



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет
імені Івана Пулюя

**Кафедра автоматизації технологічних
процесів та виробництв**

Методичні вказівки
до лабораторної роботи №5
«Проектування систем керування в середовищі Proteus VSM на
базі Arduino Uno.
Вивід швидкості обертання двигуна на LCD дисплей »
з дисципліни
«Проектування мікропроцесорних систем керування
технологічними процесами»

Тернопіль 2022

Методичні вказівки до лабораторної роботи № 5 "Проектування систем керування в середовищі Proteus VSM на базі Arduino Uno. Вивід швидкості обертання двигуна на LCD дисплей" з курсу "Проектування мікропроцесорних систем керування технологічними процесами"/ Медвідь В.Р., Пісьціо В.П. – Тернопіль: ТНТУ, 2022. – 11 с.

Відповідальні за випуск

доцент, к.т.н. Медвідь В.Р.,
асистент Пісьціо В.П.

Методичні вказівки призначені для студентів спеціальності 151 "Автоматизація та комп'ютерно-інтегровані технології", котрі вивчають курс "Проектування мікропроцесорних систем керування технологічними процесами". Методичні вказівки можуть бути корисні для інших приладобудівних спеціальностей, а також для інженерів, що займаються розробкою мікропроцесорних систем на основі Arduino.

Лабораторна робота №5

Проектування систем керування в середовищі Proteus VSM на базі Arduino Uno. Вивід швидкості обертання двигуна на LCD дисплей

Схема з використання таймера NE555

Потрібно побудувати принципову електричну схему на основі Arduino Uno, яка б забезпечувала:

- ◇ керування швидкістю обертання кроковим двигуном в повнокроковому та напівкроковому режимах його роботи,
- ◇ можливість зміни напрямку обертання крокового двигуна в процесі його роботи,
- ◇ вивід на цифровий дисплей інформації про швидкість обертання крокового двигуна.

Для зменшення навантаження на програму роботи мікроконтролера, використаємо в схемі окремий генератор імпульсів із змінною частотою на основі мікросхеми-таймера NE555 для керування швидкістю обертання двигуна, а для керування його роботою в повнокроковому та напівкроковому режимах, а також зміни напрямку обертання, використаємо драйвери L297 та L298N.

Сам же мікроконтролер будемо використовувати для виводу на LCD індикатор інформації про швидкість обертів крокового двигуна.

На рис. 1 зображена схема типового увімкнення мікросхеми-таймера NE555 як генератора прямокутних імпульсів з можливістю зміни частоти імпульсів.

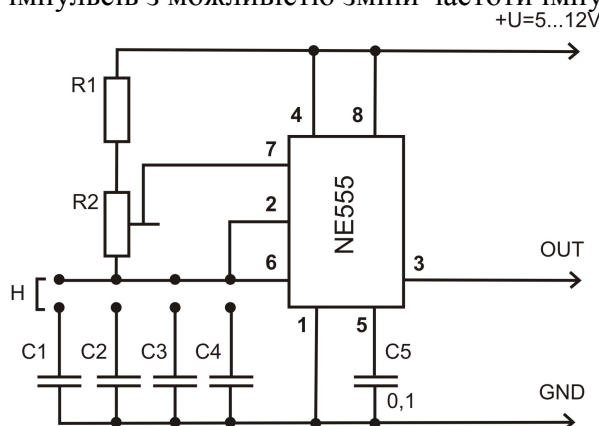


Рис. 1. Схема генератора прямокутних імпульсів з регулюванням частоти

Частота імпульсів в даній схемі визначається як:

$$f = \frac{1}{0.693 C (R1 + 2R2)}, \quad (1)$$

а період імпульсів, відповідно, як:

$$T = 0.693 C (R1 + 2R2). \quad (2)$$

Для вибору діапазону частоти імпульсів перемикачем Н вибирається один з конденсаторів конденсаторної батареї C1...C4, а резистором R2 змінюється частота імпульсів в межах вибраного діапазону. З виразу (1) видно, що значення частоти f буде зменшуватися із збільшенням величин C та R2 при сталому значенні R1.

