

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

МЕТОДИЧНІ ВКАЗІВКИ
до лабораторного практикуму та самостійної роботи
з дисципліни
КОМП'ЮТЕРНІ СИСТЕМИ ВИМІРЮВАНЬ
для студентів спеціальності
152 „Метрологія та інформаційно-вимірювальна техніка“
денної й заочної форм навчання

2020

Методичні вказівки до лабораторного практикуму та самостійної роботи з дисципліни «Комп'ютерні системи вимірювань» для студентів спеціальності 152 „Метрологія та інформаційно-вимірювальна техніка“ денної й заочної форм навчання / Укл.: Г.В. Сніжної. – Запоріжжя: НУ «Запорізька політехніка», 2020. – 52 с.

Укладач: Г.В. Сніжної, проф., доктор техн. наук

Рецензент: В.В. Погосов, проф., д-р фіз.-матем. наук

Відповідальний за випуск: А.В. Коротун, доц., канд.фіз.-мат.наук

Затверджено
на засіданні кафедри
мікро- та наноелектроніки

Протокол №5
від “ 20 “ жовтня 2020 р.

Рекомендовано до видання
НМК ФРЕТ
Протокол №3
від “ 22 “ жовтня 2020 р.

ЗМІСТ

Вступ	5
Підготовка до проведення лабораторної роботи та оформлення звіту	6
1 Лабораторна робота №1 “Можливості пристроїв на базі мікроконтролера ESP8266 (плата NodeMCU) для застосування у комп'ютерних системах вимірювання”	7
1.1 Короткі теоретичні відомості	7
1.1.1 Загальні відомості про NodeMCU (ESP8266)	7
1.1.2 Середовище Arduino IDE	15
1.1.3 Програмування на мові Arduino IDE	16
1.1.4 Програмування ESP8266	17
1.2 Хід роботи	17
1.2.1 Підключення плати NodeMCU до комп'ютера	17
1.2.2 Установка мікроконтролера та підключення елементів на макетній платі	18
1.2.3 Підключення миготливого світлодіоду до платформи NodeMCU	21
1.3 Оцінювання і контроль	23
1.4 Контрольні запитання	23
2 Лабораторна робота №2 “Підключення плат Arduino і NodeMCU до комп'ютера та вимірювання напруги”	24
2.1 Короткі теоретичні відомості	24
2.1.1 Комунікаційні порти персонального комп'ютера	24
2.1.2 Принцип роботи COM-порту. Відправка команд з ПК ..	26
2.2 Хід роботи	27
2.2.1 Підключення потенціометра через COM порт до ПК із залученням Arduino IDE	27
2.2.2 Підключення Arduino UNO до ПК за допомогою USB порта	29
2.2.3 Підключення NodeMCU до комп'ютера за допомогою USB порта	30
2.3 Контрольні запитання	30
3 Лабораторна робота №3 “Використання комп'ютера для вимірювання атмосферного тиску, температури і висоти над рівнем моря. Створення HTML-сторінки для відображення вимірюваної інформації”	31

3.1 Короткі теоретичні відомості	31
3.2 Хід роботи	32
3.2.1 Підключення датчика BMP280 для вимірювання атмосферного тиску, температури і висоти над рівнем моря .	32
3.2.2 Скетч	33
3.2.3 Створення HTML-сторінки	34
4 Лабораторна робота №4 „Передачу даних між датчиками і комп'ютером через Інтернет за допомогою WiFi-трансівера ESP8266”	36
4.1 Короткі теоретичні відомості	36
4.2 Хід роботи	40
4.2.1 Завантаження коду на ESP8266	40
4.2.2 Перехід ESP8266 в автономний режим	43
4.2.3 Підключення ESP8266 до інтернету	44
4.3 Контрольні запитання	44
5 Лабораторна робота №5 „Дослідження у мікроелектроніці за допомогою віртуальних вимірювальних приладів”	45
6 Питання для самоконтролю	49
Рекомендована література	50
Додаток А Приклад оформлення титульної сторінки	52

ВСТУП

Сучасна інформаційна система підприємства, неможлива без комплексу досить складної обробки інформації, отриманої не тільки шляхом ручного введення, а й на підставі автоматичних вимірювань. До обробки інформації можна віднести: автоматизований збір, перетворення, агрегування, зберігання, передачу і уявлення (візуалізація). Результатом цієї обробки є надання необхідної та достовірної інформації в потрібному вигляді, в потрібному місці, і в потрібний час, все це, без чого неможливе повноцінне управління. Тут мається на увазі не тільки інформація для підтримки прийняття рішень, що дозволяє операційного менеджменту виконувати свої функції, а й інформація, на підставі якої система в автоматичному або автоматизованому режимі проводить вироблення керуючих впливів і команд, що надходять на реальний об'єкт управління.

У порівнянні з інформаційними системами, в яких функції отримання первинної інформації та видачі керуючих впливів перекладаються на людину, системи, побудовані на основі реального зв'язку з об'єктом управління, мають більш високою оперативністю і достовірністю, не менш важливим є те, що в таких системах відсутній людський фактор. Апаратні та програмні засоби, що забезпечують технічну реалізацію інформаційно-вимірювальних і управляючих систем, повинні відповідати сучасним вимогам уніфікації та стандартизації, метрологічній, конструктивній, інформаційній та експлуатаційній сумісності, принципам модульної побудови. При виконанні цих вимог забезпечується надійність побудованої на основі цих коштів інформаційної системи підприємства, достовірність і оперативність одержуваної менеджментом інформації, гарантованість виконання командної інформації.

В результаті вивчення «Комп'ютерні системи вимірювань» дисципліни студент повинний знати:

- методів збору, обробки, зберігання та передачі вимірювальної інформації;
- способів побудови сучасних високопродуктивних вимірювальних систем;
- компонентів сучасних вимірювальних систем і методи їх проектування.

ПІДГОТОВКА ДО ПРОВЕДЕННЯ ЛАБОРАТОРНОЇ РОБОТИ ТА ОФОРМЛЕННЯ ЗВІТУ

Лабораторна робота включає самостійне опрацювання теоретичного матеріалу, вивчення документації з апаратних засобів, методик проведення розрахунків, програмування і моделювання, обробку й інтерпретацію результатів. При проведенні лабораторного практикуму необхідно:

- підготуватися до експрес-опитування з теоретичного матеріалу, необхідного для виконання роботи (щоб отримати допуск до лабораторної роботи);

- підготувати і оформити план виконання лабораторної роботи;

- після виконання лабораторної роботи провести необхідні розрахунки і оформити звіт, який повинен містити:

- 1) титульний аркуш (див. Додаток А);
- 2) мету роботи;
- 3) опис словами;
- 4) алгоритм роботи у вигляді блок схеми;
- 5) необхідні математичні розрахунки роботи пристрою;
- 6) текст розробленої програми;
- 7) основні результати моделювання пристрою;
- 8) стислі висновки по роботі, в яких:

- вказати, що досліджувалось;
- вказати методику (чи метод) дослідження;
- вказати особливості алгоритму роботи пристрою;
- визначити характер роботи пристрою при моделюванні;
- відзначити подібність і відмінність результатів моделювання від теоретичних даних;

- здати звіт з виконаної лабораторної роботи та відповісти на запитання викладача (отримати залік за виконану роботу).

1. ЛАБОРАТОРНА РОБОТА №1 “МОЖЛИВОСТІ ПРИБОРІВ НА БАЗІ МІКРОКОНТРОЛЕРА ESP8266 (ПЛАТА NODEMCU) ДЛЯ ЗАСТОСУВАННЯ У КОМП'ЮТЕРНИХ СИСТЕМАХ ВИМІРЮВАННЯ”

Мета роботи – вивчити базові можливості платформи NodeMCU на базі мікроконтролера ESP8266.

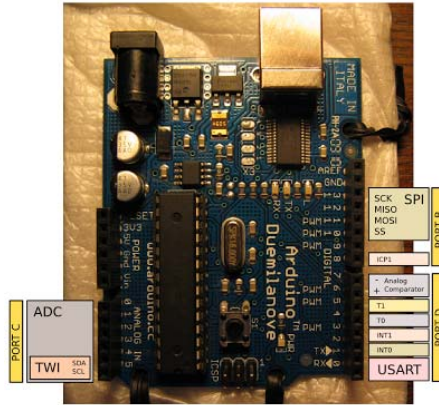
1.1 Короткі теоретичні відомості

1.1.1 Загальні відомості про NodeMCU (ESP8266)

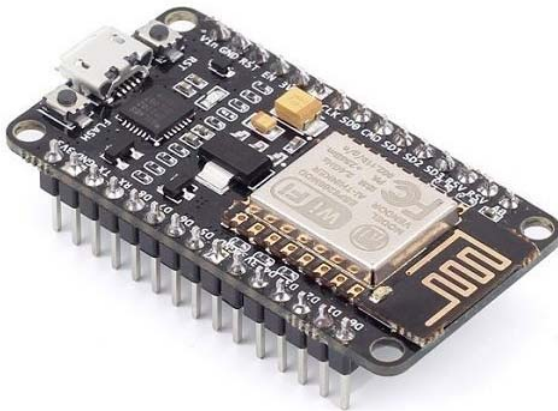
Різні пристрої та прилади підключаються до мікроконтролеру через порти введення-виведення. Мікроконтролер, який є ядром Arduino, це 8-бітний мікроконтролер, і тому його порти введення-виведення також є 8-бітними. Це загальний принцип, і хоча теоретично порт введення-виведення 8-бітного мікроконтролера може бути і 16-ти, і 32-х, і 64-бітовим, але практичного сенсу оснащувати 8-бітний мікроконтролер не 8-бітовим портом вводу-виводу немає, оскільки для читання або запису даних в такий порт довелося б використовувати кілька команд асемблера, тобто операція обміну даними з таким портом відбувалася б тільки за кілька тактів. Зовнішній вигляд контролера Arduino UNO наведено на рис. 1.1а.

