

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/172.00.1/172.00.2/ 163.00.1/Б/ОК19-2020
	Екземпляр № 1	Арк 66/ 1

ЗАТВЕРДЖЕНО

Науково-методичною радою
Державного університету
«Житомирська політехніка»
протокол від 24 червня 2024 р.
№3

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ для проведення лабораторних занять з навчальної дисципліни «МІКРОКОНТРОЛЕРИ ТА МІКРОПРОЦЕСОРНА ТЕХНІКА»

для студентів освітнього ступеня «бакалавр»
спеціальностей 163 «Біомедична інженерія»,
172 «Телекомунікації та радіотехніка»
освітньо-професійні програми «Біомедична інженерія»,
«Телекомунікації та радіотехніка», «Інформаційні
відеосистеми та системи контролю доступу»

факультет інформаційно-комп'ютерних технологій
кафедра комп'ютерних технологій у медицині та телекомунікаціях

Розглянуто та рекомендовано на
засіданні кафедри комп'ютерних
технологій у медицині та
телекомунікаціях
21 червня 2024 р.,
протокол № 6

Розробники: ст. викладач КОРНІЮК Артур, д.т.н. проф. МАНОЙЛОВ
В'ячеслав

Житомир
2024

Корніюк А.В., Манойлов В.П. Методичні рекомендації для проведення лабораторних занять з навчальної дисципліни «Мікроконтролери та мікропроцесорна техніка» для студентів освітнього ступеня «бакалавр» спеціальностей 163 «Біомедична інженерія», 172 «Електронні комунікації та радіотехніка» освітньо-професійні програми «Біомедична інженерія», «Телекомунікації та радіотехніка», «Інформаційні відеосистеми та системи контролю доступу» / А. В. Корніюк, В.П.Манойлов – Житомир: Житомирська політехніка, 2024. – 66 с.

ЗМІСТ

Вступ.....	3
Лабораторна робота №1. GPIO, порти вводу/виводу Arduino.....	4
Лабораторна робота №2. АЦП. Створення вольтметра та термометра на базі Arduino.....	8
Лабораторна робота №3. ЦАП. Використання ШІМ для ЦАП. ЦАП на матриці R2R	14
Лабораторна робота №4. UART. Універсальний послідовний інтерфейс передачі даних.....	27
Лабораторна робота №5. Цифро-буквені індикатори	36
Лабораторна робота №6. Драйвери для 7-ми сегментних індикаторів.....	42
Лабораторна робота №7. Регістри.....	49
Лабораторна робота №8. Створення простого ПЛК на базі Arduino.....	55

Лабораторна робота №1

GPIO, ПОРТИ ВВОДУ/ВИВОДУ ARDUINO

Мета роботи:

1. Практичне ознайомлення портами вводу/виводу плат Arduino.
2. Створення на базі Arduino світлодіодного маячка використанням портів та програмної затримки мікроконтролера.
3. Створення блоку вводу на Arduino за допомогою тактової кнопки.

1 Короткі теоретичні відомості

Мікроконтролер ATmega328, який використовується на платі Arduino UNO, має набір GPIO (General-Purpose Input/Output) портів, які дозволяють зчитувати вхідні сигнали, керувати виводами та взаємодіяти з зовнішніми пристроями та сенсорами. Ось деякі теоретичні відомості про GPIO порти на ATmega328.

Загальна інформація: ATmega328 має 23 GPIO порти, які розташовані на мікроконтролері. Вони позначаються як PORTB, PORTC, PORTD для виводів і PINB, PINC, PIND для входів.

1. PORTB:

- PORTB є одним із трьох портів виведення (вихідних портів).
- Має 8 виводів, позначених як PB0, PB1, ..., PB7.
- Використовується для керування зовнішніми пристроями шляхом встановлення або скидання логічних рівнів на виводах.
- Деякі виводи (наприклад, PB5) також мають спеціальні функціональні можливості, такі як PWM.

2. PORTC:

- PORTC є ще одним портом виведення.
- Має 7 виводів, позначених як PC0, PC1, ..., PC6.
- Зазвичай використовується для підключення аналогових компонентів або використовується як вихід для керування зовнішніми пристроями.

3. PORTD:

- PORTD також є портом виведення.
- Має 8 виводів, позначених як PD0, PD1, ..., PD7.
- Зазвичай використовується для керування зовнішніми пристроями і сенсорами.

4. PINB:

- PINB - це один із трьох портів вводу (вхідних портів).
- Використовується для зчитування стану вхідних сигналів на виводах

PORTB.

- Дозволяє читати логічні рівні на виводах PB0, PB1, ..., PB7.

5. PINC:

- PINC - це ще один порт вводу.

- Використовується для зчитування стану входних сигналів на виводах PORTC.

- Дозволяє читати логічні рівні на виводах PC0, PC1, ..., PC6.

6. PIND:

- PIND - це порт вводу.

- Використовується для зчитування стану входних сигналів на виводах PORTD.

- Дозволяє читати логічні рівні на виводах PD0, PD1, ..., PD7.

Ці порти дозволяють мікроконтролеру взаємодіяти з зовнішніми пристроями, зчитувати дані та керувати станами виводів у вашій програмі на Arduino UNO.

Розподілення портів: Кожен із зазначених портів має від 6 до 8 виводів, які можуть використовуватися для введення або виведення даних. Наприклад, PORTB має 8 виводів (від 0 до 7), PORTC має 7 виводів (від 0 до 6), а PORTD має 8 виводів (від 0 до 7).

Робочий напруга та струм: Мікроконтролер працює при напрузі живлення 5 вольт (Vcc). Виходи мають здатність надавати або приймати струм до 40 мА на кожний вивід.

Режими роботи: GPIO порти можуть працювати в різних режимах, таких як вхід, вихід або в ролі спеціальних функціональних виводів, таких як PWM (Pulse Width Modulation) або інші спеціальні функції.

Введення (Input): Для зчитування стану входного сигналу на GPIO порті, ви можете використовувати реєстри PINx (де x - буква порту, наприклад, PINB, PINC, PIND).

Виведення (Output): Для встановлення стану GPIO виводу ви можете використовувати реєстри PORTx (де x - буква порту, наприклад, PORTB, PORTC, PORTD).

Змінення режиму вводу/виводу (Data Direction): Для налаштування GPIO порту як входу або виходу використовується реєстр DDRx (де x – буква порту, наприклад, DDRB, DDRC, DDRD).

2 Лабораторна установка

Схема 1

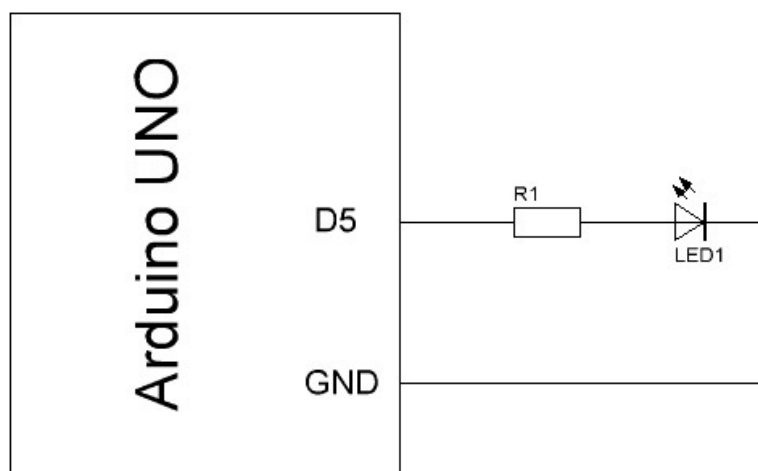
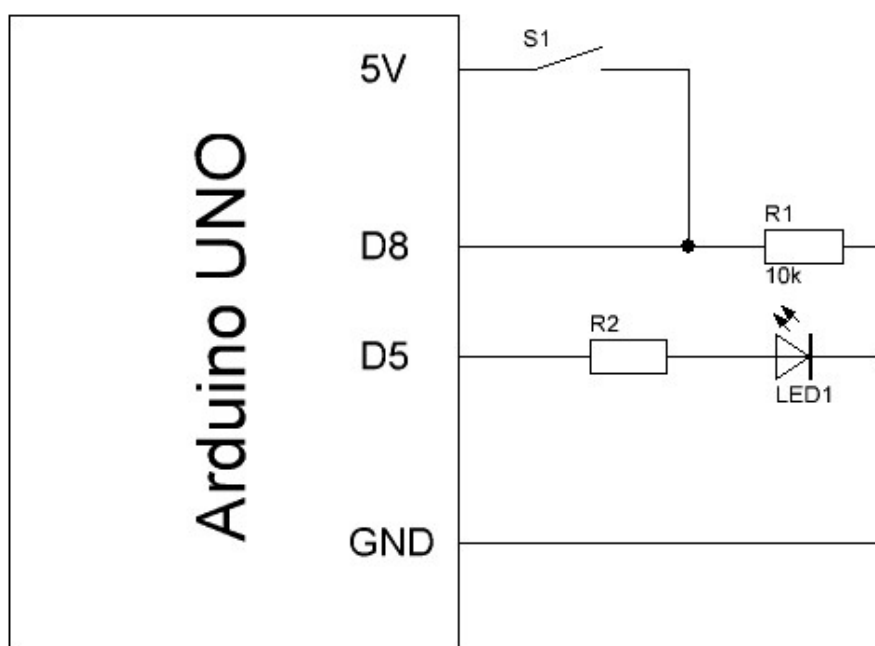


Схема 2



3 Хід виконання роботи

- 3.1 Ознайомтесь з лабораторною установкою та додатковими вказівками по роботі з приладами. Ознайомтесь з технічною документацією (даташит) на обраний вами світлодіод.
- 3.2 Зберіть першу схему, та перевірте правильність. Розрахувати струмообмежуючий резистор (по варіантах 1-червоний, 2 –зелений, 3 –жовтий, 4 –синій, 5 – ультрафіолет, 6 - інфрачервоний) за формулою

$$R = \frac{U_{\text{живл.}} - U_{\text{св}}}{I}$$

- 3.3 Напишіть програму в Arduino IDE, яка буде виконувати блимання світлодіодом відповідно до варіанту (Світиться (номер в журналі * 100 мс), не світиться ІВ- 1 с, БІ – 0,5с, ТР- 0,2с).
- 3.4 Зберіть другу схему, та перевірте правильність.
- 3.5 Напишіть програму в Arduino IDE, яка буде вмикати та вимикати світлодіод після натискання на конпку (раз натиснули світлодіод засвітився, наступний раз натиснули світлодіод погас, і т.д.)

4 Розрахункове завдання

Розрахункове завдання – див. п. 3.2.

5 Вимоги до звіту

Звіт з лабораторної роботи повинен містити:

1. Коротке описання мети і методики проведення роботи.
2. Перелік використаних приладів та матеріалів.
3. Таблиці результатів вимірювань, графічне оформлення.
4. Програмний код із середовища Arduino IDE з коментарями.
5. Розрахункове завдання.
6. Висновки.

6 Контрольні питання

1. Що таке порти вводу/виводу.
2. Що таке струмообмежуючий резистор, для чого потрібний.
3. Що таке підтягуючий резистор, призначення.
4. Що таке брязкіт контактів, як боротися.
5. Варіанти підключення кнопок..

Лабораторна робота №2
**АЦП. СТВОРЕННЯ ВОЛЬТМЕТРА ТА ТЕРМОМЕТРА НА БАЗІ
ARDUINO**

Мета роботи:

4. Практичне ознайомлення з АЦП.
5. Створення на базі Arduino вольтметра з використанням АЦП мікроконтролера.
6. Створення на базі Arduino термометра з використанням спеціалізованої мікросхеми LM35 та АЦП мікроконтролера.

1 Короткі теоретичні відомості

Аналого-цифровий перетворювач являє собою пристрій для перетворення величини, що змінюється у часі, у відповідні значення числових кадрів. Перетворення здійснюється через певний інтервал часу t_d , який визначається необхідною частотою дискретизації

$$f_d = \frac{1}{t_d}$$

вхідного аналогового сигналу у конкретній цифровій системі.

Кількісний зв'язок між аналоговою величиною y і відповідною їй цифровою величиною z , яка характеризує алгоритм квантування, має вигляд: $z=(u - \delta)/\Delta u$, де Δu - відлік квантування є аналоговий елемент одиниці молодшого розряду коду, δ - похибка перетворення на даному кроці.

Основні параметри АЦП можна розділити на дві групи: характеризуючу статичну та динамічну точність.

Параметри, які характеризують статичну точність, визначають роботу АЦП при дискретизації квазіпостійних фізичних величин. До цих параметрів відносяться:

- похибка квантування;
- інструментальна похибка;
- часова нестабільність - це здібність АЦП зберігати статичну точність на протязі відповідних інтервалів часу;

- роздільна здатність - це здібність АЦП розрізняти два значення вхідного сигналу, характеризує потенційні можливості АЦП з точки зору досяжної точності;

- діапазони вимірювальних величин - максимальні та мінімальні для даного АЦП значення вимірювальної величини;
- вхідний опір - характеризує міру впливу входу АЦП на вимірювальну величину.

Якщо вхідний опір не великий та сумісний з опором джерела сигналу він не є постійним, то це може призвести до появи додаткових похибок. Тому до величини вхідного опору пред'являють жорсткі вимоги до постійної величини.

Поява динамічних похибок пов'язана з дискретизацією сигналів, які змінюються з часом. До параметрів, що характеризують динамічну точність відносяться: частота дискретизації, час перетворення, час вибірки (стробування).

Частота дискретизації - це частота $f_d = \frac{1}{t_d}$, з якою здійснюється утворення дискретних значень сигналу.

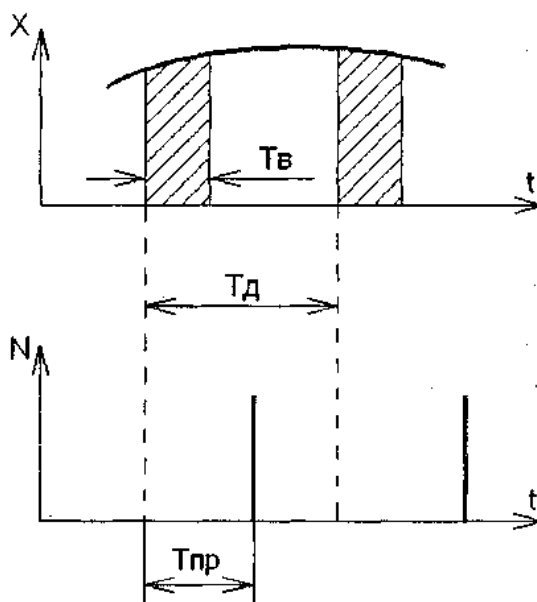


Рисунок 1 – Час вибірки так частота дискретизації

Час перетворення $T_{пр}$ - час, який відраховує від початку перетворення до появи на виході коду, відповідного даній вибірці.

Час вибірки $T_в$ (стробування) - час, на протязі якого здійснюється поява одного вибраного значення, Рис. 1.

Дискретизація неперервних сигналів. Суть процесу дискретизації в тому, що з неперервного у часі сигналу обираються окремі його значення, відповідні моментам часу, які з'являються через визначений часовий інтервал T . Інтервал T

називається тактовим інтервалом часу, а моменти часу t_1, t_2, \dots , в які беруться відліки, - тактовими моментами часу.