Типова схема управління кроковим двигуном за допомогою комплексу мікросхем L297 та L298N показана на рис. 2. Мікросхема L298N є підсилювачем потужності, до виходів якої безпосередньо під'єднуються обмотки двигуна (рис. 2). Напруга живлення вихідних каскадів мікросхеми є більшою за напругу живлення контролера L297 (+5 В) і може досягати значень від +5 В до +46 В.

Контролер L297 генерує фазові групи для режимів повнокрокового, хвильового приводу та півкрокового режиму. Послідовність станів і вихідні сигнали для цих трьох режимів, показані нижче.

Послідовність тактових імпульсів прямокутної форми подається на вхід CLOCK. У всіх випадках стан виводів контролера змінюється при переході від низького до високого рівня на цьому вході.

RESET встановлює контролер у початковий стан.

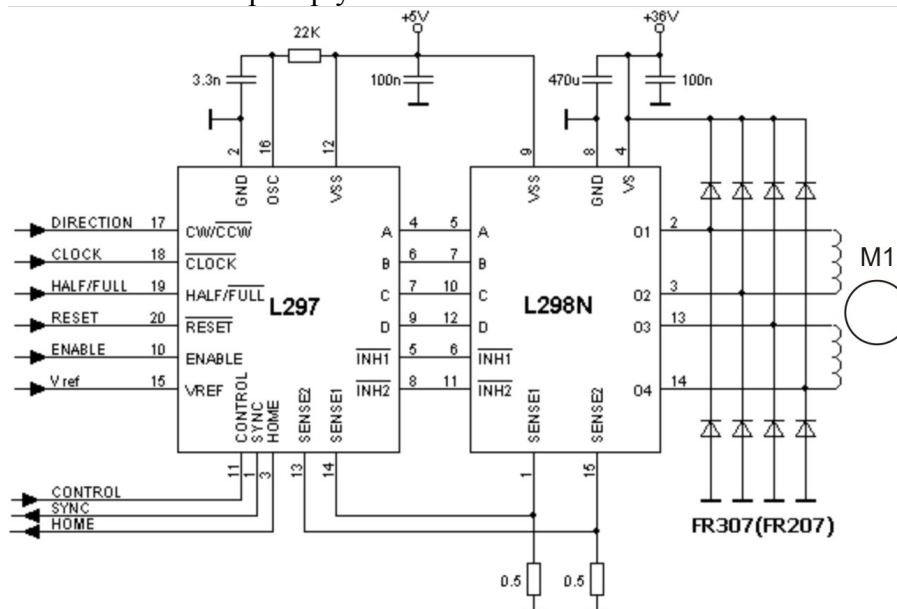



Рис. 2. Типова схема управління кроковим двигуном за допомогою мікросхем L297 та L298N

Режим роботи двигуна вибирається рівнем напруги на вході HALF/FULL. При логічній «1» буде вибрано напівкроковий режим роботи, при логічному «0» - повнокроковий.

Напрямок обертання двигуна визначається рівнем сигналу на вході CW/CCW. При логічній «1» двигун буде обертатися за годинниковою стрілкою, при логічному «0» - в протилежному напрямку.

Створення проекту з використанням Arduino Uno

Після інсталяції програми «Proteus 7» відкрити програму (двічі клацнути лівою клавішею миші на її іконці ).

Далі слід побудувати електричну схему на основі мікросхем NT555, L297 та L298, LCD індикатора та мікроконтролера Arduino Uno, як показано на рисунках.