Порт введення-виведення звичайного призначення (GPIO – General Purpose Input / Output) являє собою кілька виводів (пинов) мікроконтролера по числу біт, тобто в даному випадку це 8 виводів. Якщо подивитися на плату Arduino, то з одного боку знаходяться дві групи по 8 контактів, позначені як цифрові (Digital), а з іншого боку дві групи по 6 контактів, одна з яких - це контакти напруг живлення і пін скидання (Reset), а друга - це 6 контактів, позначених як входи для аналогових сигналів (Analog In).

З точки зору програми для мікроконтролера кожен порт – це кілька спеціальних регістрів (змінних), виробляючи читання або запис даних в які можна змінювати стан або режим роботи виводів мікроконтролера. Оскільки кожному виводу порту звичайного призначення відповідає 1 біт, то напруга на відповідному виводі може змінюватися, приймаючи логічне значення: або високий рівень (HIGH), або низький рівень (LOW).



а)



б)



в)

Рисунок 1.1 – Зовнішній вигляд контролерів Arduino UNO (а) і NodeMCU (б), блоку живлення NodeMCU (в)

Як правило високому рівню напруги відповідають напруга, рівне або трохи менше, ніж напруга живлення мікроконтролера (для Arduino це +5 вольт), а низькому рівню відповідає або рівень землі (0 В), або невелика напруга (наприклад, +0,2 або +0,6 В). Половина напруги високого рівня називається високим нулем (Hi-Z).

Для логічних мікросхем існує ряд електричних стандартів напруг, заснованих на технології, по якій дана мікросхема виконана. Наприклад, TTL, CMOS і т.п. Якщо дві мікросхеми з однаковою

напругою живлення виконані за різними технологіями (наприклад, одна з їх CMOS, а інша TTL), то вони як правило логічно сумісні між собою, якщо мова йде тільки про сигнали 1 або 0. Але рівень сигналу Hi-Z у них може відрізнятися, що може потребувати їх електричного узгодження.

Наприклад, рівень Hi-Z мікросхеми, яка отримує сигнал дорівнює +2В, а рівень Hi-Z мікросхеми, яка сигнал то приймає, то передає +2,4В). Таким чином друга мікросхема передає сигнал першій, і потім переводить порт в стан введення даних, тобто у неї на виведенні рівень Hi-Z +2.4В. А перша читає порт, а там логічна одиниця.

Але на практиці з подібними проблемами можна зіткнутися, хіба що намагаючись передати сигнал з виходу мікросхеми з напругою живлення +3,3 на вхід мікросхеми з напругою живлення + 5В. Якщо у другий рівень Hi-Z становить 2В, то електричного узгодження не буде потрібно, а якщо 2,4 В, то потрібно поставити між цими выводами схему узгодження рівнів на польовому транзисторі.

Проте, при підключенні мікросхем один до одного це слід враховувати, оскільки мені доводилося стикатися наприклад з тим, що мікросхема живиться від напруги +5В, а логічна одиниця у неї +3,2В, і хоча вона при цьому працює, але нагрівається під час роботи градусів до 60 за Цельсієм. Тобто напруга живлення +5 В, а порти введення-виведення, хоч і сумісні з 5-вольтової логікою, однак 3.3-вольтові, тобто для роботи з такою мікросхемою краще використовувати мікроконтролер з напругою живлення +3,3 (прим. Це так званий електричний стандарт інтерфейсу PCI).

Мікроконтролер Arduino, до речі, цілком може працювати на частоті 8МГц і з напругою живлення +3,3, але в конструктиві Arduino така можливість не передбачена, тому до Arduino краще підключати мікросхеми 5-вольтової логіки.

Всього мікроконтролер ATmega328p (як і ATmega168) в корпусі PDIP має три порти введення-виведення. Вони позначені, як порти В, С і D:

- Порт D - це група виводів Digital з номерами від 0 до 7;
- Порт В - це 5 виводів Digital з номерами від 8 до 13. Їм відповідають 5 молодших біт порту В (старші 2 виводи використовуються для підключення кварцового резонатора 16 МГц);

- Порт C - це група виводів Analog In. Їм відповідають 6 молодших біт порту C (7-й біт використовується як сигнал Reset, 8-й біт не використовується).

Таким чином, всього Arduino має 20 контактів введення-виведення звичайного призначення (GPIO).

Крім портів GPIO до мікроконтролеру можна підключати пристрої через спеціальні інтерфейси. Наприклад, мікроконтролер Arduino оснащений наступними інтерфейсами:

- USART - універсальний синхронний і асинхронний послідовний приймач і передавач (Universal Synchronous and Asynchronous serial Receiver and Transmitter). Виводи Digital з номерами 0 (приймач) і 1 (передавач). Він використовується, наприклад, для підключення Arduino до комп'ютера через USB, але не безпосередньо, а за допомогою мікрохеми перехідника USB-to-serial;

- Master / Slave SPI - послідовний периферійний інтерфейс (Serial Peripheral Interface). Виводи Digital з номерами від 10 до 13. У режимі slave (підлеглий) він дозволяє програмувати Arduino через ICSP за допомогою програматора. У режимі master (майстер) він дозволяє підключати до Arduino різні пристрої з інтерфейсом SPI – наприклад використовувати Arduino як програматор або підключити до Arduino флешку формату SD Card або MMC;

- TWI – двопровідний інтерфейс (Two-Wire Interface), сумісний з інтерфейсом I2C. Виводи Analog In з номерами 4 і 5. Він дозволяє підключати до Arduino, наприклад LCD-дисплеї або мікрохеми пам'яті EEPROM;

- ADC - аналого-цифровий перетворювач або АЦП (Analog-to-Digital Converter). Він дозволяє підключати до Arduino аналогові прилади. Наприклад, потенціометри, датчики (сенсори), мікрофон тощо. Всього у мікроконтролера Arduino шість 10-бітних аналогових входів - група контактів Analog In;

- Аналоговий компаратор (AIN). Виводи Digital з номерами 6 (позитивний вхід) і 7 (негативний вхід). Якщо напруга на позитивному вході компаратора перевищує напругу на негативному вході, то мікроконтролер може генерувати спеціальне переривання, або використовувати цей сигнал для детектора фронтів на 16-бітному таймері. Як негативний входу замість виведення Digital 7 може бути використаний будь-який з виводів групи Analog In. Дозволяє Arduino вимірювати частоту і рівень аналогового сигналу;

- PWM - широтно-імпульсний модулятор або ШІМ (Pulse Width Modulation). Виводи Digital с номерами 3, 5, 6, 9, 10, 11. Дозволяє мікроконтролеру Arduino генерувати аналогові сигнали з розрядністю від 2 біт з частотою 250 кГц до 8 біт з частотою 15625 кГц. Всього мікроконтролер Arduino має 6 ШІМ-модуляторів, по 2 на кожен з 3 таймерів;

- ICP - модуль захоплення введення (Input Capture Unit). Вивід Digital з номером 8. Дозволяє Arduino вимірювати частоту і цикл заповнення цифрового сигналу;

- T0 і T1 - Джерела зовнішнього тактового сигналу для таймерів-лічильників 0 і 1. Виводи Digital з номерами 4 і 5. Дозволяють підключати до Arduino, наприклад, мікросхеми годинника реального часу або синхронізувати Arduino з пристроями, що працюють на частотах, які не кратні системній частоті Arduino 16 МГц (наприклад, 5 МГц або 1.8432 МГц).

Таким чином, кожен пін Arduino може використовуватися або як вивід звичайного призначення (GPIO), або як один з виводів спеціального призначення. Після натискання на кнопку скидання або відразу після включення, всі спеціальні функції виводів відключені, а самі виводи знаходяться в стані Hi-Z, тобто при скиданні вони стають електрично і логічно нейтральними.

Тобто для того, щоб задіяти для того чи іншого вивода мікроконтролера його спеціальну функцію, його слід відповідним чином програмно ініціювати. Це робиться записом потрібних значень у відповідні регістри.

NodeMCU - це платформа на основі модуля ESP8266 (рис. 1.16). Плата призначена для зручного управління різними схемами на відстані за допомогою передачі сигналу в локальну мережу або інтернет через Wi-Fi. Можливості застосування цієї плати дуже обшірніе. Наприклад, на базі Node MCU можна створити «розумний будинок», налаштувавши управління світлом або вентиляцією через телефон, реєстрацію показань датчиків і багато іншого.

Розмір плати NodeMCU - 6x3 см. Плата досить компактна, це дозволяє використовувати її в більшій кількості проектів. «Ноги» NodeMCU розташовані так, що її без проблем можна встановити в макетну плату (breadboard). Існує кілька поколінь плат NodeMcu - V1 (версія 0.9), V2 (версія 1.0) і V3 (версія 1.0). Позначення V1, V2, V3

використовуються при продажу в інтернет-магазинах. Нерідко відбувається плутанина в платах - наприклад, V3 зовні ідентична V2. Також всі плати працюють за принципом open-source, тому їх можуть виробляти будь-які фірми. Але в даний час виробництвом плат NodeMcu займаються Amica, DOIT і LoLin/Wemos.

Відмінності від інших модифікацій

Плати покоління V1 і V2 легко відрізнити за різним розміром. Також в другому поколінні оснащено поліпшеною модифікацією чіпа ESP-12 і 4 Мб флеш-пам'яті. Перша версія, застаріла, виконана у вигляді яскравої жовтої платформи. Використовувати її незручно, так як вона покриває собою 10 контактів макетної плати. Плата другого покоління зроблена з виправленням цього недоліку - вона стала більш вузькою, виходи добре підходять до контактів плати. Плати V3 зовні нічим не відрізняються від V2, вони мають більш надійний USB-контакт. Випускає плату V3 фірма LoLin, з відмінностей від попередньої плати можна відзначити те, що один з двох зарезервованих контактів використовується для додаткової землі, а другий - для подачі USB живлення. Також плата відрізняється великим розміром, ніж попередні види.

Живлення модуля NodeMcu

Живлення модуля NodeMcu здійснюється через вбудовані мікро USB-роз'єм або від зовнішнього джерела напруги на 5 В підключеним до VIN і GND. Так як, робочі напруга ESP8266 становить від 3 В до 3.6 В., на платі встановлений стабілізатор напруги AMS1117 з вихідним напругою 3.3 В і струмом до 600 мА. Для зручності і можливості підключення зовнішніх пристроїв (датчиків), додатково виведено 3.3 В на бічні контакти плати NodeMcu з позначенням 3V3.