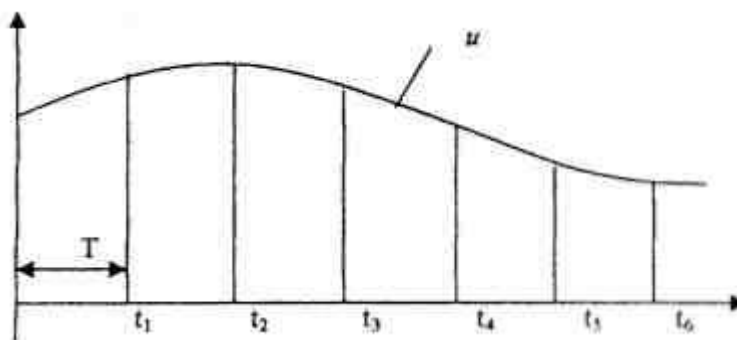


Рисунок 2 – Відліки та тактовий інтервал

Дискретні значення сигналу потрібно визначати з таким малим тактовим інтервалом T , щоб по ним можна було б встановити сигнал в аналоговій формі з потрібною точністю. (Рис.2)

Квантування і кодування. Сутність цих операцій заключається в наступному. Складається сітка так званих рівнів квантування, зсунутих один відносно іншого на величину n , яка називається кроком квантування. Кожному рівню квантування можна дати порядковий номер (0, 1, 2, 3 і т.д.). Далі, отримані в результаті дискретизації значення початкової аналогової напруги замінюються найближчими до них рівнями квантування. Так, на діаграмі значення напруги в момент t_0 замінюється найближчим до неї рівнем квантування з номером 3, в тактовий момент t_1 значення напруги ближче до рівня 6 і замінюється цим рівнем і т.д.

Описаний процес носить назву операції квантування, зміст якого складається в округленні значень аналогової напруги, обраних в тактові моменти часу. Як і будь-яке округлення, процес квантування призводить до похибки (до помилок квантування) в представленні дискретних значень напруги, створюючи так званий шум квантування. При проектуванні АЦП намагаються понизити шум квантування до такого рівня, при якому він ще забезпечує потрібну точність представлення сигналу.

Наступна операція, яка виконується при аналого-цифровому перетворенні сигналів - кодування. Зміст її складається в наступному. Округлення значення напруги, яке виконується при операції квантуванні, дозволяє ці значення представляти числами (номерами відповідних рівнів квантування). Для діаграми

на рис.3. створюється послідовність чисел: 3, 6, 7, 4, 1, 2 і т.д. Далі, отримана таким шляхом послідовність чисел представляється двійковим кодом.

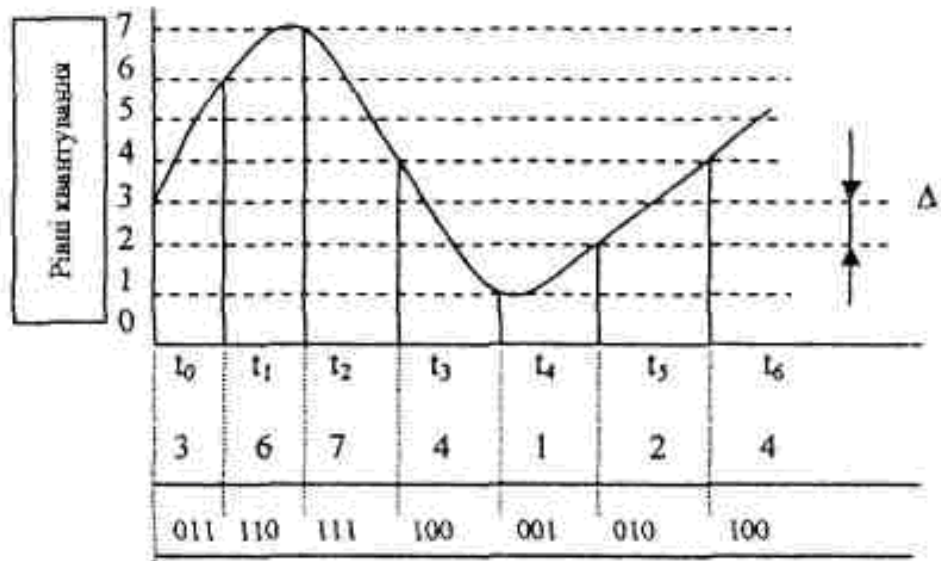


Рисунок 3 – Діаграма кодування

2 Лабораторна установка

Схема 1

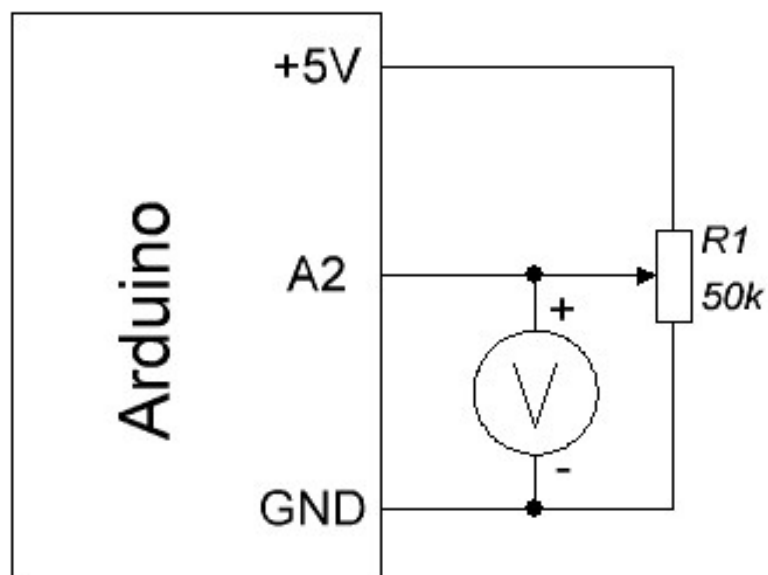
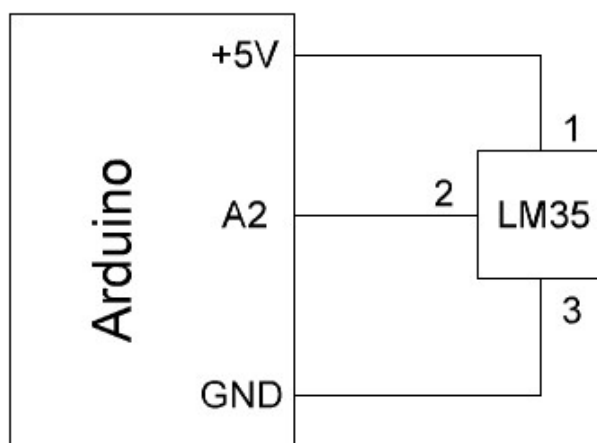


Схема 2



3 Хід виконання роботи

- 3.1 Ознайомтесь з лабораторною установкою та додатковими вказівками по роботі з приладами. Ознайомтесь з технічною документацією (даташит) на мікросхему LM35.
- 3.2 Зберіть першу схему, та перевірте правильність. Розрахуйте напругу одного відліку за формулою $U_{\text{відл}} = \frac{U_{\text{опорна}}}{2^{\text{Розрядність АЦП}}}$.
- 3.3 Напишіть програму в Arduino IDE, яка буде виводити в монітор послідовного порту кількість отриманих відліків та напругу у вольтах, так як зображено нижче.

```
Voltage in V:4,213
```

```
863
```

```
Voltage in V:4,213
```

```
863
```

```
Voltage in V:4,213
```

- 3.4 Заповніть таблицю, виконавши 5-10 вимірювань, кожного разу змінюючи положення повзунка змінного резистора.

№ п/п	Кількість відліків	Напруга на вольтметрі, В	Розрахункова напруга, В

- 3.5 Зберіть другу схему, та перевірте правильність. Знайдіть в технічній документації на мікросхему LM35 співвідношення напруги на виході мікросхеми до температури, виведіть формулу.

3.6 Напишіть програму в Arduino IDE, яка буде виводити в монітор послідовного порту температуру в градусах Цельсія, та Фаренгейта. Виводити дані так як зображено нижче.

Temperature in F: 492.8

Temperature in C: 25.6

Temperature in F: 492.8

Temperature in C: 25.6

4 Розрахункове завдання

Розрахункове завдання – див. п. 3.2 та п. 3.5.

5 Вимоги до звіту

Звіт з лабораторної роботи повинен містити:

- 1 Коротке описання мети і методики проведення роботи.
- 2 Перелік використаних приладів та матеріалів.
- 3 Таблиці результатів вимірювань, графічне оформлення.
 1. табличних даних по п.3.4.
 - 4 Результати вимірювань по пп. 3.4, 3.6.
 - 5 Програмний код із середовища Arduino IDE з коментарями.
 - 6 Розрахункове завдання.
 - 7 Висновки.

6 Контрольні питання

1. Що означає термін АЦП?
2. Які бувають види АЦП?
3. Що означає термін квантування?
4. Теорема Котельнікова-Найквіста?
5. Що таке частота дискретизації?
6. Що означає поняття опорна напруга?
7. Де використовуються АЦП?
8. Що таке квантування?
9. Чи залежить полоса пропускання від частоти дискретизації, чому?
10. Який тип АЦП та його розрядність в Arduino UNO?

Лабораторна робота №3

ЦАП. ВИКОРИСТАННЯ ШІМ ДЛЯ ЦАП. ЦАП НА МАТРИЦІ R2R

Мета роботи:

1. Практичне ознайомлення з ЦАП.
2. Цифро-аналогове перетворення на базі Arduino з використанням ШІМ.
3. Цифро-аналогове перетворення на базі Arduino з використанням R2R матриці.

1 Короткі теоретичні відомості

1.1 Принцип ЦА-перетворення. Параметри ЦАП

Основне призначення ЦАП – автоматичне перетворення (декодування) двійкових кодів на еквівалентні їм значення будь-якої фізичної величини (напруги або струму). Кількісний зв'язок між вхідним числовим, переважно двійковим, кодом N_2 і його аналоговим еквівалентом, наприклад напругою $U_{ВИХ}$ для довільного моменту часу t_1 визначається за співвідношенням:

$$U_{ВИХ.i} = \Delta U N_{2i} \pm \delta U_i, i \in \{0, 1, \dots, n-1\}, \quad (1.1)$$

де ΔU – крок квантування за рівнем напруги, тобто “вага” одного дискрету напруги, якій відповідає один двійковий розряд;

$$N_2 = 2^0 X_0 + 2^1 X_1 + \dots + 2^{n-1} X_{n-1} = \sum_{i=0}^{n-1} 2^i X_i;$$

$X_i \in X \{X_0, X_1, \dots, X_{n-1}\}$ і набуває значення 0 або 1; δU_i – похибка перетворення.

З цих виразів легко визначити вагу ΔU одного дискрету вихідної напруги $U_{ВИХ.i}$, який відповідає одиниця молодшого цифрового розряду. Якщо відомі мінімальне U_{\min} і максимальне U_{\max} значення вихідної напруги $U_{ВИХ.i}$, яка для різного типу n -розрядного ЦАП може мати різну полярність, то:

$$\Delta U = \frac{|U_{\min}| + |U_{\max}| \mp \delta U_i}{2^{n-1}}. \quad (1.2)$$

Для цього випадку цифровий код на вході ЦАП буде визначатись як:

$$N_2 = \text{int} \left[\frac{U_i \mp \delta U_i}{\Delta U} + 2^{n-1} \frac{|U_{\min}|}{|U_{\min}| + |U_{\max}|} \right]. \quad (1.3)$$

Основними параметрами ЦАП є динамічний діапазон зміни вхідних та вихідних значень, швидкодія і похибка перетворення.

Динамічні діапазони за входом (U_{\max}/U_{\min}) і виходом ($N_{2\max}/N_{2\min}$) при лінійному ЦА-перетворенні мають одне і те саме значення, яке виражають або числом розрядів цифрового коду, або в децибелах, при: $U_{\min} = 0, N_{\min} = 0$ динамічний діапазон:

$$D = U_{\max}/\delta U = N_{\max}/\delta N, \quad (1.4)$$

де δU і δN – допустимі абсолютні похибки відхилення значень статичної характеристики $U_{\text{вих}} = f(N_2)$ від лінійного закону.

Швидкодію ЦА-перетворення визначають такі часові параметри:

час перетворення – інтервал часу, протягом якого відбувається акт однозначного перетворення код-аналог;

частота квантування – величина, що обернена періоду квантування, тобто інтервалу часу між сусідніми послідовними перетвореннями.

Статична характеристика трирозрядного ЦАП з використанням прямого коду, що функціонує за наведеною таблицею істинності (табл.1), показана на рис.1.1.

Схеми ЦАП виготовляють здебільшого в інтегральному виконанні, причому з метою використання їх у мікропроцесорній техніці.

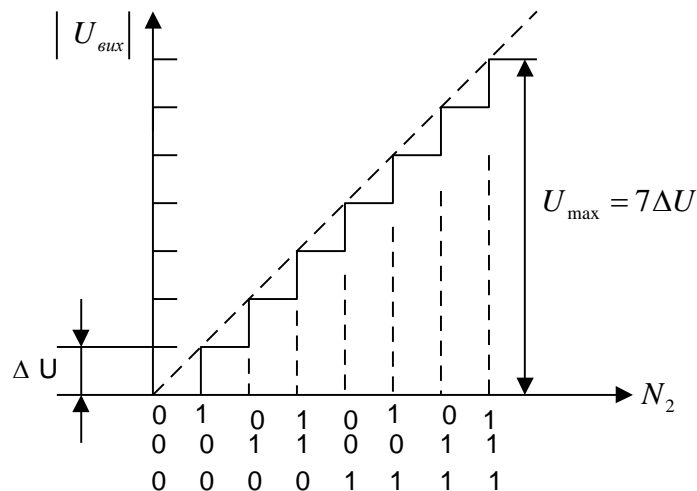


Рисунок 1.1 – Статична характеристика трирозрядного ЦАП.

Серед рівних способів ЦА-перетворення широке використання знайшли ЦАП, спільною ознакою яких в наявність матриці резисторів R з вихідним (аналоговим) суматором на операційному підсилювачі ОП (рис.1.2). Матриця резисторів призначена для “зваження” цифрового сигналу, який подано паралельним кодом залежно від його двійкового розряду. Через те що при різних способах з’єднання резисторів такі схеми нагадують сходинки, матрицю іноді називають багатоланковою резистивною схемою сходинкового типу. Для реалізації ЦА-перетворення переважно використовують два типи матриці резисторів:

- 1) складеної з двійково-зважених резисторів,
- 2) постійного імпедансу типу $R - 2R$.



Рисунок 1.2 – Спрощена структура ЦАП.

Таблиця 1

X_2	X_1	X_0	$ U_{вих} $
0	0	0	0
0	0	1	1 ΔU
0	1	0	2 ΔU
0	1	1	3 ΔU
1	0	0	4 ΔU
1	0	1	5 ΔU
1	1	0	6 ΔU
1	1	1	7 ΔU

} CP
} MP

1.2 ЦАП з широтно-імпульсною модуляцією

Дуже часто ЦАП входить у склад мікропроцесорних систем. В такому випадку, якщо не потрібна висока швидкодія, цифро-аналогове перетворення може бути дуже просто здійснене за допомогою широтно-імпульсної модуляції (ШІМ). Схема ЦАП з ШІМ наведена на рис.1.3.

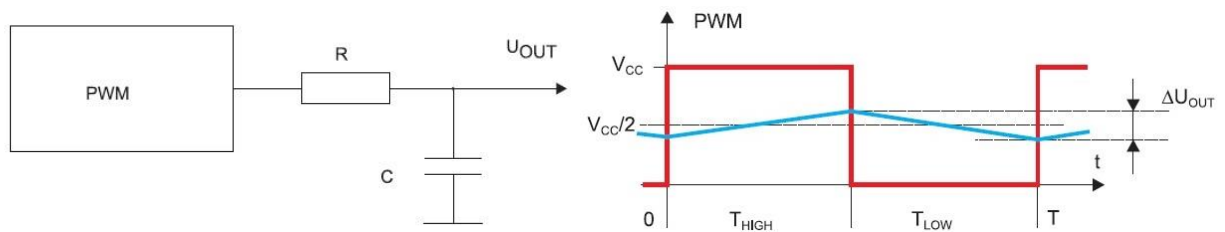


Рисунок 1.3 – Схема ЦАП і сигнал на вході та виході фільтра

Найпростіше організується цифро-аналогове перетворення в тому випадку, якщо мікроконтролер має вбудовану функцію широтно-імпульсного перетворення (на платі Arduino є вбудований ШІМ модулятор). У залежності від заданої розрядності перетворення (для контролера Arduino режим 8 біт) контролер з допомогою свого таймера/лічильника формує послідовність імпульсів, відносна тривалість яких $\gamma = t_{\text{та}} / T$ визначається співвідношенням

$$\gamma = \frac{D}{2^N} \quad (1.5)$$

де N - розрядність перетворення, а D - перетворюваний код. Фільтр нижніх частот згладжує імпульси, виділяючи середнє значення напруги. У результаті вихідна напруга перетворювача

$$U = \gamma U = \frac{DU}{2^N} \quad (1.6)$$

Розглянута схема забезпечує майже ідеальну лінійність перетворення, не містить прецизійних елементів (за винятком джерела опорної напруги). Основний її недолік - низька швидкодія.