Для цього необхідно додати за допомогою закладки «DEVICES» на полі «Панелі інструментів», яка розташована вертикально зліва від робочого аркуша, потрібні елементи схеми (рис. 3):

- ◇ - Arduino Uno R3,
- ◇ - батарею живлення BATTERY, напруга якої може змінюватися в межах 5...46 В,
- ◇ - кнопки BUTTON,
- ◇ - мікросхеми L297 та L298,
- ◇ - резистори RES,
- ◇ - крокові двигуни MOTOR-STEPPER або MOTOR-BISTEPPER.
- ◇ - LCD індикатор LM016L
- ◇ - конденсатори CAPACITOR
- ◇ - осцилограф Oscilloscope.



Рис. 3. Функціональні елементи для побудови електричної схеми

Далі слід клацнувши лівою кнопкою миші на потрібному елементі, і курсором

встановити його на робочий аркуш та з'єднуємо з іншими. При необхідності, за наявності значної кількості елементів схеми, можна попередньо збільшити розмір робочого поля (знаходиться в середині синього прямокутника) наступним чином:

- ◇ натискаємо курсором на вкладку «System» і далі «Set Sheet Sizes» (рис. 4),
- ◇ у вікні, що відкриється, вибираємо формат робочого поля, клацнувши курсором справа від вибору.

Рис. 4

LCD індикатор 16x2 з контролером HD44780 у чотирибітному режимі роботи вмикаємо за схемою, зображеною на рис. 5.

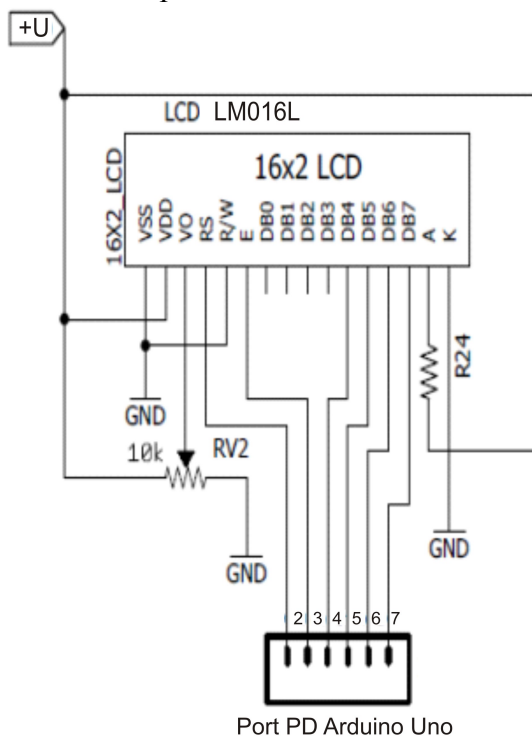


Рис. 5. Схема вмикання LCD індикатора

Потенціометром RV2, який приєднується на схемі до виводу Vo, встановлюється контрастність зображення на LCD індикаторі. Рівень напруги на виводі RS визначає, яка інформація буде надсилатися до індикатора (команди чи дані). Якщо RS=0, то надсилаються команди (встановити курсор на верхній чи нижній рядок, очистити). Якщо RS=1, то надсилаються дані або символи.

Вивід R/W вибирає режим читання або запису LCD індикатора. В режимі запису на індикатор надсилаються команди та дані, тому на схемі цей вивід заземлено.

Вивід E дозволяє записувати до регістрів індикатора дані з боку ліній D0...D7 мікроконтролера.

Виводи А (анод) і К (катод) використовуються для світлодіодного підсвічування

екрану індикатора через резистор R24. LCD індикатор може використовувати 4 або 8-бітовий режим передачі даних

Для виводу інформації на індикатор потрібно підключити виводи RS, E, DB0...DB7 до порту Arduino.

Для роботи з LCD індикатором буде використовуватись бібліотека LiquidCrystal, яка дозволяє Arduino працювати з різними LCD і яка входить до переліку бібліотек програмного середовища Arduino Ide. Бібліотека реалізує як 4-и, так і 8-бітний режим роботи індикатора.