Переваги NodeMcu v3

- Наявність інтерфейсу UART-USB з роз'ємом мікро USB дозволяє легко підключити плату до комп'ютера.
- Наявність флеш-пам'яті на 4 Мбайт.
- Можливість оновлювати прошивку через USB.
- Можливість створювати скрипти на LUA і зберігати їх в файлової системі.

Технічні характеристики модуля NodeMCU

1. Підтримує Wi-Fi протокол 802.11 b/g/n;
2. Підтримувані режими Wi-Fi - точка доступу, клієнт;

3. Вхідна напруга 3,7 В - 20 В;
4. Робоча напруга 3В-3,6;
5. Максимальний струм 220мА;
6. Вбудований стек TCP / IP;
7. Діапазон робочих температур від -40С до 125С;
8. 80 МГц, 32-бітний процесор;
9. Час пробудження і відправки пакетів 22МС;
10. Вбудовані TR перемикач і PLL;
11. Наявність підсилювачів потужності, регуляторів, систем управління живленням.

На лицьовій частині плати роз'єм Micro USB, за допомогою якого в контролер заливають скетчі або подають живлення від powerbank-а або комп'ютера.

Поруч з роз'ємом розташовуються дві кнопки: «Flash» і «Reset». Кнопка «Flash» використовується для налагодження, а кнопка «Reset» для перезавантаження плати.

Найбільше місця на платі займає чип ESP8266, на якому заставлений мікропроцесор з тактовою частотою 80 МГц (можна розігнати до 160 МГц). Плата має 4 мегабайта Flash-пам'яті.

Для живлення на плату можна подавати напругу від 5 до 12 В, але рекомендується від 10 В. Можна жити як від Micro USB, так і від контакту Vin (від 5В.). Також існують додаткові плати розширення для зручного живлення модулів. Плата для живлення NodeMCU (рис. 1.1в) споживає невелику кількість енергії. Це дозволяє використовувати її з автономним живленням.

NodeMCU має 11 портів введення-виведення загального призначення. Деякі з портів мають додаткові функції:

- D9, D10 - UART
- D1, D2 - I²C / TWI
- D5-D8 - SPI
- D1-D10 - виходи з ШІМ (PWM)
- A0 - аналоговий вхід з АЦП.

Існує кілька поколінь плат NodeMcu - v.1, v.2 і v.3. Переваги NodeMcu v.3:

1. Наявність інтерфейсу UART-USB з роз'ємом мікро USB дозволяє легко підключити плату до комп'ютера.
2. Наявність флеш-пам'яті на 4 Мбайт.
3. Можливість оновлювати прошивку через USB.

4. Можливість створювати скрипти на LUA і зберігати їх в файлової системі.

Модуль NodeMcu(v.3) має 11 контактів введення-виведення загального призначення (рис. 1.2). Крім цього деякі з контактів володіють додатковими функціями:

- D1-D10 - висновки з широтно-імпульсною модуляцією;
- D1, D2- висновки для інтерфейсу I²C / TWI;
- D5-D8 - висновки для інтерфейсу SPI;
- D9, D10 - UART;
- A0 - вхід з АЦП.

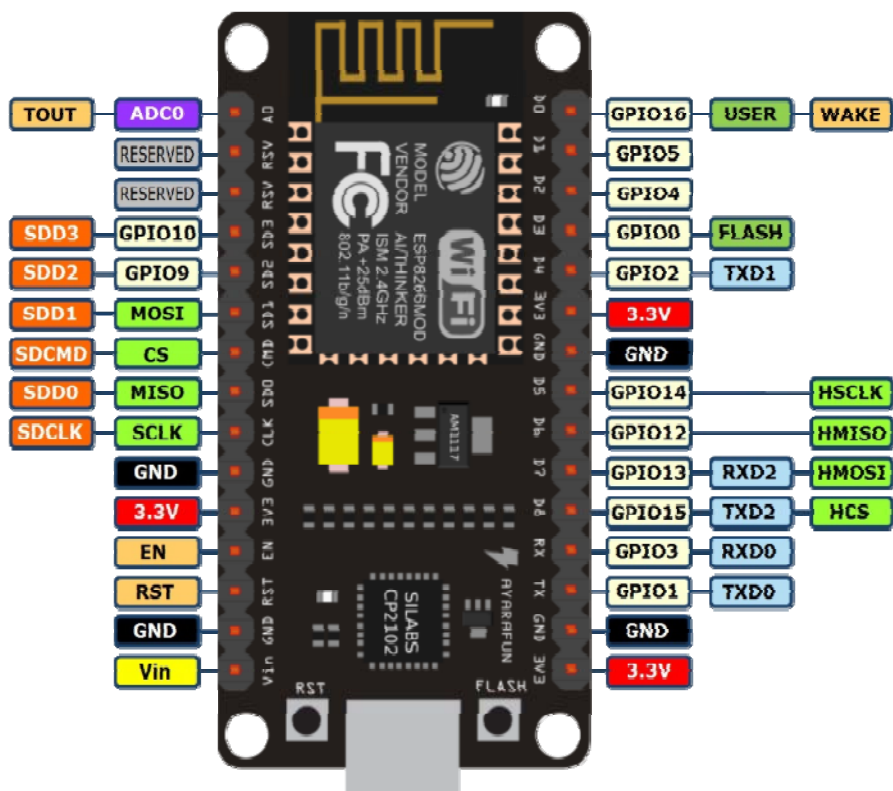


Рисунок 1.2 – Схема контактів платформи NodeMCU (v.3) с WiFi

1.1.2 Середовище Arduino IDE

Для програмування плат Arduino існує спеціальне інтегроване середовище розробки IDE Arduino. Розглянемо зовнішній вигляд програми (рис.1.3).

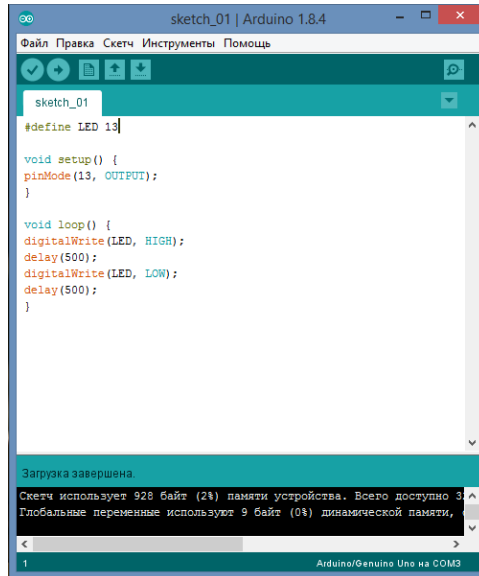


Рисунок 1.3 – Вікно середовища Arduino IDE

Вгорі, під рядком заголовка знаходиться рядок Меню, з наступними пунктами: Файл, Правка, Скетч, Налаштування, Сервіс.

Під рядком меню розташовується рядок інструментів, на яку винесені часто використовувані команди:



– перевірити,



– загрузити,



– створити,



– зберегти,



– відкрити,



– монітор порту.

Нижче розташовується рядок з назвами відкритих вкладок. В кінці рядка знаходиться піктограма, при натисканні на яку відкривається меню, що відноситься до вкладок.

У центрі вікна знаходиться робоча область, в якій пишеться код програми. Під робочою областю знаходиться вікно повідомлень, куди виводяться повідомлення про результат компіляції, завантаження і т.д.

1.1.3 Програмування на мові Arduino IDE

Програма, що написана в IDE Arduino, називається скетчем. Кожен скетч повинен складатися як мінімум з двох обов'язкових функцій:

Лістинг 1.1 – Необхідні блоки скетча

```

1 void setup() { /* відкриваюча дужка на початку
  програми ініціалізації */
  /* Ця програма виконується 1 раз і призначена
  для приведення
2 системи у робочий стан, сюди можна записувати
  функції
  ініціалізації портів і периферійних
  пристроїв */
3 } /* закриваюча дужка в кінці програми
  ініціалізації */
4 void loop() { /* відкриваюча дужка на початку
  основного циклу */
  /* по суті нескінченний цикл, це основний
5 цикл програми, всередині
  якого виконується її основна частина (крім
  початкової ініціалізації) */
6 } /* закриваюча дужка в кінці основного циклу
  */

```

Функція – структурна одиниця програми, яка має ім'я і стримає в собі деяку послідовність дій.

Більшість записів в програмі повинні закінчуватися ";", виключення складають оператори циклів, вибору, умов, а також опис прототипів функцій. З цими та іншими конструкціями ми познайомимося в процесі вивчення курсу.

На початку програми, перед функцією `setup`, зазвичай, оголошуються змінні і визначення. Після включення живлення плати першої виконується функція `setup`. Вона виконується тільки один раз. Зазвичай в ній не започатковано режими роботи портів: порти, до яких підключені різні датчики, встановлюються як входи, а порти з виконавчими пристроями як виходи.

Код, написаний в функції `loop()` починає виконуватися після виконання функції `setup()`, і виконується в нескінченному циклі знову і знову. У цій функції виконується основна робота: різні обчислення, отримання значень датчиків, висновки значень на порти.

1.1.4 Програмування ESP8266

Чіп ESP8266 - це відмінний інструмент для проектів, де використовується управління через інтернет, тобто що дозволяють управляти електронними пристроями з будь-якої точки світу. Чіп ESP8266 поставляється з прошивкою, що складається з AT-команд. Це означає, що його можна підключити до Arduino, фактично отримавши повнофункціональний WiFi модуль. Крім того, завдяки потужних обчислювальних ресурсів і вбудованій пам'яті модуль ESP8266 можна інтегрувати в системи з датчиками і іншими електронними пристроями, що зчитують або передають дані за допомогою GPIO-контактів.

