**Лабораторна робота №3.**

**Тема**: **Програмування мікропроцесорних систем.** **Робота з аналоговими датчиками**

**Мета:** ознайомити студентів методам перетворення аналогових величин в цифрові значення, які можуть бути проаналізовані мікроконтролером Arduino.

**Короткі теоретичні відомості та методичні вказівки по виконанню роботи.**

Дані про навколишній світ всі пристрої неминуче отримують в аналоговому вигляді. Цифрова інформація являє собою серію бінарних (цифрових) даних. Кожний біт приймає тільки одне з двох можливих значень. Графіки на рис. 3.1 показують, чим відрізняються один від одного аналогові і цифрові сигнали. Зліва прямокутні імпульси, амплітуда яких бере тільки два значення: 0 і 5 вольт. Точно так само, як з кнопкою з попередньої роботи: тільки HIGH або LOW. Справа зображений фрагмент синусоїдального сигналу. Незважаючи на те, що його амплітуда знаходиться в тих же межах (0 і 5 вольт), аналоговий сигнал приймає нескінченне число значень між цими двома значеннями 0 та 5.

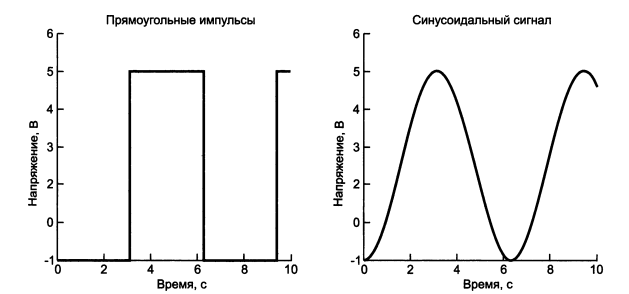


Рис.3.1. Цифрові та аналогові сигнали.

Аналогові сигнали не можна уявити кінцевим числом станів, теоретично вони можуть мати нескінченне число значень в межах деякого діапазону. Припустимо, сонячне світло-це аналоговий сигнал, який потрібно виміряти. Природно, є розумний діапазон, в межах якого змінюється освітленість (вимірюється в люксах - світловому потоці на одиницю площі). можна обґрунтовано очікувати значення показань між 0 люкс (для абсолютно чорного) і 130 000 люкс на прямому сонячному світлі. Якби вимірювальний прилад був абсолютно точний, то можна отримати нескінченне число значень в даному діапазону. Комп'ютерна система ніколи не може оперувати з нескінченним числом десятинних розрядів для аналогового значення, тому що обсяг пам'яті і продуктивність комп'ютера обмежені. Як же тоді з'єднати інтерфейс цифрового контролера Arduino з аналоговим реальним світом? Це робить аналого-цифровий перетворювач (АЦП), який перетворює аналогові значення в цифрові із заданою точністю.

Перетворення аналогового сигналу в цифровий.

Припустимо, що ви хочете виміряти освітленість в своїй кімнаті. Хороший світлочутливий датчик видає вихідну напругу, яка залежить від освітленості кімнати. Коли в приміщенні абсолютно темно, пристрій видав б 0 В, а при максимальній освітленості - 5 В. Проміжні значення пропорційні відповідній освітленості. Але як ці значення розпізнає плата Arduino, щоб дізнатися, наскільки світло в кімнаті? Перетворити аналогові значення напруги в числа, які може обробляти контролер, дозволяє аналого-цифровий перетворювач Arduino.

Точність АЦП залежить від його розрядності. На платі Arduino Uno встановлений 10-розрядний АЦП. Це означає, що АЦП може розділити аналоговий сигнал на 210 різних значень. Отже, Arduino може присвоїти 210 = 1024 аналогових значень, від 0 до 1023.

Опорна напруга визначає максимальне напруження на вході АЦП, його значення відповідає коду 1023. При нульовій вхідній напрузі АЦП видає на виході 0, при вхідній напрузі 2,5 В на виході буде значення 512 (половина від 1023), при вхідній напрузі 5 В вихідний код дорівнює 1023. Щоб краще зрозуміти це, подивіться на графіки для трьохрозрядного АЦП, зображені на Рис. 3.2. В принципі, опорна напруга АЦП можна змінити, але в наших пристроях опорним буде напруга 5 В.