Функція бібліотеки LiquidCrystal (rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7) визначає підключення та режим роботи індикатора, де rs, rw, enable – це номери ліній порту Arduino, які з'єднуються з відповідними виводами RS, RW, E LCD-індикатора, а також виводи Arduino d0, d1, d2, d3, d4, d5, d6, d7, які підключаються до відповідних виводів LCD-індикатора.

Якщо функція викликається у форматі LiquidCrystal (rs, rw, enable, d4, d5, d6, d7), то LCD буде працювати у 4-бітному режимі із лініями заданими параметрами d4, d5, d6 і d7.

Побудова робочої схеми на базі модуля Arduino Uno

На побудованій схемі (рис. 6) прямокутні імпульси з виходу генератора керованого напругою (U1) поступають на лінію PD6 порту PD мікроконтролера, а також на вхід CLOCK мікросхеми L297 (U3).

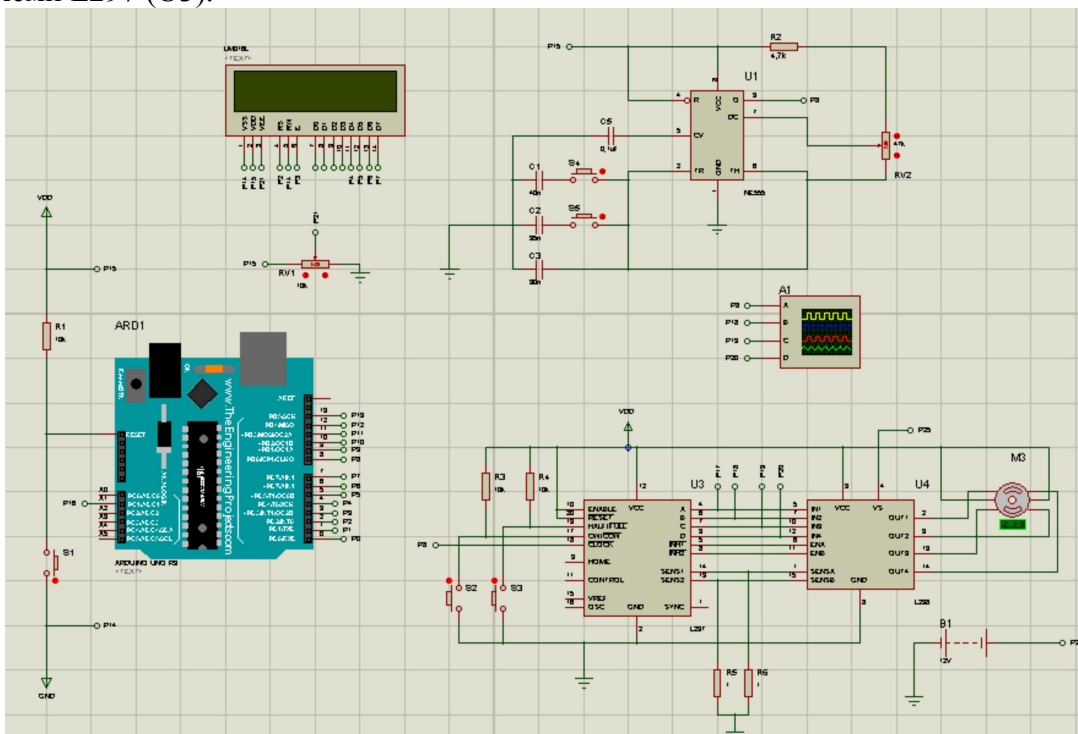


Рис. 6

Зауважимо, що виводи живлення А (анод) і К (катод) LCD індикатора на схемі не показані. Частота імпульсів визначається величиною резисторів R2 та RV2, а також конденсаторами C1, C2, C3. Кнопками S4 та S5 можна змінити діапазон регулювання частоти імпульсів. Частота в межах вибраного діапазону змінюється резистором RV2.