Чіп ESP8266 комунікує з платою Arduino через послідовний інтерфейс. Тобто, щоб отримувати команди і відправляти відповідні дані, він використовує контакти RX і TX на Arduino (тобто цифрові контакти 0 і 1). Модуль має повну підтримку стека TCP / UDP, завдяки чому його можна налаштувати як веб-сервер. IDE Arduino потрібно налаштувати на сумісність з модулем ESP-01, і для цього в IDE Arduino потрібно буде встановити ядро ESP8266.

1.2 Хід роботи

1.2.1 Підключення плати NodeMCU до комп'ютера

Підключаємо плату NodeMCU до комп'ютера за допомогою USB кабелю. Спочатку необхідно встановити драйвер CP2102, потім відкриваємо Arduino IDE, заходимо в розділ «Файл»/«Налаштування» і знайдемо рядок «додаткові посилання для менеджера плат» і вставляємо туди наступне посилання:

http://arduino.esp8266.com/versions/2.3.0/package_esp8266com_index.json

Заходимо в розділ «інструменти»/«плата»/«менеджер плат», де вибираємо «esp8266» і завантажуюмо останню версію. Заходимо в розділ «інструменти» -> «плата» і знаходимо Node MCU. Необхідно в тому ж розділі зайти в розділ «порт» і вибрати той, в який підключена плата. Встановити в розділі інструменти: Uploadspeed (115200 baud).

1.2.2 Установка мікроконтролера та підключення елементів на макетної платі

Найсучаснішим варіантом для створення прототипів є макетна плата (рис. 1.4), яка володіє безперечними перевагами:

- Можливість проводити налагоджувальні роботи велику кількість разів, змінюючи модифікацію схем і способи підключення пристроїв;
- Можливість з'єднання декількох плат в одну велику, що дозволяє працювати з більш складними і великими проектами;
- Простота і швидкість створення прототипів;
- Довговічність і надійність.

Звичайно, є у цього варіанту монтажу та недоліки:

- У реальних проектах з'єднання у плати не будуть настільки ж надійні, як при пайку. Будь-яка вібрація буде потихеньку послаблювати контакти і це обов'язково з часом призведе до несподіваних проблем. Тому в реальних проектах використовують інші види монтажу елементів.
- Зовнішній вигляд проектів з локшиною у вигляді проводів над безкрайними білими просторами плати не можна назвати професійним і естетичним. Хочуть такий вид завжди заворожує глядачів і формує у проекту імідж чогось "моторошно складного, раз стільки проводів".
- Плата з таким видом монтажу завжди буде займати більше місця за рахунок нависають проводів. Значить, для неї потрібен корпус великих обсягів з фіксацією і захистом від вібрації.
- Вартість макетної плати. Нехай плати і не є дорогими пристроями, але все одно вам потрібно буде їх придбати додатково до мікроконтролеру і інших елементів.

Не дивлячись на деякі недоліки, альтернативних варіантів по простоті і доступності для монтажу перших схем у початківців практично немає. Сьогодні можна зустріти величезну кількість проектів, в яких всі елементи розміщені саме на макетної платі. Майже всі приклади з підручників з основ робототехніки і Ардуіно використовують цей варіант монтажу.

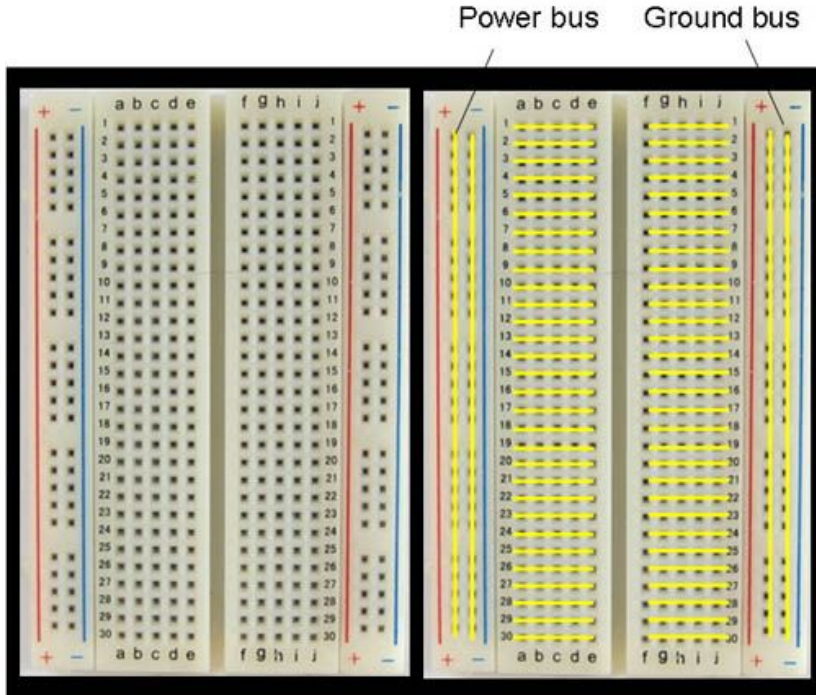


Рисунок 1.4 – Схема макетної плати

Макетна плата має пластикову основу з безліччю отворів (стандартне відстань між ними становить 2,54 мм). В середині конструкції розташовані ряди металевих пластин. На кожній пластині є кліпси, які захищені в пластиковій частині установки.

Включення проводів виконується саме в ці кліпси. При підключенні провідника до одного з окремих отворів, контакт одночасно підключається і до всіх інших контактам окремого ряду. Отже, підключаючи контакти інших пристроїв до решти кліпсам, ми пов'язуємо їх провідником - рейкою з кліпсами.

Варто звернути увагу, що одна рейка містить 5 кліпс. Це загальний стандарт для всіх макетних плат. Тобто, до кожного рельсу можна під'єднати до п'яти елементів, і вони будуть з'єднані між собою. Слід зазначити, що хоча в кожному ряду розташовані десять отворів, вони все-таки розділені на дві ізолювані частини, по п'ять у кожній. Між ними розташований рейок без контактів. Така конструкція

необхідна для ізоляції пластин один від одного, і дозволяє просто підключати мікросхеми, виконані в DIP-корпусах (рис. 1.5).

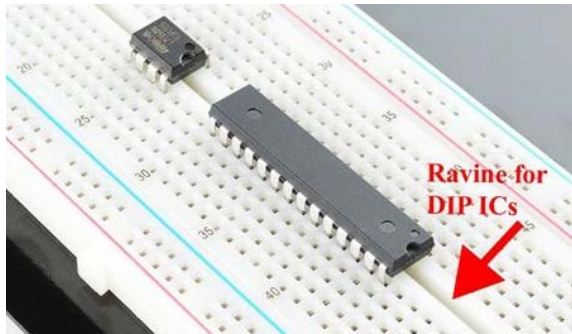


Рисунок 1.5 – Підключення мікросхеми до макетної плати

Для спрощення орієнтації на макетну плату також нанесені цифрові і буквені позначення, якими можна керуватися, створюючи, наприклад, інструкцію для підключення.

Деякі макетні плати включають також по дві лінії живлення з кожної зі сторін. Зазвичай «червона лінія» використовується для подачі «+» напруги, «синя» - для «-». За рахунок наявності двох шин живлення на плату можуть подаватися два різних рівня напруги.

Увага! Макетні плати абсолютно неприпустимо використовувати з напругою 220В. Якщо плата більша, то лінії живлення "розриваються" посередині. Це дозволяє використовувати більшу кількість варіантів підключення. Наприклад, ви зможете зібрати на одній платі пристрою з живленням 3 і 5 В.

При створенні схеми в отвори на пластиковому корпусі вставляються необхідні елементи - конденсатори, резистори, різні індикатори, світлодіоди і т.д. Ширина роз'ємів дозволяє підключати до контактів провідники з перетином від 0,4 до 0,7 мм.

Потрібно з'єднати між собою два елементи - світлодіод і резистор (рис. 1.6). Для цього ніжку першого елемента (світлодіода) вставляємо в ряд номер 2. Другу ніжку - в інший ряд, наприклад, 3. Якщо вставите ніжку в той же ряд, схема працювати не буде, тому що обидві ніжки через загальну рельсу будуть з'єднані залізним провідником. Буде коротке замикання.

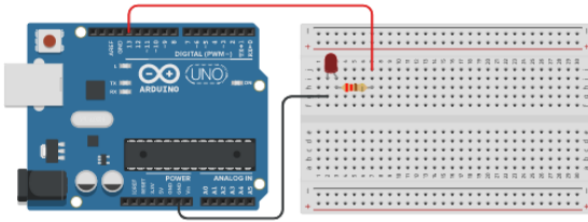


Рисунок 1.6 – Схема підключення світлодіода до монтажній платі

У Лістингу 1.2 до 13 контакту підключений світлодіод "L":

Лістинг 1.2 – Програма для миготіння діодом

```
void setup()
{
    pinMode(13, OUTPUT);    // переключаємо
    цифровий вивід в режим виходу
}
void loop()
{
    digitalWrite(13, HIGH); // вмикаємо світлодіод
    delay(1000);            // чекаємо 1 секунду
    digitalWrite(13, LOW);  // вимикаємо
    світлодіод
    delay(1000);            // чекаємо 1 секунду
}
```

1.2.3 Підключення миготливого світлодіода до платформи NodeMCU

Принципова схема підключення представлена на рис. 1.7.

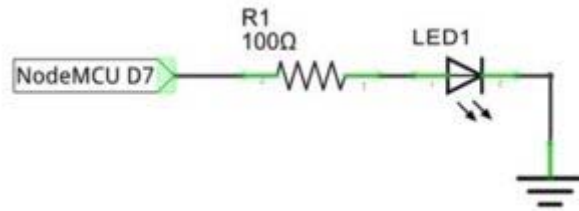


Рисунок 1.7 – Вибір порту у середовищі Arduino IDE

Підсумковий макет повинен виглядати наступним чином (рис. 1.8):

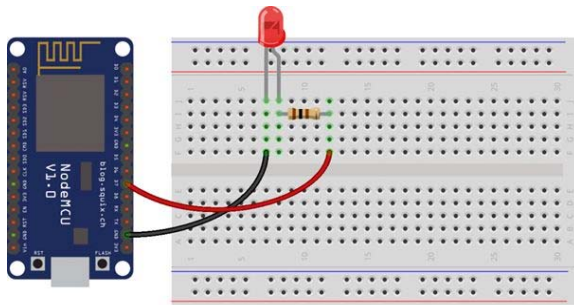


Рисунок 1.8 – Схема підключення світлодіода

Сама плата працює від напруги 3.3 В, тому для підключення світлодіода потрібно використовувати резистор. В даному прикладі для червоного світлодіода береться резистор номіналом 65 Ом. Схожим способом до плати підключається і фотодіод (рис. 1.9).