A graph and diagram of a graph

Description automatically generated

Рис.3.2. Трьохрозрядне аналого-цифрове перетворення.

У трьохрозрядного АЦП 3 біта розділення, оскільки 23 = 8, отже, у нього є 8 рівнів, від 0 до 7. Будь-якому аналоговому значенню, яке надходить на вхід такого АЦП, на виході відповідає код від 0 до 7. На рис. 3.2 показано, що рівні вхідної напруги перетворюються у вихідні дискретні цифрові коди, з якими може оперувати мікроконтролер. Чим вище розрядність, тим більше рівнів, які доступні для подання кожного значення. Як згадувалося, у Arduino Uno АЦП має 1024 ступені, а не 8, як на рис. 3.2.

У різних плат Arduino різне число аналогових контактів, на які подаються аналогові сигнали і які АЦП перетворюються у відповідний цифровий код. Для читання аналогових значень передбачена функція analogRead ().

**Зауваження.**

Аналогових виходів в Arduino немає!

**Читання даних з потенціометра.**

Найпростіший аналоговий датчик, з якого можна отримати аналоговий сигнал - це потенціометр. Їх використовують в стереосистемах, звукових колонках, термостатах і в інших виробах. Потенціометри діють як регульовані дільники напруги і забезпечені ручкою-регулятором. Вони бувають різних розмірів і форм, але завжди мають три виходи. Підключіть один крайній вивід потенціометра до землі, а інший до шини 5 В. Потенціометри симетричні, так що не має значення, з якого боку ви підключите шину напруги, а з якою землю. Середній вивід з'єднайте з аналоговим контактом 0 на платі Arduino. Як правильно підключити потенціометр до Arduino,показано на рис. 3.3. При повороті ручки потенціометра аналоговий вхідний сигнал буде плавно змінюватися від 0 до 5 В.

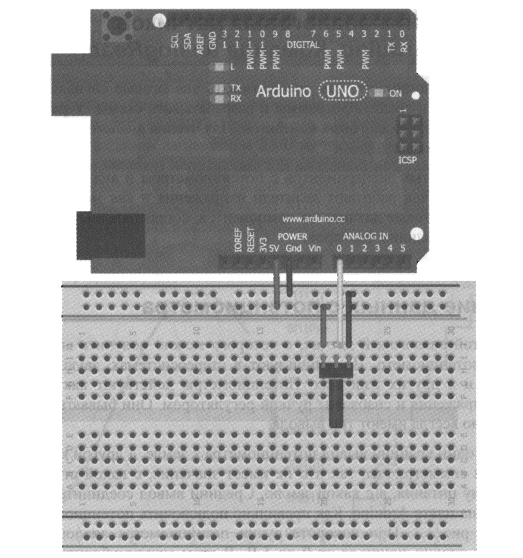
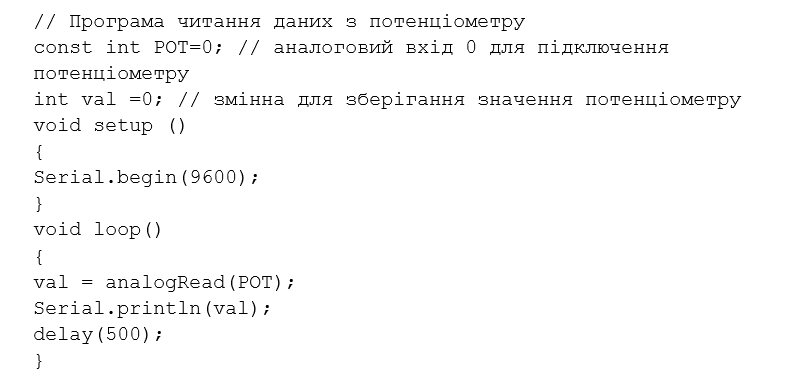


Рис.3.3. Підключення потенціометру.