1.3. ЦАП на двійково-зважених резисторах.

Це найпростіший щодо будови принципової схеми ЦАП, у якого матриця резисторів складена за принципом адекватного відтворення двійкового коду: номінал кожного резистора – аналог двійкового коду з ряду

$$2^0 R, 2^1 R, \dots, 2^{n-2} R, 2^{n-1} R.$$

Крім матриці двійково-зважених резисторів до схеми даного ЦАП (рис.1.4) входять: суматор на ОП, стабілізоване джерело опорної напруги E_0 і аналоговий мультиплексор, що являє собою комутатор на ключах, які керовані вхідним цифровим (двійковим) кодом $N_2 = \{X_{n-1} X_{n-2} \dots X_1 X_0\}$. Матриця резисторів живиться напругою E_0 і залежно від положень ключів, тобто від поданого коду N_2 на вхід аналогового мультиплексора, перетворює N_2 на постійний струм. При нульовому коді $N_{2\min} = \{00\dots 0\}$ (ключі у лівому положенні) у точці А струм мінімальний і на виході ОП, що грає роль суматора зважених струмів $I_n, U_{\text{вих}} = 0$. Якщо на вході ЦАП код $N_{2\max} = \{11\dots 1\}$, всі резистори матриці запаралелені (ключі у правому положенні) і у точці А струм I_n максимальний, що відповідає $U_{\text{вих}} = U_{\max}$.

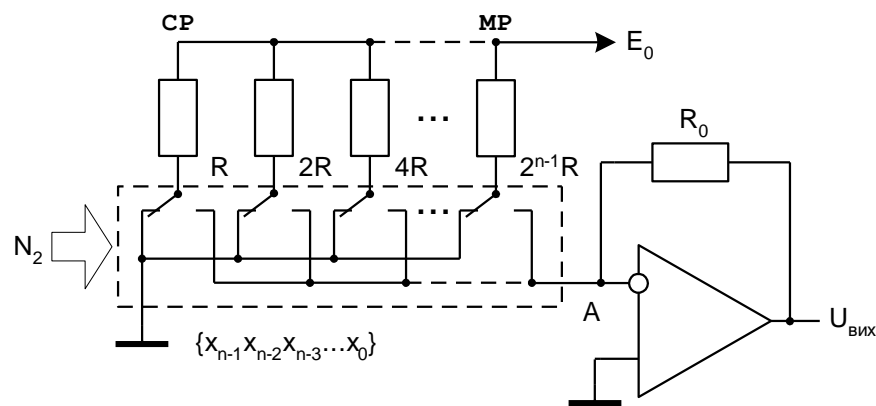


Рисунок 1.4 – ЦАП на двійково-зважених резисторах

Таким чином на виході n -розрядного ЦАП залежно від вхідного коду N_2 при $E_0 = \text{const}$ створюється напруга:

$$U_{BUX} = -R_0 I_n = -\frac{R_0}{R_M} E_0, \quad (1.7)$$

де

$$\begin{aligned} \frac{1}{R_M} &= \frac{X_{n-1}}{2^0 R} + \frac{X_{n-2}}{2^1 R} + \dots + \frac{X_1}{2^{n-2} R} + \frac{X_0}{2^{n-1} R} = \\ &= \frac{1}{2^{n-1} R} (2^{n-1} X_{n-1} + 2^{n-2} X_{n-2} + \dots + 2^1 X_1 + 2^0 X_0) = \frac{N_2}{2^{n-1} R} \end{aligned}$$

є еквівалентна провідність матриці резисторів, значення якої залежить від комбінації вхідних змінних X_i .

Значення одного дискрету напруги даного ЦАП:

$$\Delta U = \left| \frac{U_{BUX}}{N_2} \right| = \left| \frac{R_0}{2^{n-1} R} E_0 \right| = R_0 \Delta I, \quad (1.8)$$

де ΔI – значення одного дискрету струму, який визначається найбільшим опором матриці резисторів $2^{n-1} R$, тобто молодшим розрядом (МР) ЦАП.

Якщо $R = R_0$, то $\Delta U = E_0 / 2^{n-1}$, а $E_0 = U_{\max}$, що відповідає (7.2) при $U_{\min} = 0$ і $\delta U_i = 0$.

Отже, “вага” одного розряду ЦАП, яка у кінцевому, результаті визначає точність ЦА-перетворення, залежить від розрядності n ЦАП і коефіцієнта K_{UCC} підсилення ОП старшого розряду (СР) $X_0 = 1 (K_{UCC} = R_0 / R)$ при заданому $E_0 = const$. Максимальна вихідна напруга ЦАП, при якій $N_{2\max} = \{1 \dots 1\}$, буде:

$$U_{\max} = \left| \frac{(2^n - 1) R_0}{2^{n-1} R} E_0 \right|. \quad (1.9)$$

Незважаючи на простоту технічної реалізації схема ЦАП на двійково-зважених резисторах має такі недоліки: по-перше, необхідність точного підбору широкого діапазону резисторів різних номіналів, до того ж з розкидом номіналів опорів, що менше за значення найменшого з них опору R , по-друге, залежність опорів від температури, що обмежує точність ЦА-перетворення; по-третє, підвищені вимоги до джерела стабілізованої напруги E_0 , яке має працювати у широкому динамічному діапазоні зміни навантаження R_M .

1.4. ЦАП на основі матриці резисторів $R-2R$

Принцип роботи цього ЦАП не відрізняється від принципу роботи вже розглянутого ЦАП. Функціонування ЦАП на матриці резисторів $R-2R$. Його відмінність полягає лише у схемі матриці резисторів, яка тут також виконує функцію забезпечення вагового множника на двійкових входах ЦАП. Перевага матриці типу $R-2R$ – у простоті її виготовлення, бо для неї досить мати лише два номінали резисторів R і $2R$ замість їх широкого діапазону. Даний тип ЦАП через таку властивість є економічно вигіднішим і тому його матрицю резисторів разом з комутатором на КМОН-ключах виготовляють в інтегральному виконанні.

На рис.1.5 зображено схему n -розрядного ЦАП з матрицею резисторів $R-2R$, в якій показано можливість застосування ЛЕ 2І як комутаторів рівнів (0 або 1) на розрядових лініях матриці залежно від комбінації вхідного коду N_2 . Тут ОП виконує функцію суматора-підсилювача зважених на резистивному багатоступеневому подільнику та просумованих у точці B струмів.

Розглянемо роботу ЦАП при дозволяючому рівні $E_1 = 1$ на об'єднаних входах ЛЕ, на інші входи яких подається двійковий код: $N_2 = \{X_{n-1} X_{n-2} X_{n-3} \dots X_0\}$. У даній схемі опорною напругою E_0 – якою живиться матриця $R-2R$, в напруга на виході (одного або більше залежно від N_2), ЛЕ при $X_i = 1$. Отже, на потенційних виводах резисторів $2R$, що приєднані до ЛЕ, відповідно до коду N_2 буде $E_0 = 1$ при $X_i = 1$ або $E_0 = 0$ при всіх $X_i = 0$.

При $N_{2\min} = \{000\dots 0\}$ для всіх резисторів $2R$ з'являється спільна нульова шина (у тому числі і у резистора $2R$, що знаходиться між точками A і B , бо точка B має віртуальний нуль ОП). Отже, у точці A відносно нульового потенціалу еквівалентний спір матриці R_M дорівнює R , і оскільки напруга на вході ОП відсутня (нульова), $U_{\text{вих}} = 0$.

На рис.7.5 розглянуто еквівалентні схеми матриці резисторів $R-2R$ для чотирьох різних випадків вхідного коду:

- а) для $N_2 = \{100\dots 0\}$,
- б) для $N_2 = \{010\dots 0\}$,
- в) для $N_2 = \{001\dots 0\}$,
- г) для $N_2 = \{000\dots 1\}$.

При появі одиниці лише у старшому розряді, тобто при $X_{n-1} = 1$ (рис.1.6,а) коефіцієнт передачі такої схеми подільника у точці А дорівнює $1/3$. При $X_{n-2} = 1$ (рис.1.6,б) отримуємо половину коефіцієнта передачі попередньої схеми, тобто $1/(3 \cdot 2) = 1/6$, а при $X_{n-3} = 1$ (рис.1.6,в) – половину від останнього, тобто $1/(3 \cdot 2 \cdot 2) = 1/12$ тощо. У разі появи молодшого розряду (МР) $N_2 = \{000\dots 1\}$ на виході ЦАП з'явиться напруга ΔU , що відповідає елементарному кванту.

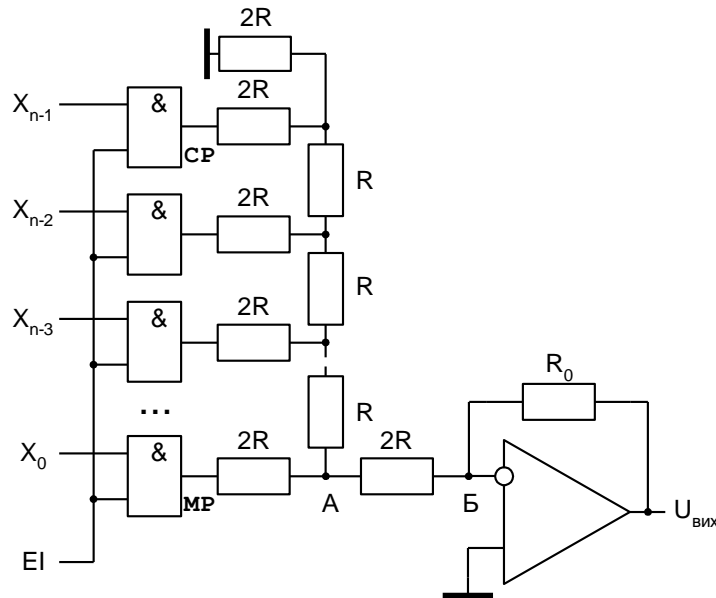
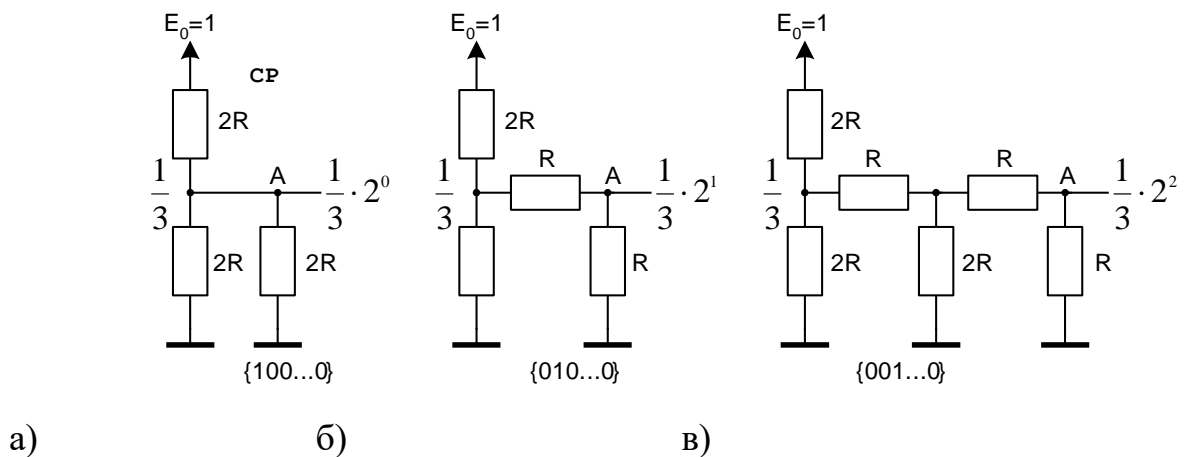
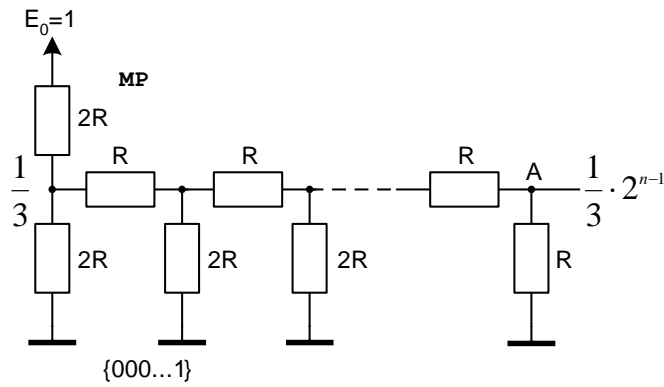


Рисунок 1.5 – ЦАП на матриці резисторів R-2R.





г)

Рисунок – 1.6 Розрахунок матриці резисторів

Отже, у результаті переміщення одиниці від старшого розряду (СР) до молодшого розряду (МР) резистори матриці $R - 2R$ утворюють числовий (двійковий) ряд розрядів вхідного регістру n -розрядного ЦАП

$$\frac{1}{3 \cdot 2^0}, \frac{1}{3 \cdot 2^1}, \frac{1}{3 \cdot 2^2}, \dots, \frac{1}{3 \cdot 2^{n-1}},$$

постійним коефіцієнтом якого є число $1/3$.

При $N_{2\max} = \{111\dots 1\}$ еквівалентний опір матриці мінімальний, причому у $3 \cdot 2^{n-1}$ менший за R , що забезпечує на виході ЦАП

$$U_{\text{вих}} = U_{\max} = \left| \frac{(2^n - 1)R_0}{2^{n-1}3R} \right|. \quad (1.10)$$

Таким чином, при $E_0 = 1$ на виході n -розрядного ЦАП на основі матриці $R - 2R$ отримуємо напругу:

$$\begin{aligned} U_{\text{вих}} &= -\frac{R_0}{3 \cdot 2^{n-1} R} (2^{n-1} X_{n-1} + 2^{n-2} X_{n-2} + \dots + 2^1 X_1 + 2^0 X_0) = \\ &= -\frac{R_0}{3 \cdot 2^{n-1} R} N_2, \end{aligned} \quad (1.11)$$

а якщо задати $R = 3R_0$, то

$$U_{\text{вих}} = \left| 2^{1-n} N_2 \right|. \quad (1.12)$$

Особливістю розглянутої схеми ЦАП з використанням ЛЕ є те, що його швидкодія визначатиметься швидкістю перемикання ЛЕ та значенням паразитних ємностей

матриці резисторів. ЦАП даного типу легко реалізувати за допомогою напівпровідникової та гібридної інтегральної технології тому, що матриця $R - 2R$ займає мінімальну площу на поверхні кристалу і дозволяє знизити до мінімуму розподілені паразитні параметри резисторів та з'єднувальних провідників.

До різновидів ЦАП належать схеми, в яких використовуються комутатори розрядних струмів на активних елементах (генераторах струму), зокрема на біполярних (або КМОН) транзисторах (ключах). Перевагою ЦАП на струмових ключах є висока швидкодія, що зумовлена малими значеннями сталих часу та прискореним перезарядом паразитних ємностей.