Кнопкою S2 вибирається напрям обертання крокового двигуна: кнопка не натиснута – двигун обертається за часовою стрілкою, натиснута – в протилежному напрямку. Кнопка S3 визначає режим роботи двигуна: кнопка не натиснута – напівкроковий режим роботи, натиснута – повнокроковий.

Створення скетчу

Для цієї схеми необхідно створити скетч за допомогою середовища Arduino Ide (Приклад 1).

Скетч повинен забезпечити вивід на LCD індикатор величини кількості обертів двигуна за 1 хвилину.

Далі слід обрати для крокового двигуна наступні параметри:

- ◇ для повнокрокового режиму приймається, що повне обертання ротора двигуна відбувається за 100 кроків,

◇ для напівкрокового режиму, відповідно, за вдвічі більшу кількість – за 200 кроків. Це означає, що для забезпечення швидкості обертання 200-крокового двигуна в 150 об/хв (2,5 об /сек) на вхід мікросхеми L297 потрібно подати імпульсний сигнал з періодом в 2 мілісекунди або 500 Гц. Відповідно, для 100 – крокового двигуна для забезпечення цієї ж швидкості обертання частота вхідних імпульсів має бути вдвічі меншою.

Далі слід обрати параметри генератора U1 так, щоб частота імпульсів була наближеною до 500 Гц (кнопки S4 та S5 не натиснуті, повзунок потенціометра RV2 знаходиться в середньому положенні).

Частота імпульсів 500 Гц відповідає кількості 200 кроків на повний оберт ротора, або 2,5 обертів за 1 сек чи 150 обертів за 1 хв.

Скетч для виводу на цифровий LCD дисплей інформації про швидкість обертання крокового двигуна за 1 хв буде таким:

```
#include <LiquidCrystal.h> // підключити бібліотеку
LiquidCrystal lcd(2, 3, 4, 5, 6, 7); // завдання ліній порту для
// роботи з LCD індикатором
int Htime; // тривалість імпульсу
int Ltime; // тривалість паузи
float Ttime; // тривалість періоду імпульсів
float speed; // оберти двигуна за хвилину
void setup ()
{
  pinMode(8, INPUT); // програмування лінії порту на ввід
  lcd.begin (16,2); //LCD дисплей на 16 символів, 2 рядки
}
void loop ()
{
  lcd.clear (); // очищення пам'яті дисплею
  lcd.setCursor (0,0); // встановлення курсора на верхній
// рядок дисплею
  lcd.print("speed"); // вивід на дисплей напису «speed»
  Htime=pulseIn(8,HIGH); // вимірювання тривалості імпульсу
  Ltime=pulseIn(8,LOW); // вимірювання тривалості паузи
  Ttime=Htime+Ltime; // визначення періоду імпульсів
// частота імпульсів (Гц) визначається
// як f=1000000/Ttime
  speed=1000000*60/200*Ttime; // визначення кількості обертів за хв
// за 200 кроків
  lcd.setCursor (0,1); // встановлення курсора на нижній
// рядок дисплею
  lcd.print(speed); // вивід швидкості обертання
  lcd.print("rpm"); // вивід на дисплей позначення
// кількості обертів за 1 хв
  delay(1000); // час індикації 1 сек
}
```

Програмування

Для роботи в програмному середовищі Proteus VSM, необхідно використовувати створений програмою Arduino Ide після компілювання скетчу файл з розширенням .hex.

Для цього потрібно запустити Arduino Ide і перед компілюванням скетчу зайти на вкладку «Файл» і далі «Налаштування». У вікні, що відкриється, поставити галочку навпроти опції «Показати детальний вивід: Компіляція».

Далі слід провести компіляцію скетчу, натиснувши курсором на вкладці «Перевірити» (рис. 7).