Перш ніж використовувати потенціометр для управління іншим обладнанням, подивимося, як зчитати значення опору потенціометра за допомогою АЦП і передати через послідовний порт Arduino для перегляду значень на комп'ютері. Для читання значення аналогового входу передбачена функція analogRead (), для виведення значень в послідовний порт Arduino IDE - функція serial.println(). Наберіть і завантажте в плату Arduino програму з лістингу 3.1.

**Лістинг 3.1. Програма читання даних з потенціометру**



**Коментар до лістингу.**

Для роботи з виведенням інформації у послідовний порт комп’ютеру, спочатку необхідно ініціювати послідовне з'єднання, викликавши функцію serial .begin (), єдиний аргумент якої задає швидкість передачі даних в бодах, тобто швидкість передачі даних визначає кількість бітів, що передаються в секунду (біт/сек=бод). Висока швидкість передачі створює враження більшого числа даних за менший час, але може привести до помилок в деяких системах зв'язку. У наших прикладах вибрана швидкість 9600 бод.

У кожній ітерації циклу змінна val отримує аналогове значення, зчитане командою anailogRead () з входу, сполученого із середнім контактом потенціометру (в нашому випадку це вхід А0). Далі це значення функція serial.println(val) виводить в послідовний порт, з'єднаний з комп'ютером. Потім йде затримка в пів секунди (щоб числа виводилися не швидше, ніж ви можете їх прочитати).

Після завантаження на плату Arduino ви помітите, що світлодіод тх, розташований на платі, блимає кожні 500 мс (так повинно бути). Цей індикатор показує, що плата Arduino передає дані через послідовний USB-інтерфейс на комп'ютер. Для перегляду даних підійдуть будь-які термінальні програми, але в Arduino IDE є вбудований монітор послідовного порту, для запуску якого натисніть кнопку, обведену кружком на рис. 3.4.

A screenshot of a computer

Description automatically generated

Рис.3.4. Кнопка запуску монітору послідовного порту.

Після запуску монітора послідовного порту на екрані комп'ютера з’являється вікно з відображенням потоку переданих чисел. Поверніть ручку потенціометру, і ви побачите, що виводяться значення змінюються. Якщо повернути ручку в одному напрямку, числа починають наближатися до 0, якщо в іншому - до 1023. Приклад відображення даних показаний на рис. 3.5.

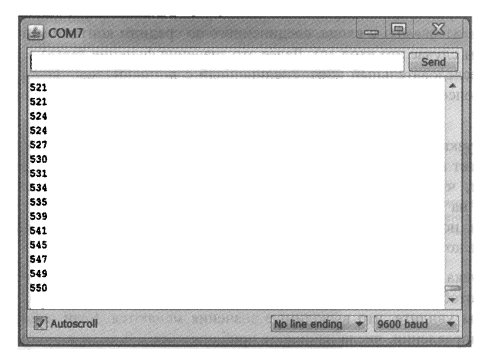


Рис.3.5. Виведення до послідовного порту.

**ПРИМІТКА.**

Якщо виводяться незрозумілі символи, переконайтеся, що швидкість передачі даних установлена правильно. У програмі порт ініціалізований на швидкість 9600 бод, таке ж значення необхідно встановити в настройках монітору послідовного порту.

Використання аналогових датчиків.

Хоча в контакті потенціометра можна отримати аналогове значення напруги, він насправді не є датчиком в традиційному сенсі. Потенціометри "відчувають" лише поворот ручки, але це не дуже цікаво. Але є реальні датчики, що видають значення на аналоговому виході, відповідно реальній дії. Приклади датчиків:

♦ акселерометри для виявлення нахилу (застосовуються в смартфонах і планшетах);

♦ магнітометри для фіксації магнітних полів (необхідні при створенні цифрових компасів);

♦ інфрачервоні датчики для визначення відстані до об'єкта;

♦ датчики для вимірювання температури.

Багато з цих датчиків підключають аналогічно потенціометру: два контакту живлення (VCC=5В і GND) і один до аналогового входу плати Arduino. Для наступного експерименту вибираємо датчик температури LM35DZ.