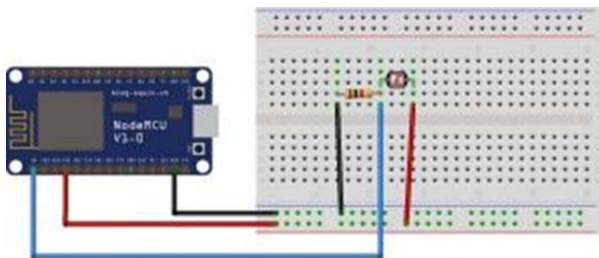


Рисунок 1.9 – Схема підключення фотодіод

Плату NodeMCU можна використовувати і для управління по ІЧ каналу. Для управління потрібен пульт дистанційного керування з ІЧ приймачем і сама платформа. Інфрачервоний приймач підключається за схемою (рис. 1.10), представленої нижче:

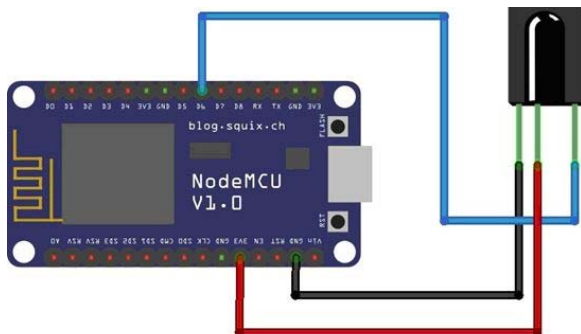


Рисунок 1.10 – Схема підключення інфрачервоного приймачу

1.3 Оцінювання і контроль

Перед включенням живлення пред'явите зібрані схеми викладачу. Переконайтеся, що всі контакти надійно зафіксовані. Проконтролюйте правильність роботи алгоритму функціонуючого приладу.

1.4 Контрольні запитання

1. Яка структура модуля (платформи) NodeMcu?
2. Який контролер використовується на платі NodeMcu?
3. Яка мова програмування використовується для написання програм у NodeMcu?
4. Якими пристроями оснащена плата NodeMcu?
5. Перрахуйте можливі канали зв'язку NodeMcu з комп'ютером.
6. Що таке Arduino UNO?
7. Що таке Arduino IDE?;
8. Який контролер використовується на платі Arduino UNO?
9. Яка мова програмування використовується для написання програм у Arduino IDE?
10. Якими пристроями оснащена плата Arduino UNO?
11. Перрахуйте можливі канали зв'язку Arduino UNO з іншими пристроями в т.ч. з комп'ютером.

2. ЛАБОРАТОРНА РОБОТА №2 “ПІДКЛЮЧЕННЯ ПЛАТ ARDUINO І NODEMCU ДО КОМП'ЮТЕРА ТА ВИМІРЮВАННЯ НАПРУГИ”

Мета роботи – ознайомитися з принципом взаємодії із пристроями за допомогою послідовного COM-порту USB (Universal Serial Bus, універсальна послідовна шина).

2.1 Короткі теоретичні відомості

2.1.1 Комунікаційні порти персонального комп'ютера

Більшість мікроконтролерів мають безліч портів введення/виведення (рис. 2.1). Для зв'язку з ПК найбільш придатний з них протокол UART. Це протокол послідовної асинхронної передачі даних. Для його перетворення в інтерфейс USB на платі є конвертор USB-RS232 - FT232RL.

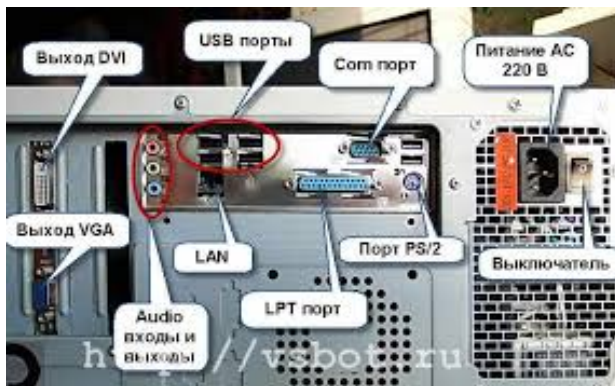


Рисунок 2.1 – Комунікаційні порти персонального комп'ютера

Для перевірки роботи програми необхідно, щоб на комп'ютері була термінальна програма, яка приймає дані з COM-порту. В Arduino IDE вже вбудована така. Для її виклику виберіть в меню Сервіс> Монітор порту. Вікно цієї утиліти наведено на рис. 2.2.

Для запуску робота порту UART служить функція Serial.begin(). Єдиний її параметр – це швидкість. Про швидкості необхідно домовлятися на передавальній і приймальній стороні заздалегідь, так як

протокол передачі асинхронний. У розглянутому прикладі швидкість 9600 біт/с.

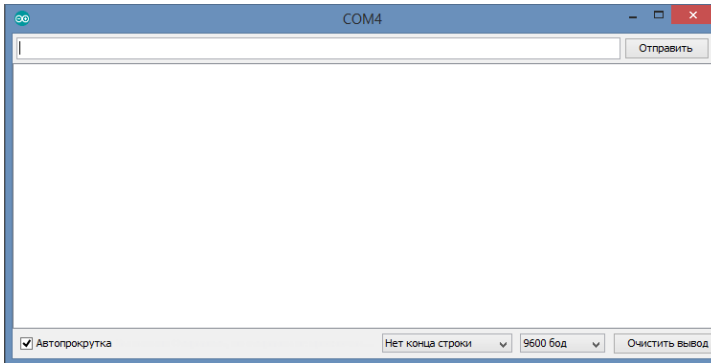


Рисунок 2.2 – Вікно монітора послідовного порту

Для запису значення в порт використовуються три функції:

- `Serial.write()` – записує в порт дані в двійковому вигляді.
- `Serial.print()` може мати багато значень, але всі вони служать для виведення інформації в зручній для людини формі. Наприклад, якщо інформація, зазначена як параметр для передачі, виділена лапками - термінальна програма виведе її без зміни. Якщо ви хочете вивести якесь значення в певній системі числення, то необхідно додати службове слово: BIN-двоичная, OCT - восьмерична, DEC - десяткова, HEX - шістнадцяткова. Наприклад, `Serial.print (25, HEX)`.
- `Serial.println()` робить те ж, що і `Serial.print()`, але ще переводить рядок після виведення інформації.

USB (Universal Serial Bus: універсальна послідовна шина, призначена для з'єднання комп'ютерів і периферійних пристроїв.

Метою створення USB стандарту є:

- уніфікація роз'ємів і кабелів;
- нормування використання енергії;
- створення протоколів обміну даними;
- уніфікація функціональності і драйверів приладів;
- можливість "гарячого" приєднання пристроїв.

USB-кабель, по суті, дві звиті пари: однією з них передаються дані в кожному напрямку (диференціальне включення), а інша використовується для живлення периферійного пристрою (+5 В, 500 мА). Вбудовані лінії живлення дозволяють використовувати USB-пристрої, що не мають власного блоку живлення, чи заряджати акумулятори переносних пристроїв (фотоокамер, плеєрів тощо), якщо ці пристрої споживають струм силою до 500 мА. Стандарт USB 3.0 допускає навантаження лінії живлення струмом до 900 мА.

2.1.2 Принцип роботи СОМ-порту. Відправка команд з ПК

Перш ніж цим займатися, необхідно отримати уявлення щодо того, як працює СОМ-порт. В першу чергу весь обмін відбувається через буфер пам'яті. Тобто коли ви відправляєте щось з ПК пристрою, дані поміщаються в певний спеціальний розділ пам'яті. Як тільки пристрій готовий - воно вичитує дані з буфера. Перевірити стан буфера дозволяє функція `Serial.available()`. Ця функція повертає кількість байт в буфері. Щоб прочитати ці байти необхідно скористатися функцією `Serial.read()`. Розглянемо роботу цих функцій на прикладі:

Лістинг 1.1 – Пиклад роботи з послідоаним портом

```
int val = 0;

void setup() {
    Serial.begin(9600);
}
void loop() {
    if (Serial.available() > 0) {
        val = Serial.read();
        Serial.print("I received: ");
        Serial.write(val);
        Serial.println();
    }
}
```

Після того, як код буде завантажений в пам'ять мікроконтролера, відкрийте монітор СОМ-порту. Введіть один символ і натисніть Enter.

В поле отриманих даних ви побачите: "I received: X", де замість X буде введений вами символ.

Програма нескінченно крутиться в основному циклі. У той момент, коли в порт записується байт функція `Serial.available()` приймає значення 1, тобто виконується умова `Serial.available() > 0`. Далі функція `Serial.read()` вчитує цей байт, тим самим очищаючи буфер. Після чого за допомогою вже відомих вам функцій відбувається висновок.

Використання вбудованого в Arduino IDE монітора COM-порту має деякі обмеження. При відправці даних з плати в COM-порт висновок можна організувати в довільному форматі. А при відправці з ПК до плати передача символів відбувається відповідно до таблиці ASCII. Це означає, що коли ви вводите, наприклад символ "1", через COM-порт відправляється в двійковому вигляді "00110001" (тобто "49" в десятковому вигляді). Змінимо код і перевіримо це твердження:

Лістинг 1.2 – Пиклад роботи з послідоаним портом

```
int val = 0;
void setup() {
    Serial.begin(9600);
}
void loop() {
    if (Serial.available() > 0) {
        val = Serial.read();
        Serial.print("I received: ");
        Serial.println(val, BIN);
    }
}
```

Після завантаження, в моніторі порту при відправці "1" ви побачите в відповідь: "I received: 110001". Змінити формат виведення і дізнайтеся, що приймає плата при інших символах.