Рис. 7. Скетч після компілювання на Arduino Uno

Після компілювання скетчу необхідно відкрити вкладку «Скетч», і у вікні, що розкриється, натиснути лівою кнопкою миші на вкладці «Експорт бінарного файлу»; Далі знову натиснути на вкладці «Скетч» і у вікні, що з'явиться, натиснути на вкладку «Показати папку скетчу». У результаті розкриється вікно з папкою скетчу, в якому, крім робочого файлу скетчу, будуть ще два файли з розширенням .hex.

Слід скопіювати перший з них і записати його у вибрану папку проекту. Це і є файл, який буде потрібен для подальшої роботи з Proteus VSM.

Запуск програми

Запуск на виконання проекту на Proteus VSM здійснюється натисканням лівою клавішею миші кнопки «Play»  на панелі, яка розташована внизу зліва від робочого поля (аркуша). Після запуску проекту на виконання значок на кнопці змінить колір з чорного на зелений (рис. 8).

Для контролю за послідовністю імпульсів, які формує генератор імпульсів та контролер L297, в схемі до виходів мікросхем слід під'єднати чотириканальний осцилограф. Для запуску осцилографа під час виконання проекту, якщо він не відкриється самостійно, потрібно клацнути на його зображенні правою кнопкою миші і у вікні, що розкриється, клацнути лівою кнопкою на нижньому рядку «Digital Oscilloscope».