Відповідний датчик змінює свій опір в залежності від температури і розрахований на діапазон (00,5000). Напруга, що знімається з виходу датчику дорівнює 0, якщо температура 00, та максимальна, тобто 5В, коли температура дорівнює 5000. Датчик температури LM35DZ дозволяє легко перетворити вихідний рівень напруги на показники температури в градуси Цельсія. Кожні 10 мВ вихідної напруги відповідають 1 ° С. Наприклад, якщо напруга 315 мВ, то відповідна температура 31,5С0. Після того, як відповідна напруга входить до аналогового входу Arduino і перетворюється у цифровий сигнал, формула для перетворення цифрового сигналу (val ) в температуру (в С °) виглядає так: T= ( val/1023.0)\*500.

Позначення на схемі A close-up of a diagram

Description automatically generated

Зверніть увагу на несиметричність датчику, а саме, що потрібно з єднати з +5В, що з землею. Середній вихід – до контакту Arduino.

A circuit board with wires

Description automatically generated

Рис.3.6. Схема підключення датчика температури до аналогового входу 0.

Програма виведення значень температурного датчика до послідовного порту.

**Код програми 3.2.**

A screenshot of a computer program

Description automatically generated

**Коментар до лістингу.**

1. Закоментовано виведення температури у градусах по Фаренгейту.
2. Функція Serial.print() аналогічна Serial.println(). Різниця в тому, що Serial.print() «залишається» на тому рядку, де було виведення інформації, а Serial.println() переводить чергове виведення інформації на наступний рядок.
3. Датчик, м’яко кажучи, не дуже точний. Тому є сенс подивитися на значення, що він видає і зробити потрібні корективи у програмі, спираючись на данні, що видає комп’ютер та на реальну температуру у приміщені. Вважаємо, що температура при напрузі 0В відповідає 00С.

Використання змінних резисторів для створення власних аналогових датчиків.

Завдяки досягненням в галузі фізики, ми маємо безліч матеріалів, які можуть змінювати опір в результаті фізичного впливу. Наприклад, проводячі фарби, що змінюють свій опір при вигині і скручуванні, напівпровідники змінюють опір під дією світла (фоторезистори), опір деяких матеріалів залежить від нагрівання і охолоджування - (термістори).

Це всього лише кілька прикладів, які дозволять створити свої власні аналогові датчики. Оскільки згадані датчики змінюють опір, а не напругу, в схемі потрібно створити дільник напруги, щоб можна було виміряти зміну опору.

Резистивний дільник напруги.

Резистивний дільник напруги складається з двох резисторів, від співвідношення опорів яких залежить вихідна напруга. Так, якщо один з резисторів змінний, то на виході можна отримати зміну напругу. Інший резистор визначає чутливість схеми, якщо це підлаштовувальний резистор, то чутливість можна коригувати.

Розглянемо нерегульований резистивний дільник (рис. 3.7) і напругу на його виході. Позначення А0 на рис. 3.9 - це аналоговий вхід А0 на платі Arduino. Залежність вихідної напруги дільника від вхідного:



У нашому випадку на вхід дільника подано напругу 5 В, а вихід підключений до аналогового контакту А0 плати Arduino. Якщо R1 і R2 однакові (як, наприклад, 10 кОм), то 5 В ділиться навпіл, і на аналоговому вході буде 2,5В. Перевірте це, підставивши значення в формулу:



A diagram of a circuit

Description automatically generated

Рис. 3.7. Простий дільник напруги.



Рис.3.8. Фоторезистор.

Тепер припустимо, що один з цих резисторів змінний, наприклад фоторезистор (рис. 3.8). Опір фоторезистору залежить від інтенсивності падаючого на нього світла. Використовується фоторезистор з номінальним опором 200 кОм. У повній темряві його опір близько 200 кОм, при яскравому світлі воно падає майже до нуля. Від того, який резистор (R1 або R2) поміняти на фоторезистори, і від номіналу постійного резистора буде залежати масштаб і точність показань. Спробуйте проекспериментувати з різними конфігураціями і подивіться через монітор послідовного порту, як змінюються показання. Як приклад замінимо R1 на фоторезистор, a R2 візьмемо постійним з номіналом 10ком (рис. 3.9). Для даної вправи можна залишити на платі RGB-світлодіод і підключити його як одноколірний.

A circuit board with wires

Description automatically generated

Рис.3.9. Підключення фоторезистору.