2 Лабораторна установка

Схема 1

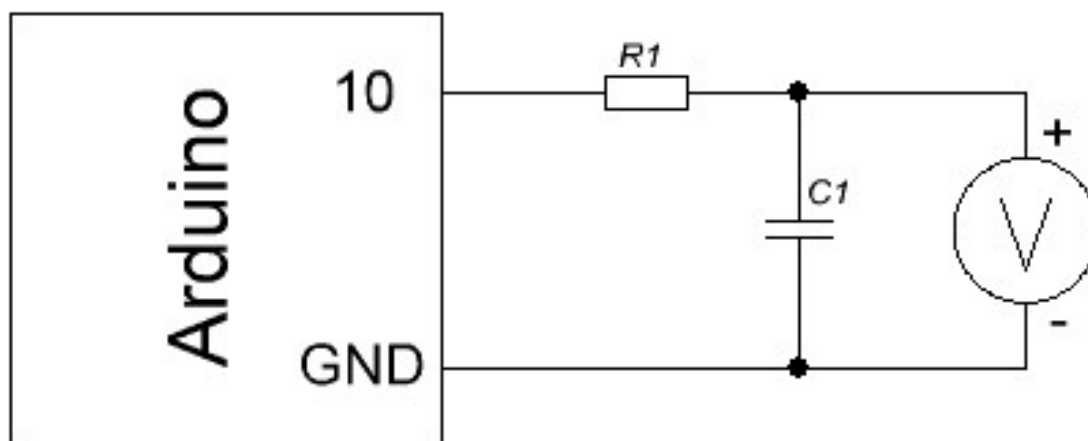
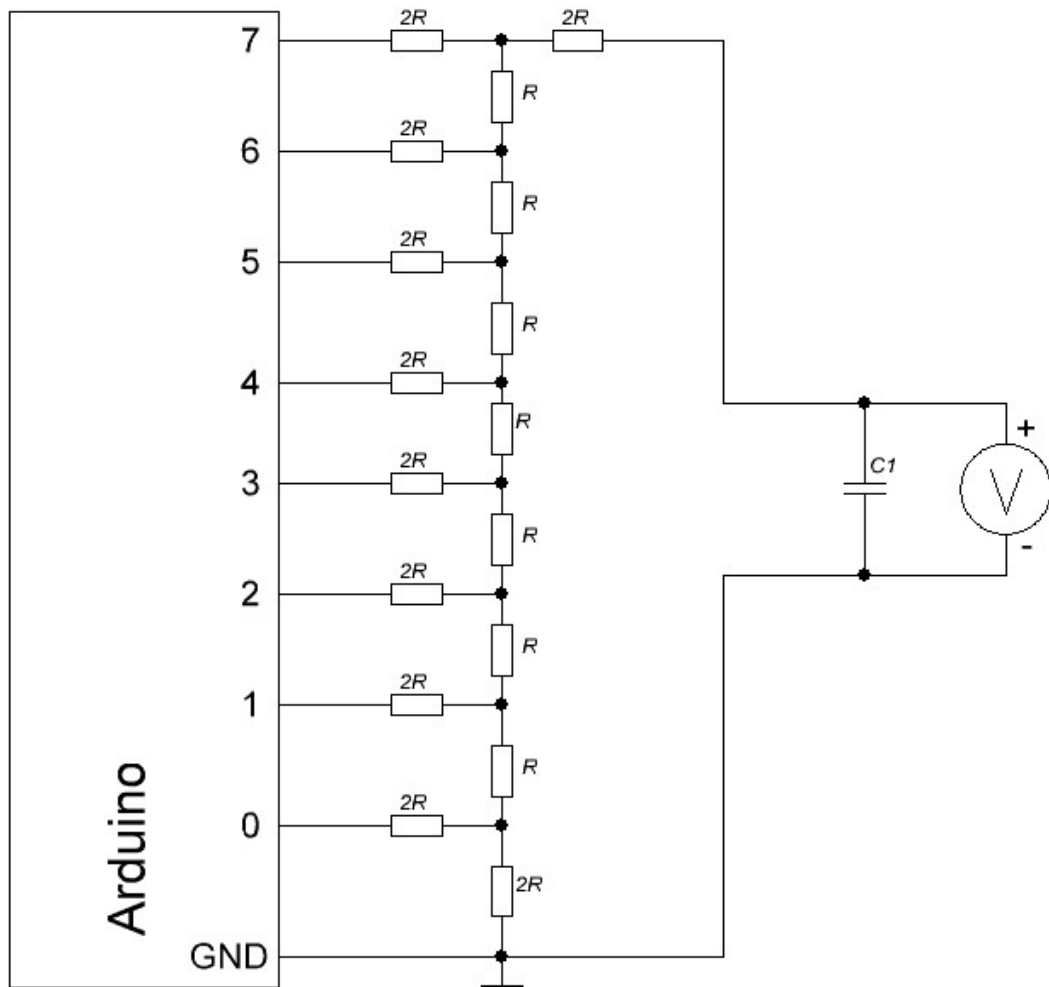


Схема 2



3 Хід виконання роботи

3.1 Ознайомтесь з лабораторною установкою та додатковими вказівками по роботі з приладами.

3.2 Розрахуйте номінал конденсатор низькочастотного фільтра для частоти зрізу 50 Гц, прийнявши опір резистора 1кОм.

3.3 Зберіть першу схему, та перевірте правильність. Розрахуйте напругу

$$\text{одного відліку за формулою } U_{\text{відл}} = \frac{U_{\text{опорна}}}{2^{\text{Розрядність ЦАП}}} \cdot$$

3.4 Розрахуйте кількість відліків для напруг в діапазоні 0-5 В та занесіть їх в таблицю.

3.5 Напишіть програму в Arduino IDE, яка буде формувати ШІМ сигнал на 10 піні. Використовуйте для цього функцію `analogWrite()`, змінюйте аргумент, відносно до кожного розрахункового значення. Після кожної зміни вносіть виміряну напругу вольтметром до відповідної колонки.

3.6 Зберіть другу схему, напишіть програму в Arduino IDE, яка буде двійковий код на цифрових пінах 0-7, пін 0 – молодший біт, 7 – старший біт. Використовуйте для цього функцію `digitalWrite()`, змінюйте значення, відносно до кожного розрахункового значення. Після кожної зміни вносіть виміряну напругу вольтметром до відповідної колонки.

№ п/п	Кількість відліків	ЦАП з використанням ШІМ		ЦАП з використанням R2R	
		Розрахункова напруга, В	Напруга на вольтметрі, В	Розрахункова напруга, В	Напруга на вольтметрі, В

4 Розрахункове завдання

Розрахункове завдання – див. п. 3.2, п. 3.3, п. 3.4 та п. 3.5.

5 Вимоги до звіту

Звіт з лабораторної роботи повинен містити:

1. Коротке описання мети і методики проведення роботи.
2. Перелік використаних приладів та матеріалів.
3. Таблиці результатів вимірювань, графічне оформлення.
2. табличних даних по п.3.4.
4. Результати вимірювань по пп. 3.4, 3.6.
5. Програмний код із середовища Arduino IDE з коментарями.
6. Розрахункове завдання.
7. Висновки.

6 Контрольні питання

1. Що означає термін ЦАП?
2. Які бувають види ЦАП?
3. Що означає термін квантування?
4. Теорема Котельнікова-Найквіста?
5. Що таке частота дискретизації?
6. Що означає поняття опорна напруга?
7. Де використовуються ЦАП?
8. Що таке квантування?
9. Чи залежить полоса пропускання від частоти дискретизації, чому?
10. Як працює R2R матриця?

Лабораторна робота №4

UART. УНІВЕРСАЛЬНИЙ ПОСЛІДОВНИЙ ІНТЕРФЕЙС ПЕРЕДАЧІ ДАНИХ

Мета роботи:

4. Практичне ознайомлення з універсальним послідовним портом передачі даних UART.
5. Підключення та налаштування UART.
6. Передача даних від плати Arduino до комп'ютера за допомогою інтерфейсу UART.

1 Короткі теоретичні відомості

1.1 Основні поняття

Універсальний асинхронний приймач-передавач UART (Universal Asynchronous Receiver Transmitter) призначений для забезпечення послідовного обміну даними. Інтерфейс UART являє собою повнодуплексний інтерфейс, тобто приймач і передавач можуть працювати одночасно, незалежно один від одного.

До складу UART входять:

- тактовий генератор зв'язку (бодрейт-генератор),
- керуючі регістри,
- статусні регістри,
- буфери
- регістри зсуву приймача й передавача.

Бодрейт-генератор задає тактову частоту прийомо-передавача для даної швидкості зв'язку.

Керуючі регістри задають режим роботи послідовного порту і його переривань.

У статусному регістрі встановлюються прапори за різними подіями.

У буфер приймача потрапляє прийнятий символ, у буфер передавача поміщають переданий.

Регістр зсуву передавача видає біти переданого символу (кадру). Регістр зсуву приймача накопичує прийняті з порту біти. По різних подіях встановлюються прапори й генеруються переривання (завершення прийому/відправлення кадру, вивільнення буфера, різні помилки).

Лінію порту приймача позначають RX, передавача - TX. Послідовною установкою рівнів на цих лініях щодо загального провідника ("землі") і передається інформація. За замовчуванням передавач встановлює на лінії одиничний рівень (логічна 1). Передача починається послілкою біта з нульовим рівнем (старт-біта), потім йдуть біти даних молодшим бітом уперед (низький рівень - "0", високий рівень - "1"), завершується послілкою біта з одиничним рівнем (стоп-біт).

Електричний сигнал кадру послілки виглядає так:

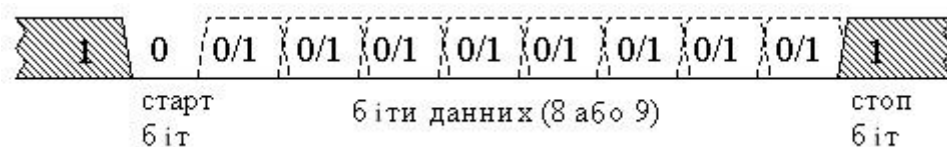


Рисунок 1 – Кадр послілки UART

Перед початком зв'язку між двома пристроями необхідно настроїти їхні прийомопередавачі на однакову швидкість зв'язку й формат кадру.

Швидкість зв'язку (baudrate) виміряється в бодах як число переданих бітів у секунду (включаючи старт і стоп-біти). Задається ця швидкість у бодрейт-генераторі діленням системної частоти на заданий коефіцієнт. Типовий діапазон

швидкостей: 2400 ... 115200 бод.

Формат кадру визначає число стоп-бітів (1 або 2), число бітів даних (8 або 9), а також призначення дев'ятого біта даних. Все це залежить від типу контролера.

Приймач і передавач тактується, як правило, з 16-кратною частотою відносно бодрейту. Приймач, визнавши наявність падаючого фронту старт-біта, відраховує кілька тактів і наступні три такти зчитує (семплює) порт RX. Це саме середина старт-біта. Якщо більшість значень семплів - "0", старт-біт вважається прийнятим, інакше приймач приймає його за шум і чекає наступного падаючого фронту.

Після вдалого визначення старт-біта, приймач так само семплює середини бітів даних і по більшості семплів визначає біт "0" або "1", записуючи їх у регістр зсуву. Стоп-біти теж семплюються, і якщо рівень стоп-біту не "1" - UART визначає помилку кадру й установлює відповідний прапор у керуючому регістрі.

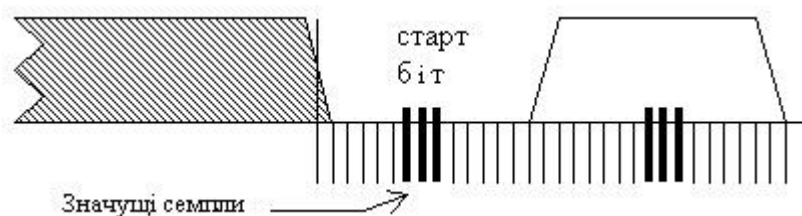


Рисунок 2 – Визначення бітів

Оскільки бодрейт установлюється діленням системної частоти, при перенесенні програми на пристрій з іншим кварцовим резонатором, необхідно змінити відповідні налаштування UART.

1.2 Організація обміну даними між платою Arduino і комп'ютером через інтерфейс UART

Для зв'язку з платою Arduino можна використовувати спеціальну програму моніторингу послідовного порту (Serial Monitor), вбудовану в програмне

забезпечення Ардуіно. Монітор послідовної шини відображає дані, що надходять в платформу Arduino (плата USB або плата послідовної шини). Для відправки даних вибирається швидкість передачі зі списку, відповідна значенню `Serial.begin` в скетчі. Потім необхідно ввести текст і натиснути кнопку Send або Enter.

Плати Arduino в яких основний мікроконтролер ATmega328 мають один послідовний порт (також відомий як UART або USART): Serial. Він пов'язаний з цифровими виводами 0 (RX) і 1 (TX), а також використовується для зв'язку з комп'ютером через USB. Таким чином, під час використання послідовного порту, виводи 0 і 1 не можуть використовуватися в якості цифрових входів або виходів.

Для забезпечення зв'язку плати Ардуіно з комп'ютером або іншими пристроями використовується клас Serial. Клас - це абстрактний тип даних. За допомогою класу описується деяка сутність (її характеристики і можливі дії). Наприклад, клас може описувати змінні, параметри і функції послідовного порту. Клас Serial містить близько 20 функцій. Розглянемо деякі з них.

1.3 Функції для роботи з послідовним портом плати Arduino

if (Serial)

Параметри – Немає. **Значення, що повертаються** - boolean: повертає true, якщо вказаний послідовний порт готовий до роботи.

Опис: Дозволяє перевірити готовність певного послідовного порту.

Serial.begin(speed)

Параметри:

speed: швидкість в бітах на секунду (бодах) – long

Опис: Задає швидкість передачі даних по послідовному інтерфейсу в бітах в секунду (бодах). Для взаємодії з комп'ютером слід використовувати одну з попередньо встановлених швидкостей обміну: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 або 115200.

Serial.end ()

Параметри – Немає. Значення, що повертаються - немає

Опис: Функція розриває послідовний зв'язок, після чого виводи RX і TX знову можуть використовувати як виводи загального призначення. Для відновлення послідовного з'єднання необхідно використовувати функцію *Serial.begin ()*.

Serial.available()

Параметри – Немає. Значення, що повертаються: кількість байт, доступних для зчитування.

Опис: Повертає кількість байт (символів) доступних для зчитування з буфера послідовного порту. Під символами розуміються дані, які вже прийняті і зберігаються в послідовному приймальному буфері (який може зберігати максимум 64 байти).

Serial.read()

Параметри – Немає. Значення, що повертаються: Перший байт прийнятих даних (або -1, якщо таких нема) - int

Опис: Зчитує дані, що надходять по послідовному інтерфейсу.

Serial.print(val)

Serial.print(val, format)

Параметри:

val: значення, яке необхідно вивести - будь-який тип даних

format: визначає систему числення (для цілочисельних типів), а також кількість десяткових знаків після коми (для чисел з плаваючою крапкою).

Значення, що повертаються: `size_t (long)`: функція *print()* повертає кількість виведених байт. Зчитування цього значення не обов'язково.

Опис: Функція виводить через послідовний порт заданий ASCII текст у вигляді, зрозумілому для людини. Ця команда може мати кілька різних форм.

Serial.println(val)

Serial.println(val, format)

Параметри:

val: значення, яке необхідно вивести - будь-який тип даних

format: визначає систему числення (для цілочисельних типів), а також кількість десяткових знаків після коми (для чисел з плаваючою точкою).

Значення, що повертаються:

`size_t (long)`: функція *println()* повертає кількість виведених байт.

Зчитування цього значення не обов'язково.

Опис: Виводить через послідовний порт ASCII-текст в зрозумілому для людини вигляді з символами повернення каретки (ASCII 13 або '\ r') і нового рядка (ASCII 10 або '\ n'). Ця команда має такі ж форми, як і *Serial.print()*.

При виведенні числа кожній його цифрі відповідає один ASCII-символ. Дробові числа теж виводяться у вигляді ASCII-цифр, при цьому після коми за замовчуванням залишається два десяткових знака. Байти виводяться у вигляді окремих символів, а символи і рядки виводяться без змін - "як є". Наприклад:

Serial.print(78) - виведе "78"

Serial.print(1.23456) - виведе "1.23"

`Serial.print('N')` - виведе "N"

`Serial.print("Hello world.")` - виведе "Hello world."