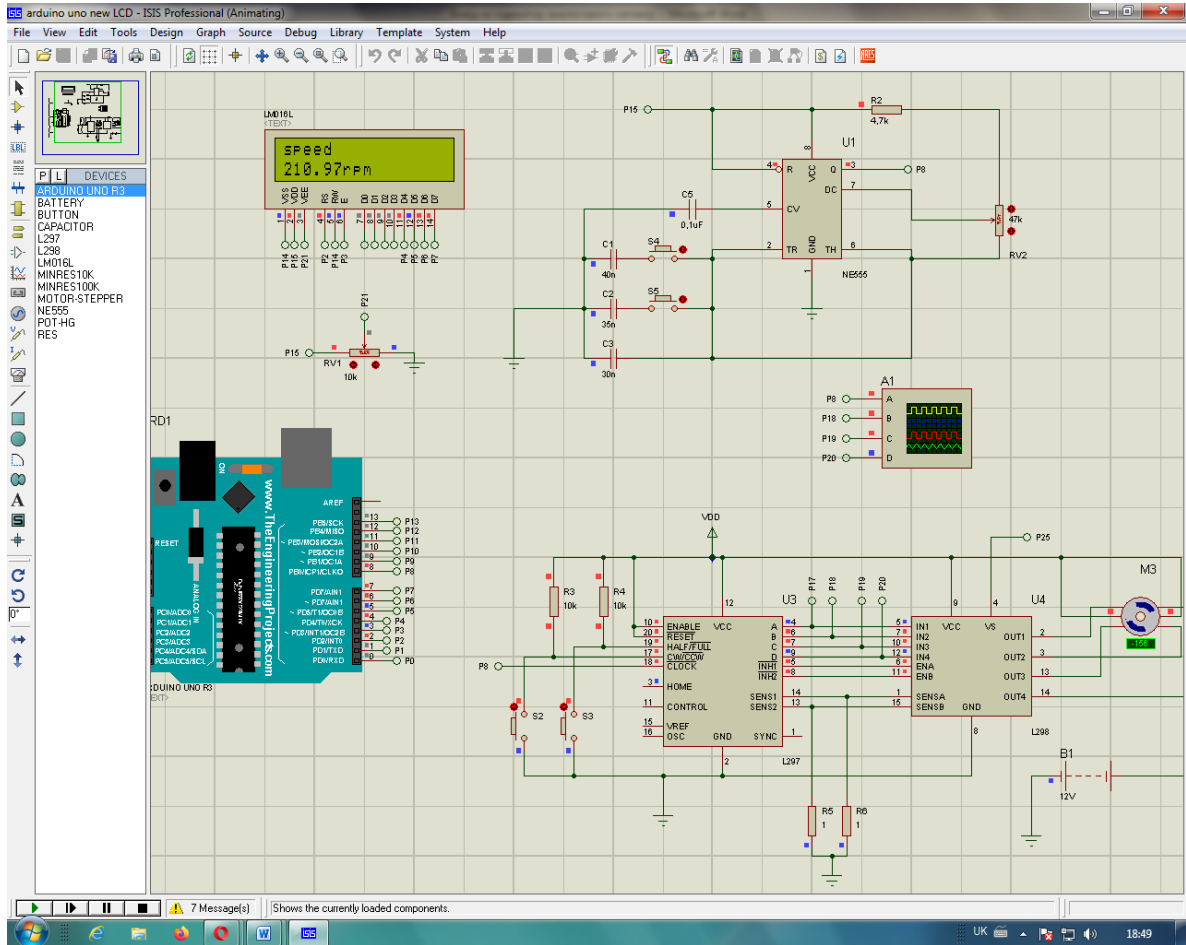


Рис. 8

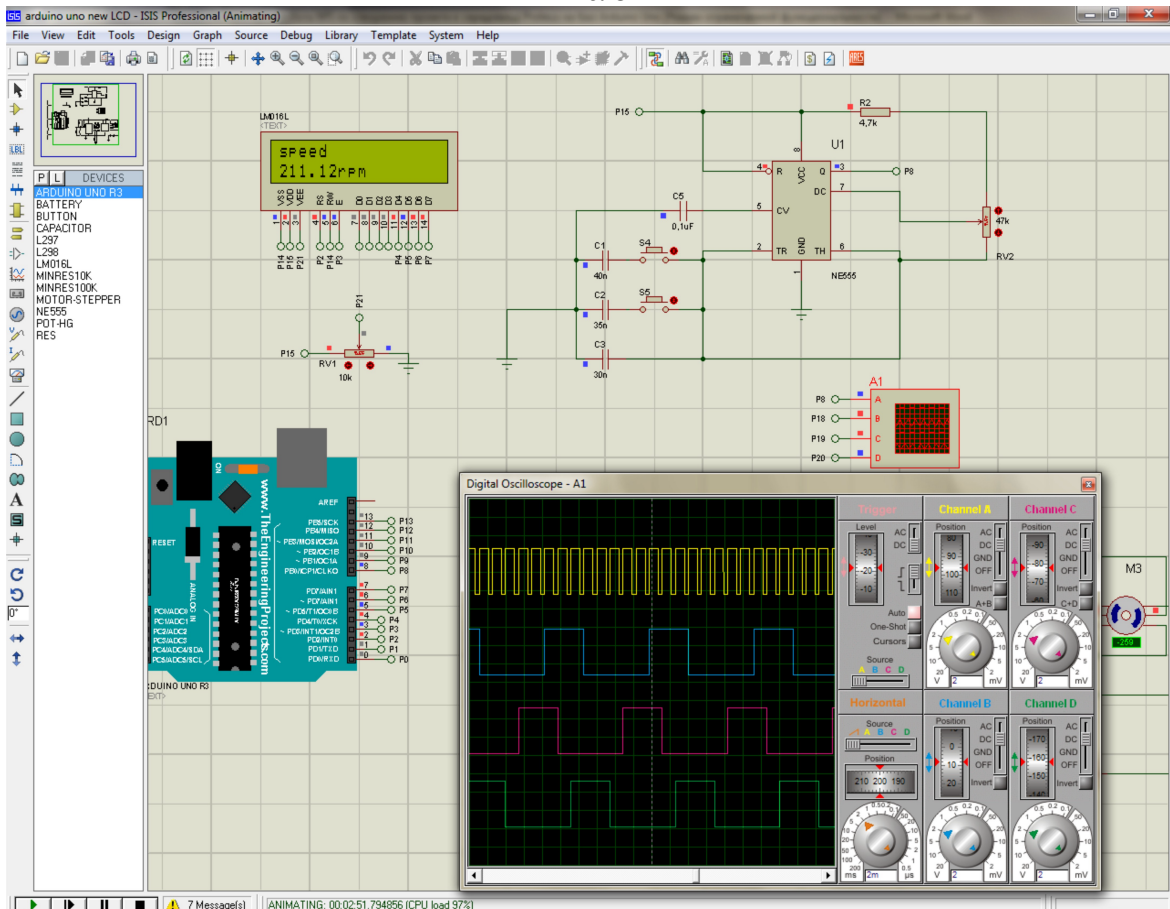


Рис. 9. Налаштування осцилографів під час виконання проекту

Далі слід звичайним чином налаштувати осцилограф, встановивши відповідним чином перемикачі на його панелі так, як це показано на рис. 9, обравши потрібні величини

розміру розгортки та масштабу амплітуди вхідного сигналу.

Виконання лабораторної роботи

Завдання №1

1. Побудувати схему електричну принципову відповідно до рис. 6 з використанням програмного середовища Proteus VSM. Зберегти робочий проект.

2. Створити скетч за Прикладом 1 на Arduino Ide і після його компілювання скопіювати файл з розширенням .hex в вибрану папку.

3. Запустити виконання проекту, використовуючи отриманий файл.

4. Зробити «Print Screen» екрану монітора з результатом виконання лабораторної роботи і додати його у звіт.

5. Зменшити частоту обертання двигуна в 2 рази, змінивши потрібні параметри в скетчі. Скомпілювати скетч, скопіювати файл з розширенням .hex і завантажити його в модуль Arduino Uno.

6. Зробити «Print Screen» екрану монітора з результатом виконання лабораторної роботи і додати його у звіт.

Завдання №2

1. Для створеного проекту (рис. 6) внести зміни у скетч з прикладу так, щоб на LCD індикатор виводилося значення частоти «frequency» генератора U1 та одиниці вимірювання «Hz».

2. Створити скетч на Arduino Ide і після його компілювання скопіювати файл з розширенням .hex в вибрану папку.