Завантажте програму зчитування аналогових даних і видачі результату в послідовний порт (див. лістинг 3.1) і поміняйте освітленість фоторезистору. Ви не зможете отримати весь діапазон значень від 0 до 1023, тому що у фоторезистору ніколи не буде нульового опору. В результаті ви визначите мінімальне і максимальне значення напруги на виході. Ці дані потрібні, щоб зробити "інтелектуальний" нічник, який буде світити більш яскраво в темному приміщенні, і навпаки. Виберіть аналогові значення для вашої кімнати, відповідні темряві і максимальної освітленості. У мене це були значення 200 (темрява) і 900 (максимальне освітлення). У вас можуть бути інші цифри. Вони залежать від умов освітлення, значення резистора R2 і характеристик фоторезистору.

**Управління аналоговими виходами по сигналу від аналогових входів.**

Нагадаємо, що функція analogWrite () дозволяє змінювати яскравість світлодіода, але не забувайте, аргумент цієї функції 8-розрядний, тобто знаходиться в діапазоні від 0 до 255, в той час як АЦП видає значення від 0 до 1023. У мові програмування Arduino є зручні функції для пропорційного перетворення значення від одного діапазону до іншого: map () и constrain (). Синтаксис функції map () виглядає наступним чином:

output = map (value, fromLow, fromHigh, toLow, toHigh).

Тут:

* value - значення, що потрібно перетворити.
* fromLow і fromHigh - це нижня і верхня межі поточного діапазону в яких лежить змінна value.
* toLow і toHigh- нижня і верхня межі нового діапазону, в якому буде лежати змінна output.

У нашому прикладі value - це мінімальна та максимальна освітленість в приміщенні (200 і 900). Аргумент функції analogwrite () повинен бути в діапазоні від 0 до 255. Але ми хочемо, щоб меншій освітленості відповідала велика яскравість світлодіода, т. е. мінімальним значенням на аналоговому вході повинні відповідати максимальні значення на виводах світлодіоду. Зручно, що функція map () робить це автоматично. Функція map () здійснює лінійне відображення. Наприклад, якщо fromLow і fromHigh рівні 200 і 900, відповідно, a toLow і toHigh рівні 255 і 0, то200 перейде у 255, 900 у 0, а 550 перетвориться в 127, тому що 550 знаходиться посередині між 200 і 900, а 127 посередині між 255 і 0.

Слід врахувати, що функція map () не обмежує значення, якщо вони виходять за межі діапазону. Якщо value виявиться менше 200 (для нашого прикладу), то output буде більше 255. Це незручно, тому передати функції analogwrite () значення, що перевищує 255, не можна. Для обмеження значень є функція constrain (), синтаксис якої виглядає наступним чином:

output = constrain (value, min, max).

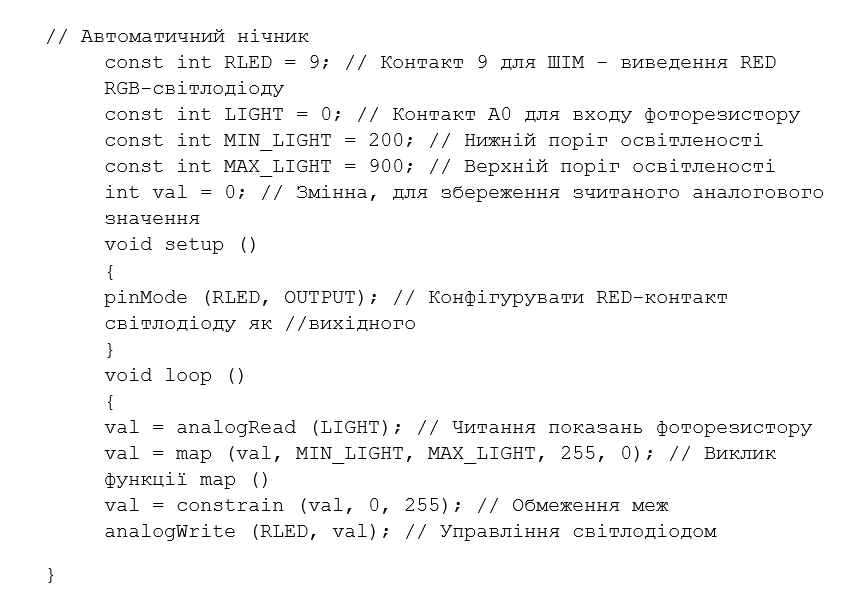
При передачі значення з функції map () в функцію constrain () можна встановити аргумент min рівним 0 і max 255, тоді величини, що виходять за рамки цього діапазону, будуть обмежені. Тобто