Другий параметр необов'язковий. Він задає формат виведення; цей параметр може набувати таких значень: BIN (двійкова система з основою 2), OCT (восьмерична система з основою 8), DEC (десятькова система з основою 10), HEX (шістнадцятирична система з основою 16). Для чисел із плаваючою крапкою (комою) цей параметр визначає кількість десяткових знаків після коми.

Наприклад:

`Serial.print (78, BIN)` - виведе "1001110"

`Serial.print (78, OCT)` - виведе "116"

`Serial.print (78, DEC)` - виведе "78"

`Serial.print (78, HEX)` - виведе "4E"

`Serial.println (1.23456, 0)` - виведе "1"

`Serial.println (1.23456, 2)` - виведе "1.23"

`Serial.println (1.23456, 4)` - виведе "1.2346"

2 Лабораторна установка

Лабораторна установка складеться з плати Arduino (в разі відсутності, емулятор плати) та персонального компютера з програмним забезпеченням Arduino IDE.

3 Хід виконання роботи

3.1 Ознайомтесь з лабораторною установкою та додатковими вказівками по роботі з приладами.

3.2 Підключіть плату Arduino до комп'ютера та запустіть Arduino IDE .

3.3 Налаштуйте порт на роботу при швидкості 9600 бод/с.

3.4 Напишіть програму яка буде виводити в монітор порту ваше прізвище, ім'я та групу в якій ви навчаєтесь, використовуючи наведені вище функції.

3.5 Намалюйте фрейм який буде передавати символ першої літери вашого імені, параметри такі: 1- стартовий біт, 1- стоповий, парність – відсутня. Як приклад можете використати зображений фрейм нижче, в якому виводиться символ велика англійська буква 'K' з таблиці ASCII, яка має порядковий номер 75.

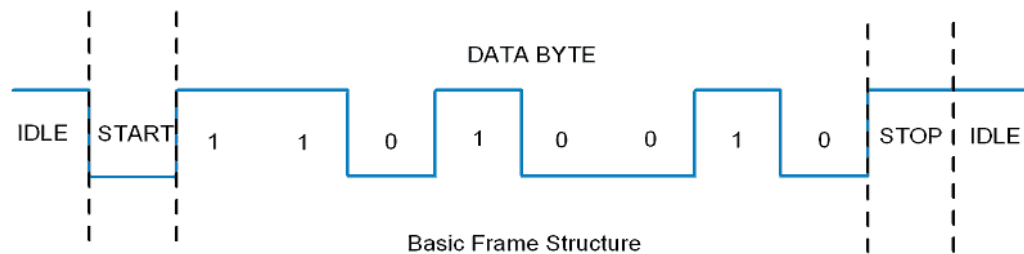


Рисунок 3 – Фрейм який передає значення 75.

3.6 Намалюйте фрейм який буде передавати символ першої літери вашого прізвища, параметри такі самі як і в попередньому пункті 3.5.

3.7 Намалюйте фрейм який буде передавати ваш порядковий номер в журналі груп в якій ви навчаєтесь, параметри такі самі як і в попередньому пункті 3.5.

4 Розрахункове завдання

Розрахункове завдання – див. п. 3.5, п. 3.6, п. 3.7.

5 Вимоги до звіту

Звіт з лабораторної роботи повинен містити:

- 4 Коротке описання мети і методики проведення роботи.
- 5 Перелік використаних приладів та матеріалів.

- 6 Таблиці результатів вимірювань, графічне оформлення.
- 7 Програмний код із середовища Arduino IDE з коментарями.
- 8 Розрахункове завдання.
- 9 Висновки.

6 Контрольні питання

1. Що означає термін UART?
2. Якою може бути швидкість передачі даних через UART?
3. Яка структура пакету даних?
4. Скільки інтерфейсів UART має плата Arduino?
5. Як організовано обмін між платою Arduino та комп'ютером?
6. Що означає поняття baudrate (бодрейт)?
7. Де використовується інтерфейс UART?
8. Який клас використовується в Arduino для роботи з UART?
9. Чи потрібно якось налаштувати приймаючий пристрій, як і чому?
10. Чи можливе використання протоколу передачі даних UART в інших інтерфейсах передачі даних, яких?

ЦИФРО-БУКВЕНІ ІНДИКАТОРИ

Мета роботи:

7. Практичне ознайомлення з цифро-буквеними індикаторами.
8. Підключення індикаторів.
9. Вивід числових даних на 7-ми сегментні індикатори.

1 Короткі теоретичні відомості

1.1 Основні поняття

Цифро-буквені індикатори на основі світлодіодних являють собою інтегральну мікросхему з діодних структур (у вигляді сегментів або точечних елементів) та необхідних електричних з'єднань.

Світлодіоди являють собою напівпровідникові діоди, призначені для перетворення електричної енергії в енергію некогерентного світлового випромінювання. При протіканні через діод прямого струму відбувається інжекція неосновних носіїв заряду (електронів або дірок) у базову область діодної структури. Процес мимовільної рекомбінації інжектованих неосновних носіїв заряду, що відбувається як у базовій області, так і в самому р – n переході, супроводжується їхнім переходом з більш високого енергетичного рівня на більш низький. При цьому надлишкова енергія виділяється шляхом випромінювання кванта світла.

Для виготовлення світлодіодів використовуються наступні напівпровідникові матеріали:

- фосфід галію;
- карбід кремнію;
- тверді розчини галій-миш'як-фосфор та галій-миш'як-алюміній.

Шляхом додавання в напівпровідникові матеріали атомів речовин-активаторів можна змінювати колір випромінювання діодів. Наприклад, на основі фосфіду галію з домішкою цинку, кисню або азоту одержують прилади зеленого, жовтого і червоного кольорів перетину.

Готові сегменти (смужки) з зазначених матеріалів розташовують на одній підкладці таким чином, щоб при відповідній комбінації збуджених сегментів досягалося чітке зображення однієї цифри або букви.

Однорозрядні цифро-буквені індикатори дозволяють відтворювати будь-яку цифру від 0 до 9, а також ряд букв та символів. По числу сегментів, що приходяться на один розряд, індикатори поділяються на 7-, 9-, та 35- елементні. Деякі індикатори мають децимальну крапку.

Для відображення цифрової інформації найчастіше застосовують семи сегментні індикатори, в яких цифра складається з семи лінійних світлодіодних сегментів розташованих у вигляді цифри 8. (рис. 1) На основі світлодіодів та семисегментних індикаторів будуються підсистеми відображення інформації.

Семисегментні індикатори являють собою збірки світлодіодів. Існують два варіанти збірок. Схема с загальним анодом, та схема с загальним катодом. Якщо використано схему с загальним катодом, то світлодіоди сегментів цифр з'єднано катодами. Для того, щоб отримати зображення на такому індикаторі, необхідно на загальний катод цифри подати рівень напруги логічного нуля, а на ті сегменти, які беруть участь у формуванні цифри – подати високий рівень напруги (логічна „1”). Наприклад, для формування цифри “2” на виводи А, В, G, Е, D потрібно подати логічну одиницю.



Рисунок 1 – Різновиди 7-ми сегментних індикаторів

При побудові підсистем відображення інформації розрізняють два підходи – динамічна і статична схема побудови підсистеми індикації. Статична індикація полягає в постійному підсвічуванні індикаторів від одного джерела інформації. Сутність динамічної індикації полягає в почерговому циклічному підключенні кожного індикатора до джерела інформації через загальну шину даних. При такому включенні значно зменшуються апаратні витрати. Але необхідно забезпечити достатній час світіння одного індикатора, для того щоб не зменшувалась яскравість. Також необхідно забезпечити таку частоту перебору індикаторів, щоб не було помітно мерехтіння.

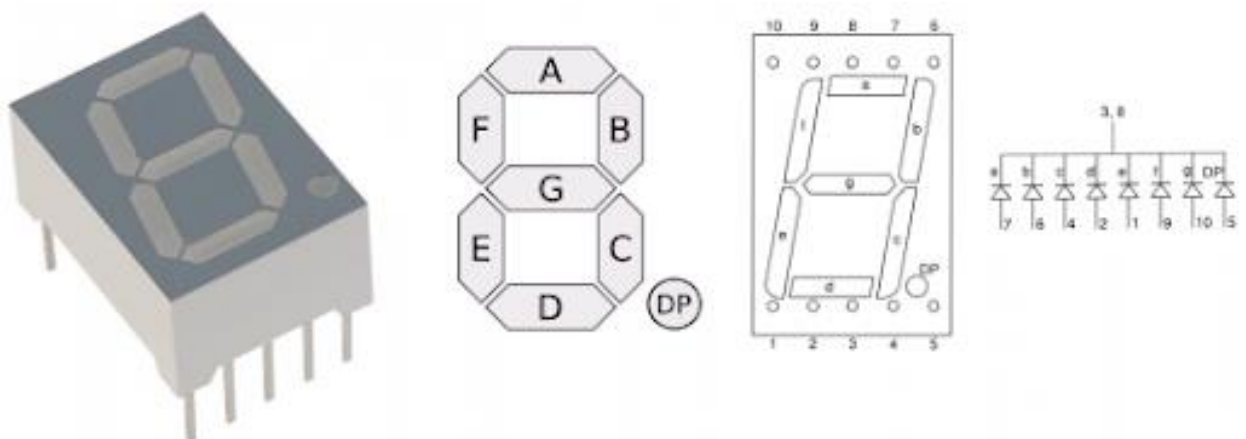


Рисунок 2 – Положення сегментів та схема 7-ми сегментного індикатора

2 Лабораторна установка

Лабораторна установка складеться з плати Arduino (в разі відсутності, емулятор плати), семисегментного індикатора, струмообмежуючих резисторів, транзисторів та персонального компютера з програмним забезпеченням Arduino IDE.

3 Хід виконання роботи

3.1 Ознайомтесь з лабораторною установкою та додатковими вказівками по роботі з приладами.

3.2 Підключіть плату Arduino до комп'ютера та запустіть Arduino IDE .

3.3 Складіть таблицю істинності виводу для виводу цифр 0-9 та символів А,b,C,d,E,F, занесіть дані до таблиці.

Сегмент	A	B	C	D	E	F	G
Символ							
0							
1							
2							
3							
4							
5							
6							
7							
8							
9							
A							
b							
C							
d							
E							
F							

3.4 Складіть схему зображену на рис.3.

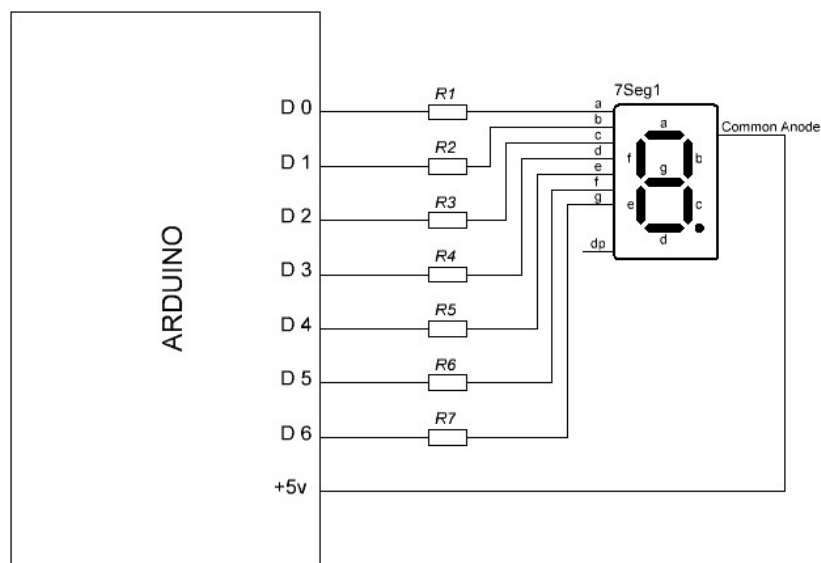


Рисунок 3 – Схема лабораторної установки

3.5 Розрахуйте опір струмообмежуючих резисторів для червоного кольору світлодіодів сегментів, напруга живлення 5 В.

3.6 Напишіть програму яка буде виводити на індикатор Ваш порядковий номер в журналі, якщо він більший за 9, тоді останній знак (наприклад якщо номер 13, тоді виводите знак 3).

3.7 Зробіть фото або скрін виведеного знаку.

4 Розрахункове завдання

Розрахункове завдання – див. п. 3.3, п. 3.5.

5 Вимоги до звіту

Звіт з лабораторної роботи повинен містити:

1. Коротке описання мети і методики проведення роботи.
2. Перелік використаних приладів та матеріалів.
3. Таблиці результатів вимірювань, графічне оформлення.
4. Програмний код із середовища Arduino IDE з коментарями.

5. Розрахункове завдання.
6. Висновки.

6 Контрольні питання

1. Що є основними електричними параметрами індикаторів?
2. За допомогою чого змінюють колір випромінювання світлодіодних індикаторів?
3. Як зменшити кількість використовуваних пінів МК для роботи з індикаторами?
4. Що таке динамічна індикація?
5. Де застосовуються індикатори?

ДРАЙВЕРИ ДЛЯ 7-МИ СЕГМЕНТРИХ ІНДИКАТОРІВ

Мета роботи:

1. Практичне ознайомлення з цифро-буквеними індикаторами та допоміжними мікросхемами для них.
2. Підключення індикаторів.
3. Вивід числових даних на 7-ми сегментні індикатори за допомогою мікросхеми CD4511.

1 Короткі теоретичні відомості

1.2 Основні поняття

Цифро-буквені індикатори на основі світлодіодних являють собою інтегральну мікросхему з діодних структур (у вигляді сегментів або точечних елементів) та необхідних електричних з'єднань.

Світлодіоди являють собою напівпровідникові діоди, призначені для перетворення електричної енергії в енергію некогерентного світлового випромінювання. При протіканні через діод прямого струму відбувається інжекція неосновних носіїв заряду (електронів або дірок) у базову область діодної структури. Процес мимовільної рекомбінації інжектованих неосновних носіїв заряду, що відбувається як у базовій області, так і в самому р – n переході, супроводжується їхнім переходом з більш високого енергетичного рівня на більш низький. При цьому надлишкова енергія виділяється шляхом випромінювання кванта світла.

Для виготовлення світлодіодів використовуються наступні напівпровідникові матеріали:

- фосфід галію;
- карбід кремнію;

- тверді розчини галій-миш'як-фосфор та галій-миш'як-алюміній.

Шляхом додавання в напівпровідникові матеріали атомів речовин-активаторів можна змінювати колір випромінювання діодів. Наприклад, на основі фосфіду галію з домішкою цинку, кисню або азоту одержують прилади зеленого, жовтого і червоного кольорів перетину.

Готові сегменти (смужки) з зазначених матеріалів розташовують на одній підкладці таким чином, щоб при відповідній комбінації збуджених сегментів досягалося чітке зображення однієї цифри або букви.

Однорозрядні цифро-буквені індикатори дозволяють відтворювати будь-яку цифру від 0 до 9, а також ряд букв та символів. По числу сегментів, що приходяться на один розряд, індикатори поділяються на 7-, 9-, та 35- елементні. Деякі індикатори мають децимальну крапку.

Для відображення цифрової інформації найчастіше застосовують семи сегментні індикатори, в яких цифра складається з семи лінійних світлодіодних сегментів розташованих у вигляді цифри 8. (рис. 1) На основі світлодіодів та семисегментних індикаторів будуються підсистеми відображення інформації.

Семисегментні індикатори являють собою збірки світлодіодів. Існують два варіанти збірок. Схема с загальним анодом, та схема с загальним катодом. Якщо використано схему с загальним катодом, то світлодіоди сегментів цифр з'єднано катодами. Для того, щоб отримати зображення на такому індикаторі, необхідно на загальний катод цифри подати рівень напруги логічного нуля, а на ті сегменти, які беруть участь у формуванні цифри – подати високий рівень напруги (логічна „1”). Наприклад, для формування цифри “2” на виводи А, В, G, Е, D потрібно подати логічну одиницю.