2.2 Хід роботи

2.2.1 Підключення потенціометра через COM порт до ПК із залученням Arduino IDE

Потенціометр це змінний резистор, тобто опір якого можна регулювати вручну. У потенціометра три контакта: перший живлення,

другий вихід (визначає положення), третій земля. На рис. 2.3 показаний зовнішній вигляд потенціометра. Зберіть схему, яку наведено на рис. 2.4:



Рисунок 2.3 – Потенціометр

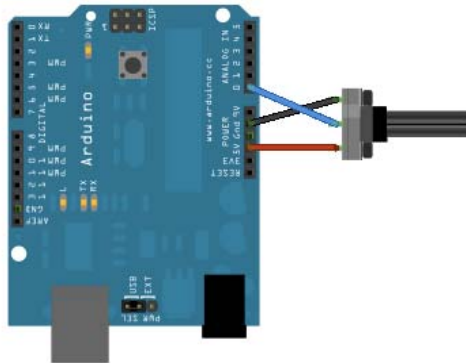


Рисунок 2.4 –Схема підключення потенціометра

Напишіть скетч (лістинг 1.3) и загрузити його в плату.

Лістинг 1.3 – Отримання даних з Arduino UNO через послідовний порт

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

void loop() {
    int sensorValue = analogRead(A0);
    Serial.println(sensorValue);
    delay(1);
}
```

Здійснити передачу значення напруги через COM порт за допомогою програми (лістинг 1.4). Написати скетч и загрузити його в плату:

Лістинг 1.4 – Отримання даних з Arduino UNO через послідовний порт

```
void setup() {
  Serial.begin(9600);
}
void loop() {
  int sensorValue = analogRead(A0);
  float voltage = sensorValue * (5.0 / 1023.0);
  Serial.println(voltage);
}
```

2.2.2 Підключення Arduino UNO до ПК за допомогою USB порта

Для Arduino Nano потрібен кабель з роз'ємами типу A і mini-B.

При підключенні пристрою до комп'ютера слід мати на увазі наступне. Arduino Uno, Mega, Duemilanove і Arduino Nano автоматично розпізнають джерело живлення - будь це USB або зовнішнє джерело живлення. Якщо ж ви використовуєте Arduino Diecimila, то перед підключенням до комп'ютера необхідно переконатися, що пристрій налаштовано на живлення від USB. Джерело живлення на цій платі вибирається джампером (невелика пластмасова деталь, що дозволяє замкнути між собою тільки два висновки з трьох), що підключає ланцюг живлення плати або до USB, або до роз'єму живлення. Перед підключенням Ардуіно до комп'ютера переконайтеся, що джампер знаходиться на двох контактах, які розташовані ближче до USB-роз'єму.

За допомогою USB-кабелю підключіть Ардуіно до комп'ютера. При цьому повинен загорітися зелений світлодіод живлення (зазначений на платі як PWR).

Підключіть Ардуіно до комп'ютера і дочекайтеся, поки Windows не почне процес установки драйверів. Незважаючи на всі зусилля системи, через кілька миттєвостей процес завершиться невдачею. Зайдіть в Пуск, відкрийте Панель управління. В Панелі керування перейдіть на сторінку Система і безпека. Далі клацніть по пункту

Система і відкрийте Диспетчер пристроїв. Знайдіть розділ Порти (COM & LPT). У ньому ви побачите відкритий порт під ім'ям "Arduino UNO (COMxx)". Клацніть правою кнопкою по пункту "Arduino UNO (COMxx)" і виберіть "Оновити драйвер". Далі, у вікні, виберіть пункт "Виконати пошук драйверів на цьому комп'ютері". На завершення, виберіть файл драйвера під ім'ям "arduino.inf", розташований в папці "Drivers" в директорії завантаженого ПО Ардуіно (а не в піддиректорії "FTDI USB Drivers"). Якщо у вас стара версія IDE (1.0.3 або старіше) - вибирайте файл під ім'ям "Arduino UNO.inf". Windows завершить установку драйвера.

2.2.3 Підключення NodeMCU до комп'ютера за допомогою USB порта

Для початку роботи з NodeMCU потрібно підключити плату до комп'ютера. Першим кроком буде установка драйвера CP2102 і відкриття Arduino IDE. Потім потрібно знайти в «Файл» - «Налаштування» і в вікно «додаткові посилання для менеджера плат» вставити посилання:

http://arduino.esp8266.com/versions/2.3.0/package_esp8266com_index.json.

Після цього в меню «документи» - «плата» «менеджер плат» вибрати «esp8266» і встановити останню версію. Після виконаних дій в меню «інструменти» - «плата» потрібно знайти NodeMCU.

2.3 Контрольні запитання

- 1 Принцип передачі даних по послідовному порту.
- 2 Як здійснити конфігурацію COM-порту на платі Arduino UNO?
- 3 Функції роботи зі строками.
- 4 Алгоритм передачі через послідовний порт одного символу.
- 5 Алгоритм передачі через послідовний порт строки.
- 6 Керування Arduino UNO через послідовний порт.
- 7 Керування NodeMCU через USB порт.

3. ЛАБОРАТОРНА РОБОТА №3 „ВИКОРИСТАННЯ КОМП'ЮТЕРА ДЛЯ ВИМІРЮВАННЯ АТМОСФЕРНОГО ТИСКУ, ТЕМПЕРАТУРИ І ВИСОТИ НАД РІВНЕМ МОРЯ. СТВОРЕННЯ HTML-СТОРІНКИ ДЛЯ ВІДОБРАЖЕННЯ ВИМІРЮВАЕМОЇ ІНФОРМАЦІЇ”

Мета роботи – зробити цифровий прилад до комп'ютера для вимірювання тиску, температури і висоти над рівнем моря. Створення HTML-сторінки.

3.1 Короткі теоретичні відомості

Сьогодні персональний комп'ютер усе частіше залучається до вимірювання електричних величин. Унаслідок науково-технічного прогресу в техніці все ширше використовуються вбудовані комп'ютерні системи, засновані на мікроконтролерах.

Постає питання про розробку та конструювання вимірювальних приладів, які здатні оцифровувати та передавати дані аналогових сигналів на ПК із подальшою їх обробкою та відображенням у потрібній формі. Це вимірювання ємності, електричного опору, малих зміщень, вологості повітря, температури та інтенсивності світла. Програмні засоби, що дозволяють експлуатацію ЕОМ у ролі осцилографа для демонстрації та дослідження зміни фізичних величин, розроблено й з успіхом використовуються у наукових дослідженнях. Таким чином, ПК може бути перетворений в осцилограф, спектроаналізатор, функціональний генератор, вольтметр, систему керування різними установками чи іншими приладами, необхідними для експерименту. Причому всі ці прилади можуть бути активізовані на одному ПК одночасно. Завдяки підвищенню продуктивності сучасних мікроелектронних пристроїв при одночасному зниженні їх вартості інженери та вчені знайшли ефективні розв'язки складних завдань шляхом додавання процесорів та “інтелектуальних” мікропроцесорних компонентів у свої системи. І, як наслідок, сучасні контрольно-вимірні пристрої та системи все більш стають розподіленими.

На сьогодні персональний комп'ютер усе частіше залучається до вимірювання електричних величин. Унаслідок науково-технічного прогресу в техніці все ширше використовуються вбудовані

комп'ютери, засновані на мікроконтролерах. Постає питання про розробку та конструювання вимірювальних приладів, які здатні оцифровувати та передавати дані аналогових сигналів на ПК із подальшою їх обробкою та відображенням у потрібній формі.

3.2 Хід роботи

3.2.1 Підключення датчика BMP280 для вимірювання атмосферного тиску, температури і висоти над рівнем моря

Датчик являє собою високоточний цифровий вимірювач атмосферного тиску на базі мікрочіпа BMP280 від фірми BOSH. Датчик BMP280 був розроблений фірмою як більш технологічна модель свого попередника BMP180. Дана модифікація, на відміну від свогопредшественника, надає користувачеві 2 послідовних інтерфейсу обміну даними (SPI і I2C), а також 3 режими роботи:

- **NORMAL** – в даному режимі модуль «прокидається» з певною періодичністю, виконує необхідні вимірювання і знову засинає. Частота вимірів задається програмним шляхом, а результат зчитується при необхідності.
- **SLEEP** – режим максимально низького споживання електроенергії.
- **FORCED** – цей режим дозволяє «будити» модуль подачею зовнішнього керуючого сигналу. Після виконання вимірювань, модуль автоматично переходить в режим зниженого енергоспоживання.

Крім здатності вимірювати показання атмосферного тиску, розробник наділив BMP280 можливістю визначати температуру навколишнього середовища. Всі вироблені обчислення можуть бути відфільтровані налаштованим програмним фільтром.

Технічні характеристики BMP280:

- Напруга живлення: 1.71V - 3.6V;
- Інтерфейс обміну даними: I2C або SPI;
- Струм споживання в робочому режимі: 2.7uA при частоті опитування 1 Гц;
- Діапазон вимірювання атмосферного тиску: 300hPa-1100hPa ($\pm 0.12\text{hPa}$), що еквівалентно діапазону від -500 до 9000 м над рівнем моря;
- Діапазон вимірювання температури: $-40^{\circ}\text{C} \dots +85^{\circ}\text{C}$ ($\pm 0.01^{\circ}\text{C}$);

- Максимальна частота роботи інтерфейсу I2C: 3.4MHz;
- Максимальна частота роботи інтерфейсу SPI: 10 МГц;
- Розмір модуля: 21 x 18 мм;

До пінів D1 і D2 підключаємо виходи датчика SDA і SCL (рис. 3.1). Живити датчик будемо на пряму від ESP, так як напруга на виході у неї 3.3 В. По-цьому підключаємо пін VIN до живлення 3.3 В на ЕСП і пін ГНД до пину ГНД на платі. На схемі присутній 3 світлодіода для індикації роботи станції. Підключаються світлодіоди згідно схеми. Зелений в пін D0, жовтий в D3 і червоний в D4. Кнопка служить для зручності перезавантаження мікроконтролера.

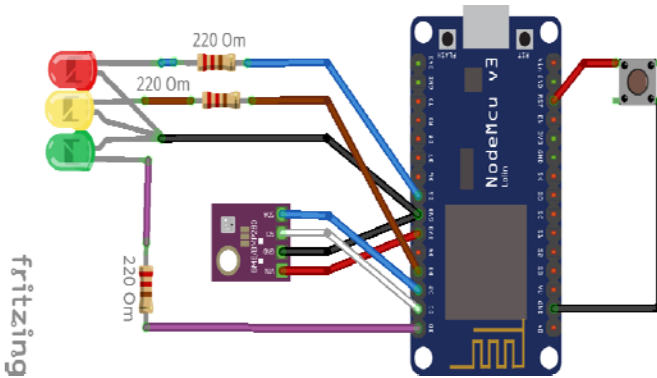


Рисунок 3.1 –Схема підключення датчика BME280

3.2.2 Скетч

Для початку підключаємо бібліотеки.

Лістинг 1.1 – Використовується 5 бібліотек