3. Запустити проект на виконання в Proteus VSM.

4. Натиснути кнопку S4 або S5 для зміни діапазону частоти генератора U1, а резистором RV1 змінити швидкість обертання двигуна.

5. Зробити «Print Screen» екрану монітора з результатом виконання лабораторної роботи і додати його у звіт.

Література

1. Мікропроцесорні системи управління: навч. посіб./ В.О.Денисюк, С.М.Цирульник; Вінн. нац. аграр. ун-т. Вінниця: ТВОРИ, 2021. 204.

2. Методичні вказівки до лабораторних робіт з дисципліни «Електротехніка та електроніка» для студентів спеціальності 122 Комп'ютерні науки і інформаційні технології» денної форм навчання / Укл.: А.В. Пархоменко, О.М. Гладкова. – Запоріжжя: ЗНТУ, 2016. – 41 с.

3. Розробка програмних модулів для обміну даними у промислових мережах: [Електронний ресурс] : навч. посіб. для студ. спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / Укладачі: А. В. Сагун, В. В. Хайдуров, І. А. Поліщук; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 16,2 МБайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 103 с.

4. L297.STEPPER MOTOR CONTROLLERS. Datasheet.\www.datasheetcatalog.com.

Зміст

Схема з використання таймера NE555	3
Створення проекту з використанням Arduino Uno	4
Побудова робочої схеми на базі модуля Arduino Uno	6
Створення скетчу	6
Програмування	7
Запуск програми	8
Виконання лабораторної роботи.....	10
Завдання №1	10
Завдання №2.....	10
Література	10
Зміст.....	11