Рисунок 1 – Різновиди 7-ми сегментних індикаторів

При побудові підсистем відображення інформації розрізняють два підходи – динамічна і статична схема побудови підсистеми індикації. Статична індикація полягає в постійному підсвічуванні індикаторів від одного джерела інформації. Сутність динамічної індикації полягає в почерговому циклічному підключенні кожного індикатора до джерела інформації через загальну шину даних. При такому включенні значно зменшуються апаратні витрати. Але необхідно забезпечити достатній час світіння одного індикатора, для того щоб не зменшувалась яскравість. Також необхідно забезпечити таку частоту перебору індикаторів, щоб не було помітно мерехтіння.

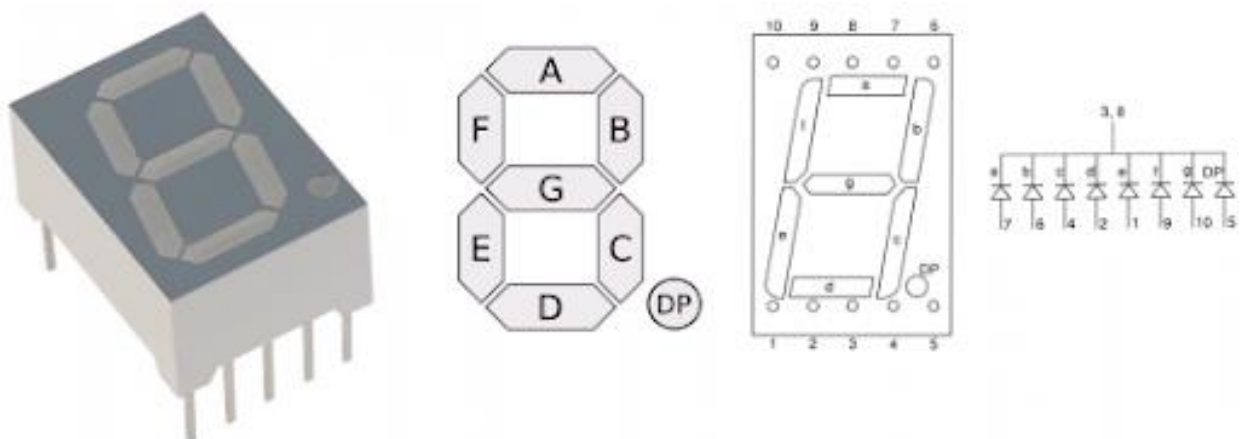


Рисунок 2 – Положення сегментів та схема 7-ми сегментного індикатора

Щоб скоротити число задіяних виводів мікроконтролера зазвичай на практиці використовуються різні рішення. Наприклад, популярним способом управління 7-сегментним індикатором є застосування мікросхеми CD4511 - двійково-десятькового перетворювача. Він переводить двійковий код числа в напругу на відповідних сегментах індикатора. Такий перетворювач буде використовувати всього 4 ніжки Arduino. Тобто, наприклад, якщо необхідно відобразити на індикаторі десяткове число 7, необхідно виставити на вході перетворювача двійкове 0111 ("LOW HIGH HIGH HIGH"), для індикаторів зі спільним катодом. Якщо використовується індикатор зі спільним анодом, тоді потрібно використовувати додатковий інвертор на транзисторі, так як зображено на рис.3.

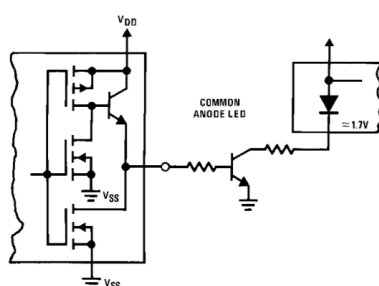


Рисунок 3 – Підключення індикатора зі спільним анодом.

Мікросхема CD4511 виконується в різних типах корпусів. Призначення виводів у виконанні з 16-ма ніжками, таке:

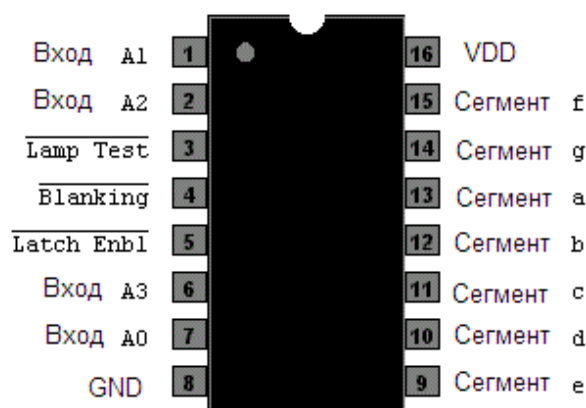


Рисунок 4 – Призначення виводів CD4511

2 Лабораторна установка

Лабораторна установка складеться з плати Arduino (в разі відсутності, емулятор плати), 2 семисегментних індикатора, мікросхеми CD4511, струмообмежуючих резисторів, транзисторів та персонального компютера з програмним забезпеченням Arduino IDE.

3 Хід виконання роботи

- 3.1 Ознайомтесь з лабораторною установкою та додатковими вказівками по роботі з приладами.
- 3.2 Підключіть плату Arduino до комп'ютера та запустіть Arduino IDE .
- 3.3 Складіть таблицю істинності виводу для виводу цифр 0-9 та символів A,b,C,d,E,F, занесіть дані до таблиці.

Сегмент	A	B	C	D	E	F	G
Символ							
0							
1							
2							
3							
4							
5							
6							
7							
8							
9							
A							
b							
C							
d							

E							
F							

3.4 Випишіть таблицю істинності для мікросхеми CD4511 з документації (даташиту) на дану мікросхему.

3.5 Складіть схему зображену на рис.3.

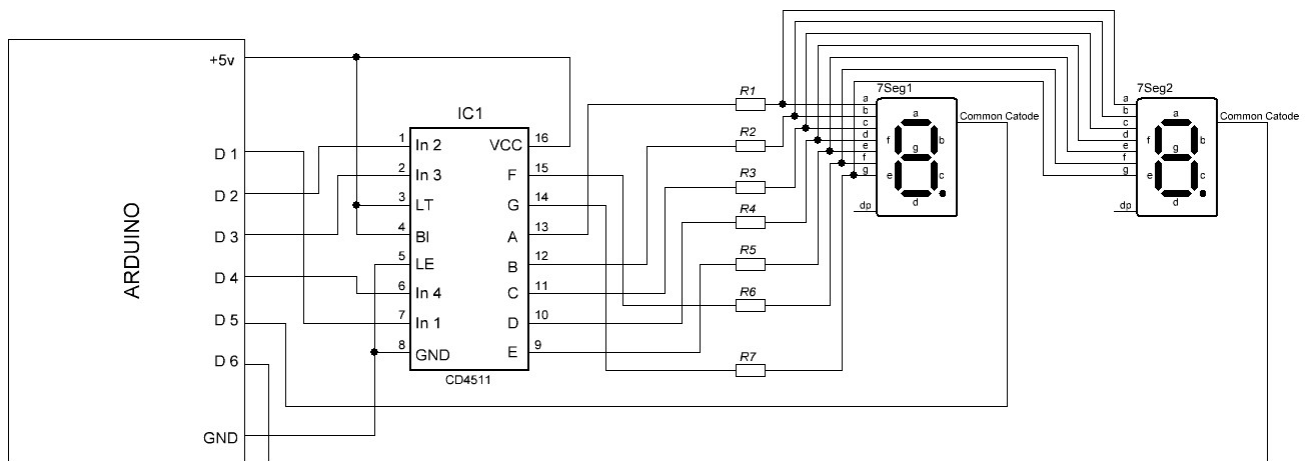


Рисунок 3 – Схема лабораторної установки

3.6 Розрахуйте опір струмообмежуючих резисторів для червоного кольору світлодіодів сегментів, напруга живлення 5 В.

3.7 Напишіть програму яка буде виводити на індикатор Ваш порядковий номер в журналі, якщо він менший за 10, тоді перший знак поставите 0 (наприклад якщо номер 8, тоді виводите два знаки 08).

3.8 Для виведення знаків напишіть код який спочатку буде виводити перший символ, а потім другий, і так по колу. Потім розрахуйте тривалість паузи між виведенням символів, схематично буде виглядати так:

```
void loop()
{
  //виводимо перший символ
  // пауза
  // виводимо другий символ
  // пауза
```

}

- 3.9 Для того щоб виводити символи окремо, потрібно вмикати і вимикати спільний катод обох елементів в потрібний момент часу.
- 3.10 Зробіть фото або скрін виведених знаків.

4 Розрахункове завдання

Розрахункове завдання – див. п. 3.3, п. 3.4 та п. 3.6.

5 Вимоги до звіту

Звіт з лабораторної роботи повинен містити:

7. Коротке описання мети і методики проведення роботи.
8. Перелік використаних приладів та матеріалів.
9. Таблиці результатів вимірювань, графічне оформлення.
10. Програмний код із середовища Arduino IDE з коментарями.
11. Розрахункове завдання.
12. Висновки.

6 Контрольні питання

1. Що є основними електричними параметрами індикаторів?
2. За допомогою чого змінюють яскравість випромінювання світлодіодних індикаторів?
3. Як зменшити кількість використовуваних пінів МК для роботи з індикаторами?
4. Що таке динамічна індикація?
5. Що таке статична індикація?
6. Що таке драйвер для 7-ми сегментного індикатора?
7. Де застосовуються індикатори?

Лабораторна робота №7

РЕГІСТРИ

Мета роботи:

10. Практичне ознайомлення з регістрами.
11. Підключення регістрів та їх застосування.
12. Практичне використання регістру зсуву 74НС595.

1 Короткі теоретичні відомості

1.3 Основні поняття

Регістр є важливим функціональним вузлом будь-якого цифрового пристрою (комп'ютера). На сьогоднішній день регістри вже не використовуються у вигляді окремих інтегральних мікросхем (див. рисунок 1). Вони є частиною мікросхем більш високого ступеня інтеграції (мікропроцесори, мікроконтролери).

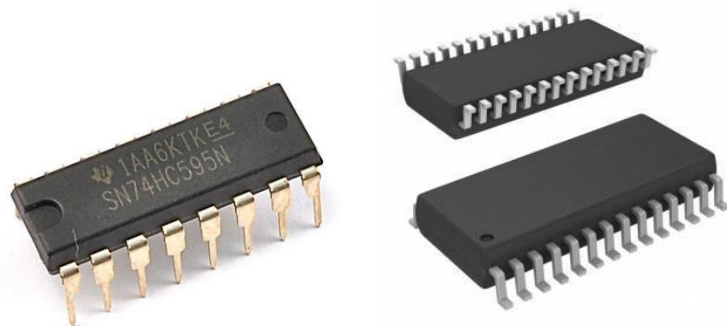


Рисунок 1 – Регістри у вигляді окремих інтегральних мікросхем

Регістри – це електронні пристрої, побудовані на основі тригерів і призначені для прийому, зберігання, перетворення і передачі інформації у формі двійкових чисел.

Розрядність регістра визначається кількістю тригерів, що входять до його складу, а отже і кількістю розрядів двійкового числа, яке може зберігати регістр.

Швидкодія регістрів визначається швидкодією тригерів, що входять до складу регістрів.

За способом прийому і передачі інформації у двійковій формі регістри поділяють на *паралельні, послідовні та універсальні*.

1.2 Паралельні регістри

Паралельні регістри використовуються як пам'ять з великою швидкістю у цифрових пристроях обробки інформації. Їх називають також *регістрами пам'яті*.

Запис і видача двійкових чисел (слів) проводиться одночасно усіма розрядами регістра.

Схема паралельного регістра на D-тригерах представлена на рисунку 2.

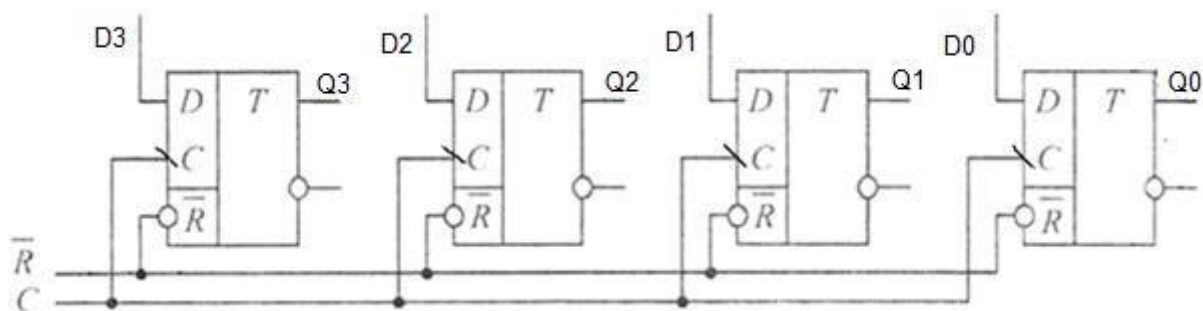


Рисунок 2 - Паралельний регістр на D-тригерах

Дана схема призначена для зберігання чотирирозрядного двійкового числа.

Розглянемо принцип роботи паралельного регістра на D-тригерах.

Для того, щоб зберегти будь-яке число у такому регістрі, потрібно подати це число на входи D0, D1, D2, D3. Далі на вхід C схеми подається імпульс запису. По задньому фронту цього імпульсу число записується в регістр. Причому кожний розряд числа (0 або 1) записується у свій окремий D-тригер. Записане в регістр число можна зчитати з нього за допомогою виходів Q0, Q1, Q2, Q3. Вхід

\bar{R} використовується для початкової установки всіх розрядів регістра у нульовий стан. Вхід \bar{R} – інверсний, тому при подачі на нього логічного 0 відбувається скидання всіх тригерів регістра у нульовий стан.

1.3 Послідовні регістри

Послідовні регістри використовують операцію зсуву під час послідовного введення чи виведення інформації.

Зсув – це одночасне просторове переміщення двійкового слова у розрядній сітці зі збереженням порядку прямування нулів та одиниць.

Зсув двійкового числа реалізується шляхом перезапису стану між сусідніми тригерами регістра у напрямі зсуву. Таким чином, кожний тригер регістра одночасно приймає інформацію з попереднього розряду (тригера) і передає у наступний.

Схема чотирирозрядного регістра з послідовним введенням інформації і зсувом її праворуч на базі D-тригера представлена на рисунку 3.

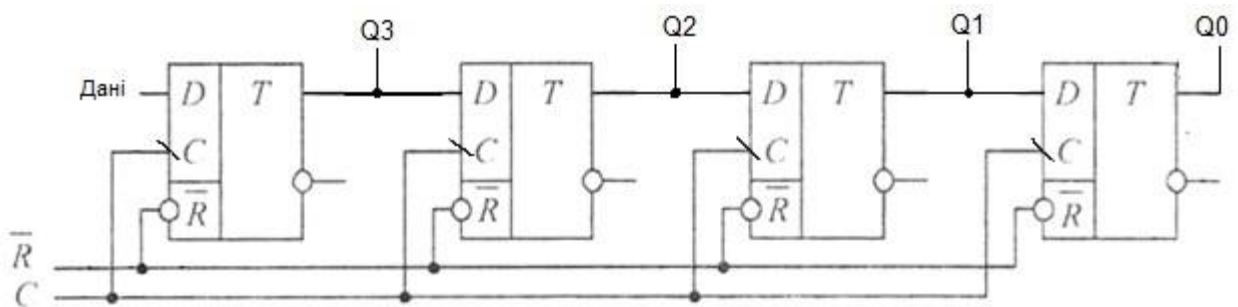


Рисунок 3 - Послідовний регістр на D-тригерах

Розглянемо принцип роботи послідовного регістра на D-тригерах.