```
#include <SPI.h>
#include <ESP8266WiFi.h>
#include <Wire.h>
#include <Adafruit_BMP280.h>
#include <Adafruit_Sensor.h>
```

Бібліотека «SPI» - це послідовний синхронний протокол передачі даних використовується мікроконтроллерами для обміну даними з одним або декількома периферійними пристроями.

Бібліотека «ESP8266WiFi» - надається з включеною прошивкою від виробника Espressif. Прошивка допомагає реалізувати роботу з Wi-Fi модулем, виставивши його в ролі модему, за допомогою команд AT, які передаються в послідовний порт. В даному проекті бібліотека використовується для підключення до мережі Wi-Fi і обміном даними з користувачем. Бібліотека «Wire» Дана бібліотека дозволяє вам взаємодіяти з пристроями по шині I2C. Бібліотека «AdafruitBMP280» використовується для роботи з датчиком BMP280. Бібліотека «AdafruitSensor» використовується для коректної роботи бібліотеки «AdafruitBMP280». В змінні *SSID* і *PASSWORD* потрібно вписати назву точки доступу і пароль від неї. Для того що б наша ESP підключилася і розпочала роботу в локальній мережі роутера. Оголошуємо змінні і порт на якому буде працювати пристрій:

Лістинг 1.2 – Змінні і порт на якому буде працювати пристрій

```
const char* ssid = "netis_XXXXX";
const char* password = "password";

WiFiServer server(80);
```

Рядок «WiFiServer server (80); »Використовується для вказівки порту через який буде працювати наш сервер.

У наступному фрагменті відправляємо в порт дані про IP-адресу, запускаємо сервер, гасимо червоний світлодіод і запалюємо зелений тим самим сигналізуючи про те що станція провела діагностику і готова до роботи:

Лістинг 1.3 – Перевірка стану

```
Serial.println("");
Serial.println("Есть подключение к сети.");
Serial.println("IP адрес устройства: ");
Serial.println(WiFi.localIP());
server.begin();
digitalWrite(16,1);
digitalWrite(14,0);
```

3.2.3 Створення HTML-сторінки

Створюємо HTML сторінку і працюємо з клієнтом за допомогою функції `client.println`.

Лістинг 1.4 – HTML-сторінка

```
client.println("<!DOCTYPE html><html>");
client.println("<eahd><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">");
client.println("<meta charset='UTF-8'>");
client.println("<link rel=\"icon\" href=\"data:.,\">");

client.println("<style>body { text-align: center; font-family: \"Trebuchet MS\", Arial;}");
client.println("<table { border-collapse: collapse; width:40%; margin-left:auto; margin-right:auto; }");
client.println("<th { padding: 12px; background-color: #0043af; color: white; }");
client.println("<tr { border: 1px solid #ddd; padding: 12px; }");
client.println("<tr:hover { background-color: #bcbcbc; }");
client.println("<td { border: none; padding: 12px; }");
client.println("<.sensor { color:black; font-weight: bold; background-color: white; padding: 1px; }");

client.println("</style></head><body><h1>Метеостанция</h1>");
client.print(ssid);
client.println("<table><tr><th>Параметр</th><th>Показания</th></tr>");
client.println("<tr><td>Температура</td><td><span class=\"sensor\">");
client.println("<bmp.readTemperature()");
client.println("<*C</span></td></tr>");
client.println("<tr><td>Температура</td><td><span class=\"sensor\">");
client.println("<bmp.readTemperature()+273);
client.println("<K</span></td></tr>");
client.println("<tr><td>Давление</td><td><span class=\"sensor\">");
client.println("<bmp.readPressure()+0.0075);
client.println("<мм рт.ст.</span></td></tr>");
client.println("<tr><td>Приблизительная высота над уровнем моря</td><td><span class=\"sensor\">");
client.println("<bmp.readAltitude(SEALEVELPRESSURE_HPA));
client.println("<m</span></td></tr>");
```

Це фрагмент коду є нашою сторінкою в браузері.

4. ЛАБОРАТОРНА РОБОТА №4 „ПЕРЕДАЧУ ДАНИХ МІЖ ДАТЧИКАМИ І КОМП'ЮТЕРОМ ЧЕРЕЗ ІНТЕРНЕТ ЗА ДОПОМОГОЮ WIFI-ТРАНСІВЕРА ESP8266”

Мета роботи – ознайомлення з WiFi-трансівером ESP8266 та здійснити передачу даних між датчиками і комп'ютером через Інтернет.

4.1 Короткі теоретичні відомості

ESP8266 - це модуль Wi-Fi (рис. 4.1), який відмінно підходить для проектів автоматизації з використанням Інтернет. ESP8266 - це модуль Wi-Fi, який дозволяє управляти входами і виходами, як якщо б ви це робили з Arduino, але в ньому є Wi-Fi.

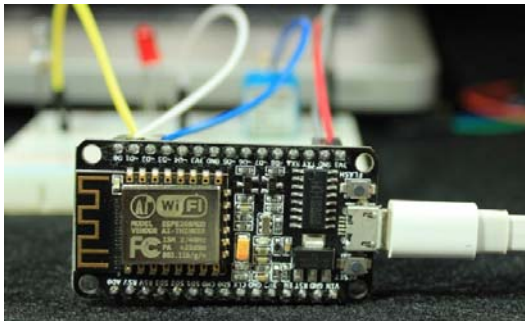


Рисунок 4.1 – Модуль Wi-Fi

ESP8266 має наступні технічні характеристики: Wi-Fi Direct (P2P), програмна точка доступу; інтегрований стек протоколів TCP/IP; вбудований 32-бітний процесор з низьким енергоспоживанням.

За допомогою цього модуля можна:

- 1) створити веб-сервер,
- 2) відправляти HTTP-запити,
- 3) керуючі виходи,
- 4) читати входи і переривання,
- 5) відправляти електронну пошту.

Для створення веб-сервера з ESP8266 викладено інструкцію: <https://randomnerdtutorials.com/esp8266-web-server/>. Наступником

ESP8266 є ESP32, який є двоядерним і поєднує в собі Wi-Fi і Bluetooth.

NodeMCU v3 - нова версія отлаждочной плати на базі Wi-Fi модуля ESP8266 ESP-12E.

Чіпом ESP8266 можна управляти з локальної WiFi-мережі або через інтернет (рис. 4.2). У модуля ESP-01 є GPIO-контакти, які можна запрограмувати на включення/вимикання світлодіода або реле. Модуль можна запрограмувати за допомогою конвертера USB-TTL через контакти для послідовної комунікації (RX і TX).

Підключення ESP8266 до программатору.

Для програмування ESP8266 можна використовувати не тільки конвертер USB-TTL, але і плату Arduino. Нижче представлено 3 способи, за допомогою яких можна програмувати ESP8266 - виберіть той, який відповідає Вашому варіанту або зазначено викладачем.

Програмування ESP8266 за допомогою конвертера USB-TTL з контактом DTR (рис. 4.3). Якщо ви використовуєте конвертер USB-TTL з контактом DTR, то майте на увазі, що монітор порту в цьому випадку працювати не буде. Схема підключення контактів показана в таблиці 4.1.

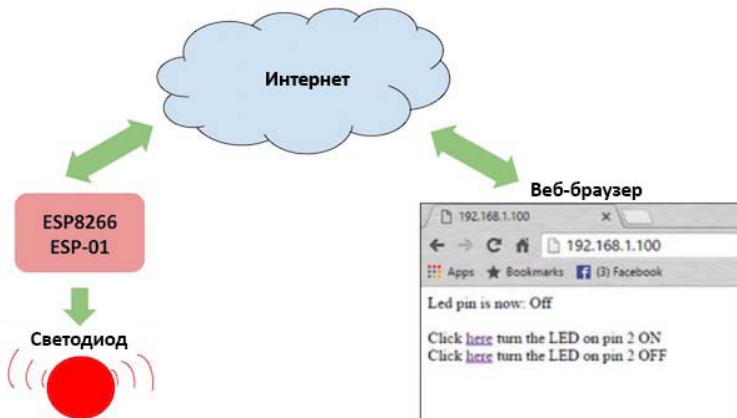


Рисунок 4.2 – Передачу даних між датчиками і комп'ютером через Інтернет за допомогою WiFi-трансивера ESP8266

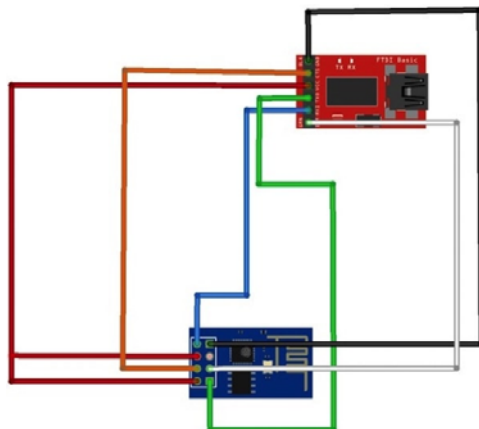


Рисунок 4.3 – Схема для програмування ESP8266 за допомогою конвертера USB-TTL з контактом DTR

Таблиця 4.1 – Схема підключення контактів ESP8266 та конвертера USB-TTL (спосіб 1)

Конвертер USB-TTL		Модуль ESP8266 ESP-01
GND	<--->	GND
TX	<--->	RX
RX	<--->	TX
RTS	<--->	RST
DTR	<--->	GPIO0

За допомогою конвертера USB-TTL без контакту DTR.

При використанні конвертера USB-TTL без контакту DTR формувати запис прошивки доведеться вручну. Для цього знадобиться дві кнопки (рис. 4.4). Схема підключення контактів показана в таблиці 4.2.

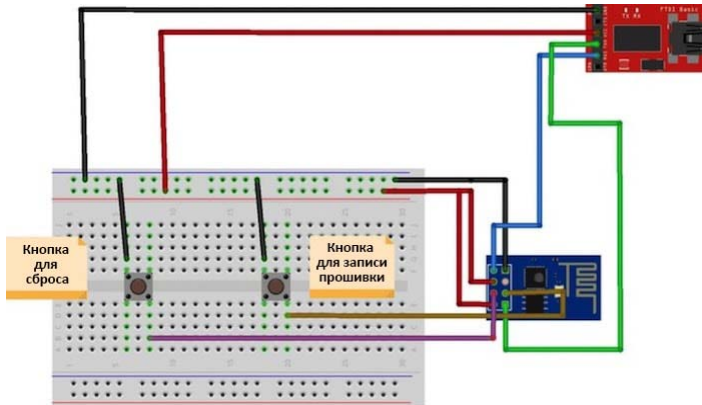


Рисунок 4.4 – Схема для програмування ESP8266 за допомогою конвертера USB-TTL без контакту DTR

Таблиця 4.2 – Схема підключення контактів ESP8266 та конвертера USB-TTL (спосіб 2)

Конвертер USB-TTL		Модуль ESP8266 ESP-01
GND	<--->	GND
TX	<--->	RX
RX	<--->	TX
Кнопка скидання	<--->	RST
Кнопка для запису прошивки	<--->	GPIO0

Спочатку чіп потрібно переключити в режим завантаження прошивки. Для цього натисніть на кнопку, яка на картинці зображено як «кнопка для запису прошивки» (рис. 4.4). Тримайте її натиснутою, а потім один раз натисніть на кнопку, зображену як «кнопка для скидання». Тепер кнопку для запису прошивки можна відпустити - чіп ESP8266 переключено в режим запису прошивки та можливо завантажувати скетч.

Програмування ESP8266 за допомогою плати Arduino Uno.

Для запису прошивки на чіп ESP8266 можна також скористатися платою Arduino Uno (рис. 4.5). Щоб переключити чіп в режим запису прошивки, дотримуйтесь тією ж процедурою, що і в

розділі вище - затисніть кнопку для запису прошивки, один раз натисніть на кнопку для скидання, а потім відпустіть кнопку для запису прошивки.

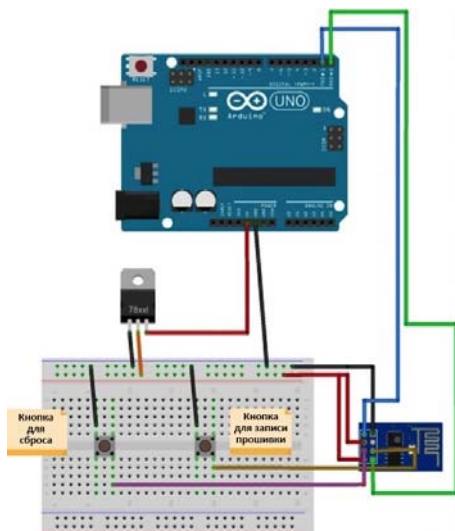


Рисунок 4.5 – Схема для програмування ESP8266 за допомогою плати Arduino Uno

Таблиця 4.3 – Схема підключення контактів ESP8266 та плати Arduino Uno (спосіб 3)

Arduino		Модуль ESP8266 ESP-01
GND	<--->	GND
TX	<--->	TX
RX	<--->	RX
Кнопка скидання	<--->	RST
Кнопка для запису прошивки	<--->	GPIO0

4.2 Хід роботи

4.2.1 Завантаження коду на ESP8266

Підключивши ESP8266 до програмує пристрою одним із способів, описаних вище, відкрийте IDE Arduino і клікніть по

Інструменти> Плата> Generic ESP8266 Module (Tools> Board> Generic ESP8266 Module).

Тепер скопіюйте в IDE Arduino код нижче і натисніть на кнопку «Завантаження». Поміняйте в коді напис YOUR_SSID на SSID вашої точки доступу WiFi, а також напис YOUR_PASSWORD на пароль, який ви використовуєте для своєї точки доступу. Потім скопіюйте скетч (лістинг 1.1).

Лістинг 1.1 – Завантаження коду на ESP8266

