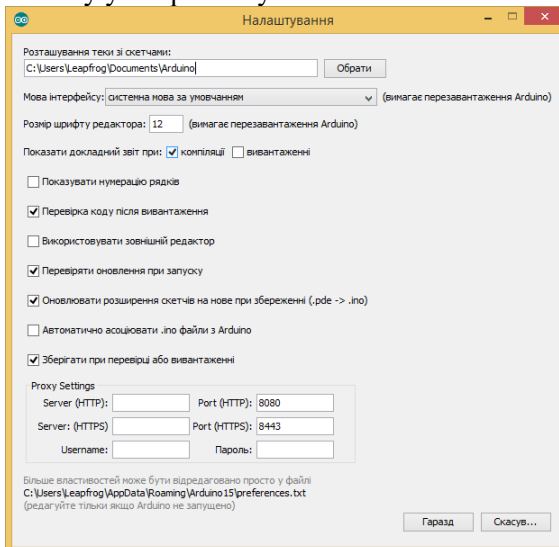


Рисунок 6.2 – Налаштування докладного звіту.

Копіюємо код приведений нижче, зберігаємо та компілюємо:

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the
// interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
```

```

// set up the LCD's number of columns and rows:
lcd.begin(16, 2);
// Print a message to the LCD.
lcd.print("hello, world!");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting
  begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis() / 1000);
}

```

У вікні результатів компіляції знаходимо папку до якої зберігається файл з розширенням *.hex. (рис.6.3).

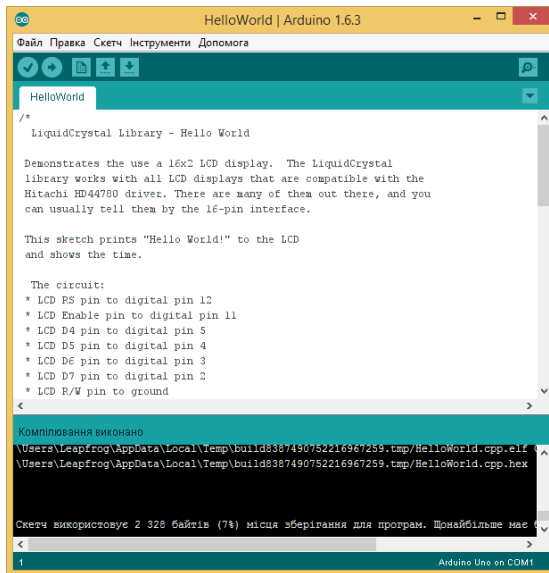


Рисунок 6.3 – Пошук теки з результатами компіляції.

Переходимо до цієї папки та копіюємо два файли (рис. 6.4) до своєї папки.

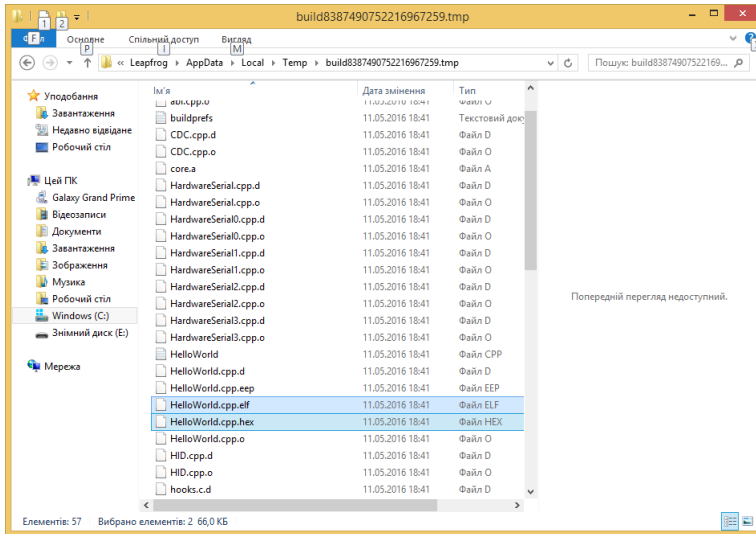


Рисунок 6.4 – Пошук теки з результатами компіляції.

Для Arduino Uno у Proteus вказуємо шлях до файлу з розширенням *.hex. Запускаємо симуляцію і дивимось результат (рис.6.5).

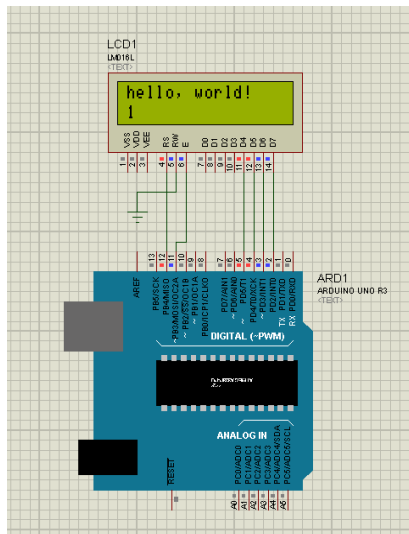


Рисунок 6.5 – Результати симуляції.

2. Завдання до самостійної роботи

2.1 Виконати симуляцію роботи з дисплеєм в Tinkercad <https://www.tinkercad.com/>.

2.2 Зареєструватися на <http://swed.zntu.edu.ua> та провести віддалений експеримент з дисплеєм у віддаленій лабораторії RELDES (REmote Laboratory for Design of Embedded Systems)

3. Зміст звіту

- 3.1. Тема та мета роботи
- 3.2. Схема електрична-принципова
- 3.3. Код програми
- 3.4. Висновки з роботи

4. Контрольні запитання

- 4.1 Основні функції Arduino для роботи з дисплеєм?
- 4.2 Які піни на платі Arduino використовуються для створення проєкту роботи з дисплеєм?
- 4.3 Особливості LCD I2C модуля?

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Інженерія вбудованих систем: навчальний посібник / Пархоменко А.В., Гладкова О.М., Залюбовський Я.І., Пархоменко А.В. - Запоріжжя: Дике поле, 2017.- 220 с.
2. Мілих, В.І. Електротехніка, електроніка та мікропроцесорна техніка: підручник для ВНЗ/В.І. Мілих, О.О.Шавьолкін; за ред. В.І.Мілих.-К.: Каравела, 2007.-688с.
3. Arduino. [Електронний ресурс]. – Режим доступу: <https://www.arduino.cc/>
4. Autodesk Tinkercad. [Електронний ресурс]. – Режим доступу: <https://www.tinkercad.com/>
5. Atmel Studio. [Електронний ресурс]. Microchip Atmel. – Режим доступу: <http://www.atmel.com/tools/atmelstudio.aspx>
6. Proteus VSM. Пошаговая отладка [Електронний ресурс]. – Режим доступу: <http://we.easyelectronics.ru/CADSoft/proteus-vsm-poshagovaya-otladka.html>