Вхід D кожного тригера, окрім крайнього ліворуч, підключений до прямих виходів попереднього тригера. Входи \bar{R} об'єднані у спільну шину «Скидання». Входи C об'єднані у спільну шину «Синхронізація». Тому при кожному тактовому імпульсі (синхроімпульсі), який подається на шину «Синхронізація» усі тригери, за виключенням крайнього ліворуч, будуть приймати стан лівого сусіднього тригера. Стан першого лівого тригера визначається сигналом на лінії «Дані».

Стан крайнього праворуч тригера втрачається при кожному синхроімпульсі, тобто відбувається зсув інформації праворуч.

Робота послідовного регістра зведена у наступну таблицю істинності:

Вхід «Дані»	Q3	Q2	Q1	Q0
0	0	X	X	X
1	1	0	X	X
1	1	1	0	X
0	0	1	1	0
1	1	0	1	1
0	0	1	0	1
1	1	0	1	0

У реверсивних регістрах зсув інформації може здійснюватись в обох напрямках.

2 Лабораторна установка

Лабораторна установка складеться з плати Arduino (в разі відсутності, емулятор плати), світлодіодів, мікросхеми 74НС595, струмообмежуючих резисторів та персонального компютера з програмним забезпеченням Arduino IDE.

3 Хід виконання роботи

- 3.1 Ознайомтесь з лабораторною установкою та додатковими вказівками по роботі з приладами.
- 3.2 Підключіть плату Arduino до комп'ютера та запусіть Arduino IDE .
- 3.3 Розгляньте документацію на мікросхему 74НС595 та выпишіть основні характеристики даної мікросхеми.
- 3.4 Складіть схему зображену на рис.3.

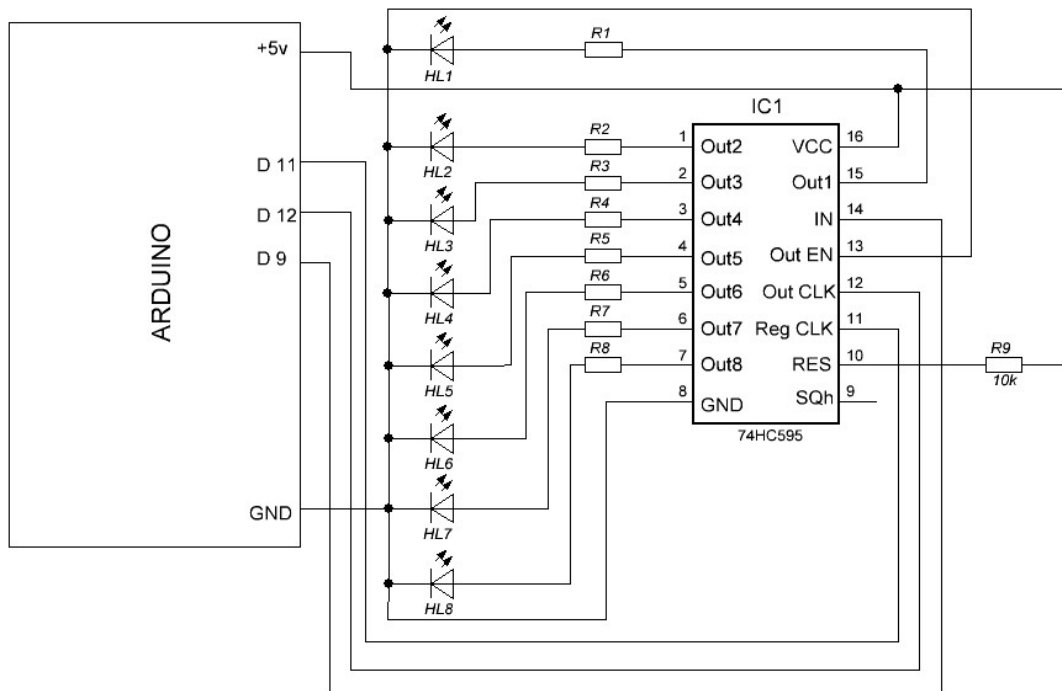


Рисунок 3 – Схема лабораторної установки

- 3.5 Розрахуйте опір струмообмежуючих резисторів для червоного та зеленого кольору світлодіодів, напруга живлення 5 В.
- 3.6 Напишіть програму яка буде виводити на світлодіоди в бінарному вигляді (8 біт) ASCII код першої літери Вашого прізвища, та порядковий номер в журналі по черзі, якщо він менший за 10, тоді перший знак поставите 0 (наприклад якщо номер 8, тоді виводите два знаки 08).
- 3.7 Для виведення інформації через регістр зсуву скористайтесь функцією `shiftOut(dataPin, bitOrder, value)`, попередньо налаштуйте використовувані піни на вихід, схематично буде виглядати так:

```
void setup()
{
  // налаштуємо піни
}

void loop()
{
  //вмикаємо передачу
```

```
// виводимо ASCII код першої літери
// завершуємо передачу
// пауза
// вмикаємо передачу
// виводимо порядковий номер
// завершуємо передачу
// пауза
}
```

3.8 Зробіть фото або скрін виведених даних.

4 Розрахункове завдання

Розрахункове завдання – див. п. 3.3, п. 3.5.

5 Вимоги до звіту

Звіт з лабораторної роботи повинен містити:

1. Коротке описання мети і методики проведення роботи.
2. Перелік використаних приладів та матеріалів.
3. Таблиці результатів вимірювань, графічне оформлення.
4. Програмний код із середовища Arduino IDE з коментарями.
5. Розрахункове завдання.
6. Висновки.

6 Контрольні питання

1. Які основні параметри регістрів?
2. За допомогою чого змінюють яскравість випромінювання світлодіодів?
3. Як зменшити кількість використовуваних пінів МК для роботи з індикаторами, та світлодіодами?
4. Чи можна використовувати регістри зсуву для зчитування даних?
5. Де використовуються регістри зсуву?

Лабораторна робота №8 **СТВОРЕННЯ ПРОСТОГО ПЛК НА БАЗІ ARDUINO**

Мета роботи:

13. Практичне ознайомлення з принципами роботи ПЛК.
14. Складання схеми найпростішого ПЛК.
15. Створення комунікаційного протоколу обміну для ПЛК.
16. Написання програми для ПЛК на базі Arduino.

1 Короткі теоретичні відомості

1.1. Загальні властивості контролерів

Програмований логічний контролер (ПЛК), який у міжнародних документах позначається як PLC (Programmable Logic Controller) – це мікропроцесорний пристрій, призначений для керування виробничими процесами в умовах промислового середовища в реальному масштабі часу, що програмується з використанням спеціалізованих мов програмування. Окрім ПЛК в промисловості використовуються й інші типи контролерів, наприклад РС-сумісні. У даній дисципліні розглядаються тільки програмовані логічні контролери, тому надалі в тексті терміни "контролер" та "ПЛК" вважаються синонімами, якщо інше не буде вказане явно.

Можна виділити декілька основних рис ПЛК:

1. Наявність спеціалізованих мов програмування, що максимально наближені до потреб управління в реальному часі, спрощують розробку, налагодження та модифікацію прикладних програм, можливість оперативної зміни алгоритму керування програмним шляхом. Ці мови стандартизовані в міжнародному стандарті МЕК 61131.

2. Блочно-модульний принцип побудови ПЛК (крім моноблочних контролерів), що дає можливість за рахунок використання різноманітних модулів входу-виходу оптимізувати його компонування для керування конкретним об'єктом. Тобто конфігурація ПЛК (перелік модулів, які входять до складу контролера) залежить від конкретного об'єкта управління і алгоритму управління ним. Це не тільки зменшує витрати на впровадження систем автоматизації, а й підвищує ремонтоздатність ПЛК.

3. Призначення ПЛК для використання в промислових умовах ставить досить жорсткі вимоги до надійності ПЛК та захищеності його від впливу різноманітних електромагнітних, вібраційних, кліматичних та інших перешкод. Це досягається за рахунок використання надійної елементної бази, стійких і надійних схемних рішень, спеціальних гальванічних розподільників,

резервування, дублювання та інших заходів, а також високого технологічного рівня виробництва ПЛК.

4. Наявність широко розвинутої системи самодіагностики та тестування, за допомогою яких можна швидко визначити несправність та усунути її. Це є дуже важливою функцією, оскільки ПЛК - досить складний технічний засіб і визначення причини відмови ПЛК потребує досить високої кваліфікації обслуговуючого персоналу. Тому в разі виникнення технічної несправності ПЛК система діагностики допомагає швидко визначити модуль, у якому виникла несправність. Цей модуль повинен бути швидко замінений технічним персоналом і роботоздатність системи управління буде поновлена. А несправний модуль повинен бути переданий на ремонт у сервісну службу виробника ПЛК.

5. Для забезпечення роботи ПЛК у складі розподіленої АСУТП передбачена можливість організації обміну інформацією між окремими ПЛК та передачі технологічної інформації в системи організаційно-економічного управління за рахунок широкого використання промислових мереж, польових шин та комп'ютерних мереж.

1.2. Структура контролерів

На рис. 1.1 показано узагальнену структуру програмованого логічного контролера. Центральною частиною ПЛК є мікропроцесорний пристрій (МП, CPU), який керує операціями збору і оброблення даних від зовнішніх пристроїв і вироблення керуючих дій відповідно до розробленої програми користувача.

Пам'ять контролера використовується для декількох цілей. Частина призначена для зберігання та виконання операційної системи, якою керується мікропроцесорний модуль, а також для розміщення системної інформації про оперативний стан апаратного і програмного забезпечення. Саме ця інформація використовується для функціонування системи самодіагностики і тестування контролера. Інша частина ресурсів пам'яті використовується для зберігання розробленої програми користувача, яка й визначає виконання алгоритму керування об'єктом. Ємність пам'яті, яка використовується для зберігання програми користувача, визначає можливості даного ПЛК для створення прикладного програмного забезпечення.

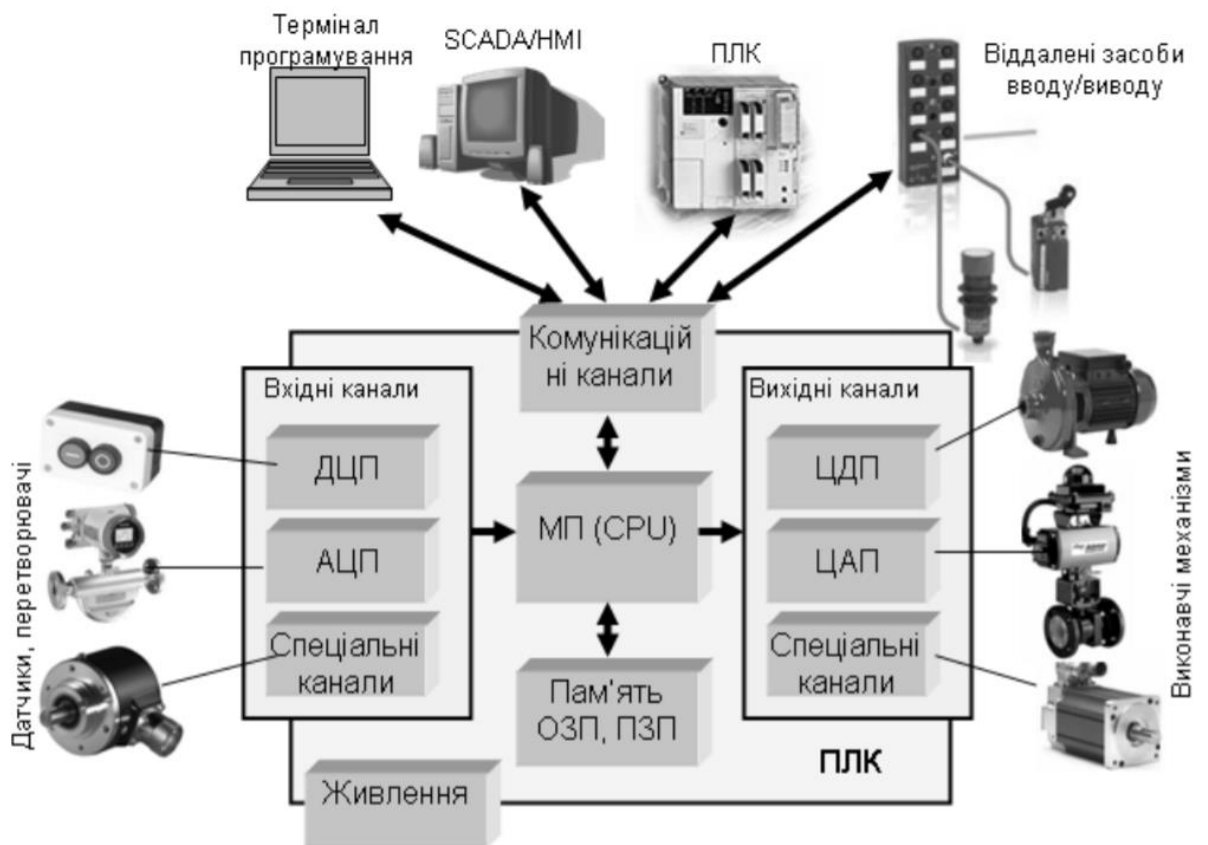


Рис. 1.1. Фізична структура промислового контролера

У ПЛК використовуються всі види пам'яті – ОЗП (оперативна, тобто енергозалежна), ПЗП (постійна, енергонезалежна) і ППЗП (перпрограмована, енергонезалежна з можливістю запису). Вони можуть вбудовуватись у процесорний модуль, і/або входити до складу контролера у вигляді окремих модулів або карт пам'яті. Останнє дає можливість так сформувати пам'ять контролера, щоб за характеристиками і ємністю вона найбільш відповідала прикладній задачі керування.

Функціональні і технічні можливості мікропроцесорного модуля і модулів пам'яті визначають одну з основних характеристик ПЛК – його швидкодію, яка, як правило, вимірюється в швидкості оброблення 1 кілобайта (або кілослова) програми користувача або/і як час одноразового обслуговування всіх входів-виходів контролера, тобто робочого циклу ПЛК.

Промисловий контролер по своїй суті являється спеціалізованим комп'ютером. Однак він має спеціалізоване призначення - управління об'єктом автоматизації в реальному часі, що відображається на його програмній та технічній структурі. Для постійного слідкування за процесом та керування ним він повинен отримувати інформацію від різноманітних датчиків і, згідно із заданим алгоритмом, виробляти сигнали для управління виконавчими механізмами. Для реалізації цього завдання до складу ПЛК входять різноманітні

канали (часто в складі модулів або блоків), які забезпечують його зв'язок з датчиками і виконавчими механізмами. Враховуючи що ПЛК працює з цифровою формою сигналу, ряд каналів забезпечують можливість його роботи з іншими формами сигналів. Основними завданнями каналів є:

– **вхідних**: перетворення фізичних сигналів від датчиків в цифрову форму, зрозумілу для мікропроцесорного модуля;

– **вихідних**: перетворення цифрової форми управляючих сигналів, вироблених мікропроцесорним модулем, на фізичні сигнали, що можуть керувати виконавчими механізмами;

– **комунікаційних**: забезпечити зв'язок з іншими мікропроцесорними (інтелектуальними) засобами;

Конструктивно контролери мають блоково-модульний принцип (крім моноблочних контролерів), тобто вони є проектно-компонованими виробами. Це значить, що типи модулів і їхня кількість залежать від особливості об'єкта автоматизації і алгоритму керування ним. До складу сучасних ПЛК може входити велика кількість різноманітних каналів, як правило, в складі модулів або блоків. Ці модулі умовно можна поділити на декілька основних груп, в залежності від того, які типи датчиків і виконавчих механізмів можуть бути підключені до нього. Нижче коротко перерахуємо їх властивості.

Вхідні канали (модулі) – це канали, які забезпечують контролер необхідною інформацією з об'єкту, тобто до них підключаються різноманітні датчики (напряму або через перетворювач), контакти кнопок, контакторів тощо. До вхідних каналів відносять:

- канали (модулі) дискретно-цифрового перетворення (ДЦП), до яких підключаються датчики з дискретним виходом (сигналізатори рівня, електроконтактні манометри та ін.) та управляючі технічні засоби (кнопки, перемикачі, датчики положення та ін.);

- канали (модулі) аналогового-цифрового перетворення (АЦП), до яких підключаються датчики з аналоговим уніфікованим електричним виходом, термометри опору, термомпари;

- спеціалізовані вхідні канали (модулі), до яких підключаються специфічні не уніфіковані вхідні сигнали, наприклад, модулі для підключення тензометричних датчиків вимірювання ваги; модулі лічильників; модулі безпеки; кодери і енкодери тощо.