output = min, якщо value < min

output = max, якщо value > max

Тепер все готово, щоб написати програму керованого нічника. Подивимося, як буде виглядати остаточно наш проект

**Лістинг 3.3.**



Зверніть увагу, що в лістингу змінна val використовується повторно. В принципі, можна задати й іншу змінну. У таких функціях, як map (), попереднє значення змінної val служить в якості аргументу і після завершення виконання функції перезаписується заново.

Завантажте програму в плату Arduino і подивіться, чи працює нічник, як очікувалося. Чутливість нічника можна відрегулювати, підібравши мінімальну і максимальну межі комфортного діапазону за допомогою монітора послідовного порту.

**Завдання по роботі.  
В кожній з робіт потрібно розробити схеми та заставити її працювати за правилами, що викладені в задачах. До кожної із задач у звіті повинні бути намальовані відповідні схеми! Всюди n – номер варіанту. Тексти програм записати у зошит.**

**Усі завдання виконувати з допомогою інтернет-сервісу Tinkercad**

**(n – номер варіанта).**

**Задача №1. Робота з потенціометром**.

Створити проект (електронну схему і написати програмний код) в якому при повороті ручки потенціометра, якщо його значення знаходиться в діапазоні від n\*20 до n\*40 загорявся б червоний світлодіод. Значення потенціометра має постійно видаватися в монітор послідовного порту. При введенні в моніторі послідовного порту числа n повинен включатися режим миготіння світлодіоду в цьому діапазоні.

**Задача №2. Робота з датчиком температури.**

Створити проект (електронну схему і написати програмний код) електронного термометра, який видавав би значення температури в град. Цельсія і Фарінгейта в монітор послідовного порту через проміжок часу рівний n\*50 (задавши його за допомогою функції millis ()).

**Задача №3. Робота з фоторезистором.**

Створити проект (електронну схему та написати програмний код) світильника, що складається з 3-х білих світлодіодів. Він повинен працювати у двох режимах: 1- ступеневий та 2 -плавний (режим задається у моніторі послідовного порту).

1. Світильник працює на трьох рівнях (low=(350-n\*2), middle, high=(700+n\*4)). Якщо освітленість менше low - горять усі 3 світлодіоди, якщо від low до middle - 2, якщо від middle до high – 1, якщо більше high – світлодіоди вимкнено.

2. Світильник працює плавно: при зменшенні освітленості збільшується яскравість усіх світлодіодів, а при збільшенні - зменшується (за допомогою ШІМ).

Використайте функції **map()**, та **constrain().**

Кнопка керує світильником (вкл./викл.).

Інформація про освітленість видається у моніторі послідовного порту.

**Питання по роботі.**

1. Чим відрізняються аналогові та цифрові сигнали?
2. Що означає термін «оцифровування» аналогового сигналу?
3. З якими числами «спілкуються» аналогові входи Arduino?
4. Чи є в Arduino аналогові виходи? Як імітується в Arduino аналоговий вихід?
5. Що таке потенціометр? Як підключити потенціометр в Arduino ?
6. Як програма на Arduino спілкується з комп’ютером через послідовний порт?
7. Для чого потрібна функція Serial.begin()? Що за аргумент вона використовує?
8. Що таке бод?
9. Для чого потрібна функція Serial.print()?Що за аргумент вона використовує?
10. Яка різниця у функціях Serial.print() та Serial.println()?
11. Що таке резистивний дільник напруги?
12. Як використовується резистивний дільник напруги при роботі із фоторезистором?
13. Навіщо потрібна функція map()? Як вона працює?
14. Навіщо потрібна функція constrain ()? Як вона працює?