Вихідні канали (модулі) – це канали, які забезпечують перетворення керуючого сигналу цифрової форми, що формується програмою, в необхідний рівень сигналу для виконавчого механізму. До вихідних каналів відносять:

– канали (модулі) цифро-дискретного перетворення (ЦДП), до яких підключаються дискретні виконавчі механізми і засоби комутації (магнітні пускачі, сигнальні лампи та ін.), а також електричні виконавчі механізми з постійною швидкістю обертання (типу МЕО, МЕК та ін.);

– канали (модулі) цифро-аналогового перетворення (ЦАП), до яких підключаються аналогові виконавчі механізми, для управління якими використовуються уніфіковані електричні сигнали.

– спеціалізовані вихідні канали (модулі), до яких підключаються специфічні вихідні сигнали, наприклад, модулі для керування кроковими двигунами, модулі безпеки та ін.

При модульній або блочній побудові, до одного модуля можуть входити канали різного типу. Такі модулі (або блоки) прийнято називати змішаними (mixed).

Крім вхідних та вихідних каналів ПЛК має комунікаційні канали (модулі), які забезпечують підключення ПЛК до різноманітних промислових мереж. Сучасний контролер завжди має як мінімум один комунікаційний канал, щоб мати можливість підключити до ПЛК термінал програмування. На сьогоднішній день в якості терміналів програмування використовуються комп'ютери зі встановленим спеціальним програмним забезпеченням – середовищем програмування ПЛК.

1.3. Типи ПЛК

У світі випускається велика кількість промислових мікропроцесорних контролерів серед яких особливої популярності в системах автоматизації виробництв України набули ПЛК фірм Siemens, Schneider Electric, Mitsubishi Electric, VIPA та ін.

За конструктивним виконанням всі ПЛК можна поділити на декілька груп. Для завдань автоматичного управління з невеликою кількістю входів-виходів випускаються моноблочні ПЛК, які також називають компактними. Вони являють собою функціонально закінчений технічний засіб, у якому розташовані всі функціональні модулі: мікропроцесор, усі види пам'яті, блок живлення, комунікаційні, вхідні та вихідні канали. Основна перевага таких контролерів – їхня низька ціна. Основний недолік – неможливість розширювати кількість

каналів входів-виходів. Для забезпечення можливості вибору оптимальної структури такого типу контролерів випускається серія таких контролерів, які мають різну кількість і співвідношення типів каналів входів виходів. На рис. 1.2 показано моноблочний контролер Twido Compact фірми Schneider Electric (Франція).



Рис. 1.2. Приклад ПЛК у моноблочному виконанні

Для систем із середньою кількістю каналів (порядку десятків) можливою альтернативою моноблочним контролерам можуть бути компактні контролери з можливістю нарощення додатковими модулями (рис. 1.3). У такому конструктиві основу ПЛК становить базовий модуль, який функціонально являє собою повністю закінчений контролер, подібно до компактного. Особливістю його виконання є можливість нарощення додатковими модулями. Таким чином він більш універсальний і гнучкіший, так як має властивості компактного і модульного ПЛК. На рис. 1.3 показано компактний контролер типу FX фірми Mitsubishi Electric (Японія) з установленими додатковими модулями.



Рис. 1.3. Приклад компактного ПЛК з можливістю нарощення додатковими модулями

Для систем автоматизації технологічними процесами та великими енергетичними об'єктами частіше використовуються модульні ПЛК, кількість і типи модулів у яких вибираються залежно від кількості і типів вхідних/вихідних сигналів. Це і є основною перевагою модульних контролерів перед нерозширюваними компактними. Такі контролери призначені для побудови систем управління, кількість датчиків та виконавчих механізмів у яких вимірюється десятками та сотнями. Для ПЛК такого типу базовими складовими для забезпечення їхньої працездатності є модуль живлення та процесорний модуль (їх інколи суміщують в єдиному корпусі). Всі інші модулі набираються залежно від поставленого завдання. На рис. 1.4 показано модульний контролер M340 фірми Schneider Electric (Франція).



Рис. 1.4. Приклад ПЛК модульного типу

До складу модульного ПЛК може входити велика кількість модулів (порядку ста), з одним процесорним модулем у якості керуючого пристрою. Такий багатоканальний контролер може керувати технологічними процесами всього виробництва. Але недоліком такої системи є те, що в разі виходу з ладу мікропроцесорного модуля втрачається управління всім виробництвом. Тому як варіант ставлять модель з резервуванням процесорного модуля. Інший підхід – це створення розподіленої системи, в якій на кожній технологічній ділянці (відділенні) встановлюється свій окремий ПЛК для управління технологічними процесами цієї частини виробництва. Для координації роботи ПЛК їх об'єднують промисловими мережами в єдину інтегровану систему управління. Використання розподіленого принципу управління та модульної структури дає змогу вибрати і конфігурувати контролери, для якого буде мінімізований показник функціональні можливості/вартість ПЛК.

Останнім часом великої популярності набули структури систем управління на базі контролерів з розподіленими (віддаленими) засобами вводу/виводу (рис. 1.5). У таких системах до промислової мережі ПЛК можуть підключатися:

- віддалені модулі вводу/виводу (модулі I/O); - перетворювачі частоти (ПЧ), які призначені для управління частотою обертів асинхронного двигуна;
- сервоприводи;
- інтелектуальні контактори (магнітні пускачі);
- інтелектуальні датчики та виконавчі механізми.

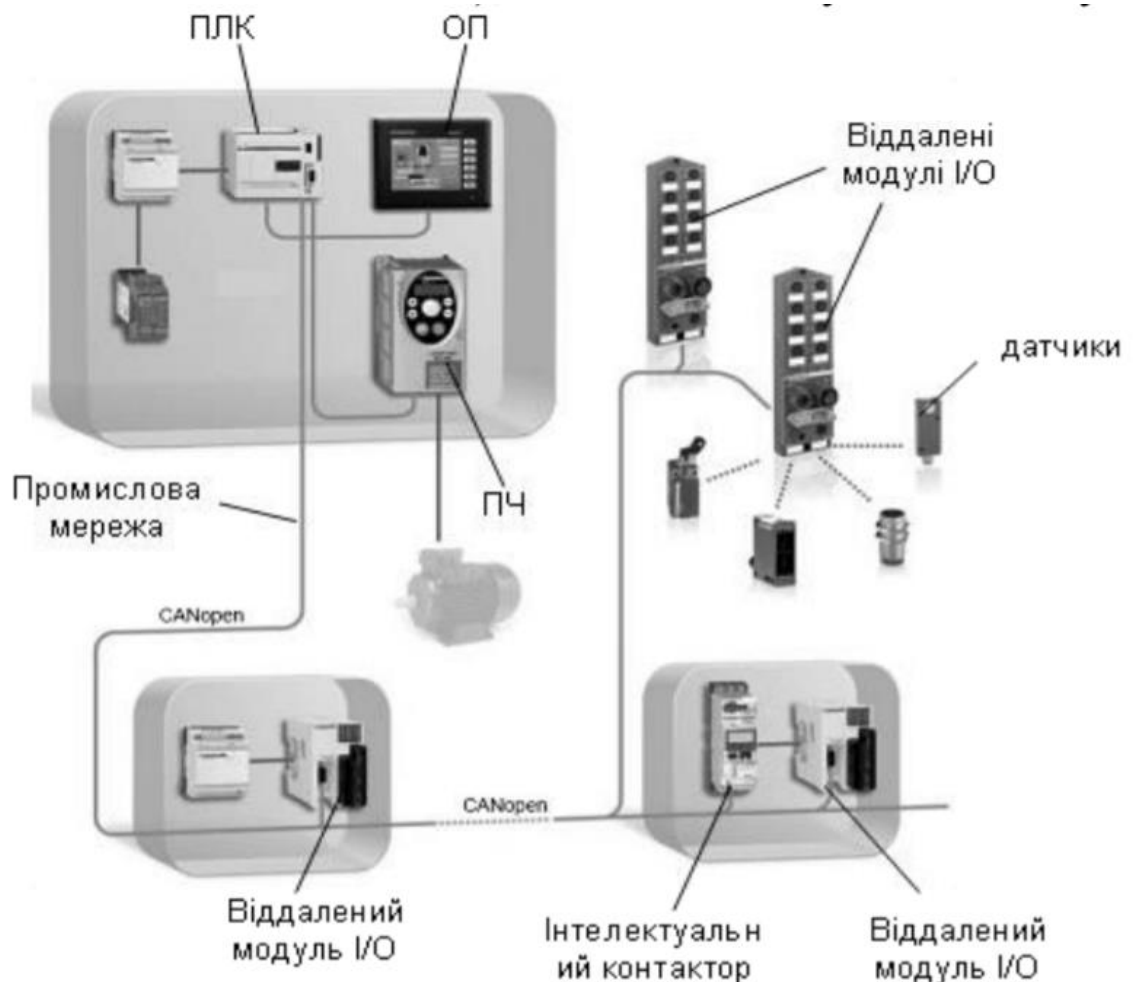


Рис. 1.5. Структура ПЛК з віддаленими модулями

З точки зору функціональності, всі засоби польового рівня, які підключені по промисловій мережі до ПЛК, являються його модулями вводу/виводу. На відміну від локальних модулів вводу/виводу модульного ПЛК (тобто які розміщені на його шасі), віддалені модулі I/O можуть знаходитись у місці розташування датчиків, тобто на порівняно великій відстані від базового ПЛК, що значно зменшує кількість кабелів і спрощує монтаж обладнання. Крім того, застосування стандартних промислових мереж дає можливість використовувати віддалені модулі вводу/виводу та периферії розроблених різними виробниками, що неможливо для локальних модулів ПЛК. На рис. 1.5 показано структуру

розподіленої системи управління, в якій до ПЛК за допомогою промислової мережі (CANopen) під'єднуються віддалені модулі входів-виходів (I/O). До них в свою чергу підключаються датчики і виконавчі механізми, перетворювачі частоти (ПЧ) та інтелектуальні контактори. Ці віддалені модулі I/O можуть знаходитись на відстані в декілька сотень метрів від ПЛК, до якого вони підключені.

2 Лабораторна установка

В даній лабораторній роботі лабораторна установка складається з: персонального комп'ютера, програмного забезпечення Arduino IDE, плати Arduino UNO, макетної плати, кнопок, світлодіодів, резисторів, провідників. В разі відсутності фізичних плати та інших компонентів скористатися сервісом <https://www.tinkercad.com/>.

3 Хід виконання роботи

3.1 Зібрати схему зображену на рис 3.1.

3.2 Запустіть ПЗ Arduino IDE.

3.3 Написати програму яка буде керувати включенням світлодіодів через термінал.

3.4 Розширити функціонал програми таким чином, щоб можна було присвоювати кожному світлодіоду свою кнопку. І при натисканні на кнопку засвічувався відповідний світлодіод.

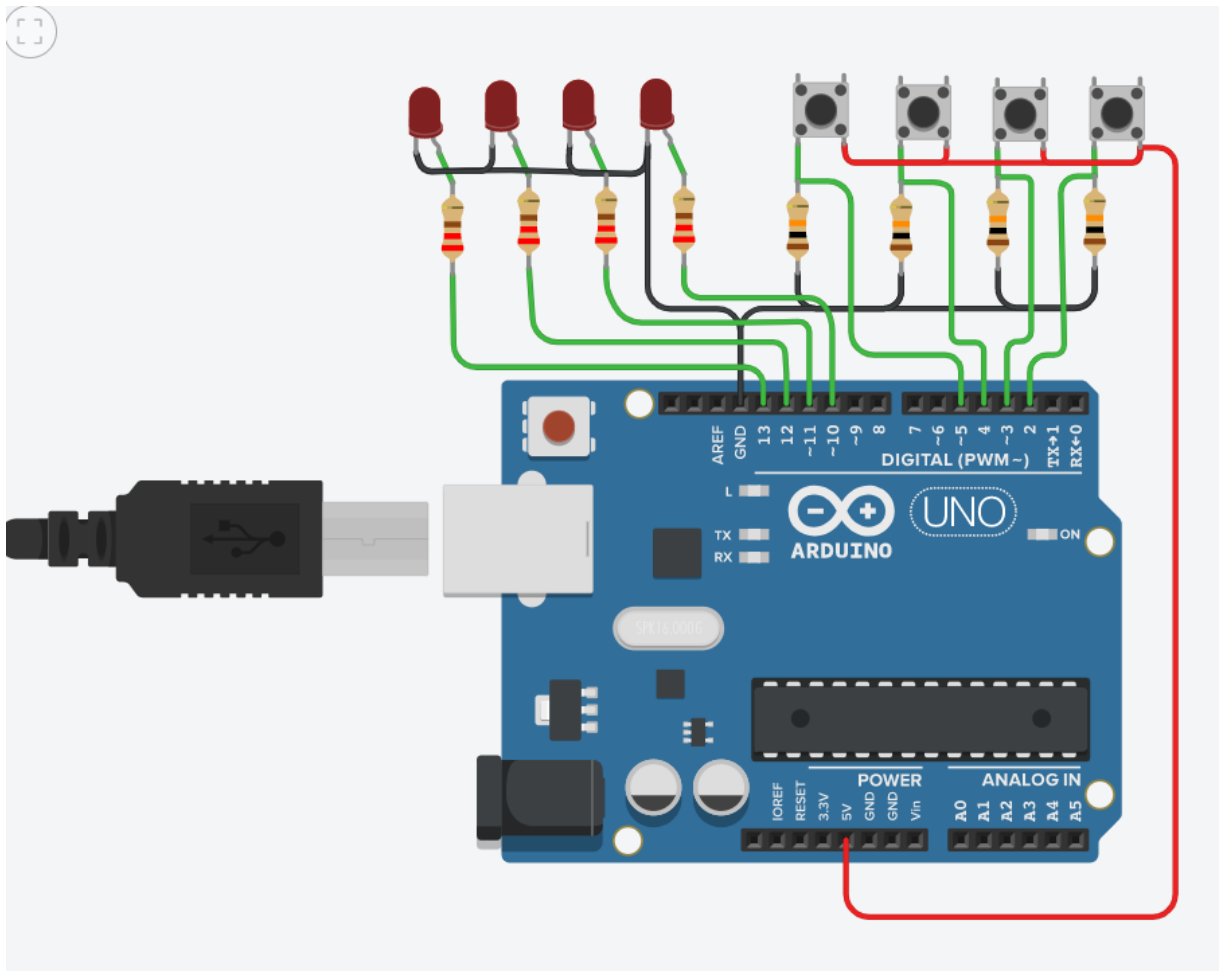


Рис.3.1 – Схема установки

3.5 Виконання присвоєння виконується командою в терміналі, наприклад команда яка присвоїть другу кнопку 3-му світлодіоду буде виглядати так : « B2=L3» , першу кнопку присвоїти першому світлодіоду буде виглядати так: «B1=L1» і т.д.

4 Розрахункове завдання

Розрахункове завдання – див. п. 3.2, п.3.3, п.3.4 та 3.5.

5 Вимоги до звіту

Звіт з лабораторної роботи повинен містити:

1. Коротке описання мети і методики проведення роботи.
2. Перелік використаних приладів та матеріалів.
3. Розрахункове завдання.
4. Висновки.

6 Контрольні питання

1. Що таке ПЛК?
2. Де використовуються ПЛК?
3. Які ви знаєте види каналів в ПЛК?
4. До яких каналів підключають електричні мотори?
5. Яка можлива максимальна напруга на вході ПЛК?
6. Чи може бути ПЛК без зовнішніх інтерфейсів?
7. Скільки входів у ПЛК?
8. До яких каналів підключають енкодер?