```
#include <ESP8266WiFi.h>
2
3 const char* ssid = "YOUR_SSID"; // для SSID точки доступа
4 const char* password = "YOUR_PASSWORD"; // для пароля к точке доступа
5
6 int ledPin = 2; // контакт GPIO2 на ESP8266
7 WiFiServer server(80); // порт веб-сервера
8
9 void setup() {
10   Serial.begin(115200);
11   delay(10);
12
13   pinMode(ledPin, OUTPUT);
14   digitalWrite(ledPin, LOW);
15
16   // подключаемся к WiFi-сети:
17   Serial.println();
18   Serial.println();
19   Serial.print("Connecting to "); // "Подключение к "
20   Serial.println(ssid);
21
22   WiFi.begin(ssid, password);
23
24   while (WiFi.status() != WL_CONNECTED) {
25     delay(500);
26     Serial.print(".");
27   }
28   Serial.println("");
29   Serial.println("WiFi connected"); // "Подключение к WiFi
30                                   // выполнено"
31
32   // запускаем сервер:
33   server.begin();
34   Serial.println("Server started"); // "Сервер запущен"
```

```

35
36 // печатаем IP-адрес:
37 Serial.print("Use this URL to connect: "); // "Используем этот URL
38                                     // для подключения: "
39 Serial.print("http://");
40 Serial.print(WiFi.localIP());
41 Serial.println("/");
42 }
43
44 void loop() {
45 // проверяем, подключен ли клиент:
46 WiFiClient client = server.available();
47 if (!client) {
48     return;
49 }
50
51 // ждем, когда клиент отправит какие-нибудь данные:
52 Serial.println("new client"); // "новый клиент"
53 while(!client.available()){
54     delay(1);
55 }
56
57 // считываем первую строчку запроса:
58 String request = client.readStringUntil('\r');
59 Serial.println(request);
60 client.flush();
61
62 // обрабатываем запрос:
63 int value = LOW;
64 if (request.indexOf("/LED=ON") != -1) {
65     digitalWrite(ledPin, HIGH);
66     value = HIGH;
67 }
68 if (request.indexOf("/LED=OFF") != -1){
69     digitalWrite(ledPin, LOW);
70     value = LOW;
71 }
72
73 // выставляем значение на ledPin в соответствии с запросом:
74 digitalWrite(ledPin, value);
75
76 // возвращаем ответ:
77 client.println("HTTP/1.1 200 OK");
78 client.println("Content-Type: text/html"); // "Тип контента:
79                                     // text/html "

```

```

80 client.println(""); // не забуваємо це...
81 client.println("<!DOCTYPE HTML>");
82 client.println("<html>");
83
84 client.print("Led pin is now: "); // "Контакт світодиода тепер
85 // в стані: "
86
87 if(value == HIGH) {
88   client.print("On"); // "Вкл"
89 } else {
90   client.print("Off"); // "Викл"
91 }
92 client.println("<br><br>");
93 client.println("Click <a href=\"/LED=ON\">here</a> turn the LED on pin 2
ON<br>"); // "Клікніть тут, щоб увімкнути світодиод
94 // на контакті 2"
95 client.println("Click <a href=\"/LED=OFF\">here turn the LED on pin 2 OFF<br>");
// "Клікніть тут, щоб вимкнути світодиод
96 // на контакті 2"
97
98 client.println("</html>");
99
100 delay(1);
101 Serial.println("Client disconnected"); // "Клієнт відключений"
102 Serial.println("");
103 }

```

Далі відкрийте монітор порту, а потім через веб-браузер відкрийте URL, показаний в моніторі порту. Підключіть контакт GPIO2 на ESP8266 до довгій ніжці (анода) світлодіода. Тепер можна управляти світлодіодом через Інтернет.

4.2.2 Перехід ESP8266 в автономний режим

Приберіть всі дроти, які використовувалися для завантаження прошивки. Потім підключіть LM1117 - це регулятор напруги, який видає 3,3 вольт (рис. 4.6). Це дозволить модулю ESP8266 ESP-01 працювати в автономному режимі.

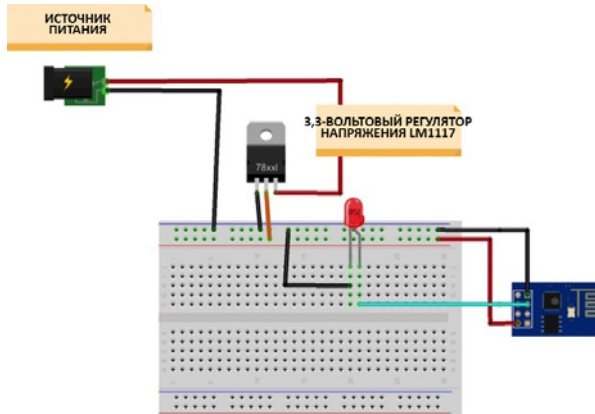


Рисунок 4.6 – ESP8266в автономному режиму

2.3 Підключення ESP8266 до інтернету

У такому вигляді модулем ESP8266 можна управляти тільки з локальної WiFi-мережі. Щоб їм можна користуватися для управління пристроями через інтернет, на роутері потрібно виконати переадресацію портів. Для цього знайдіть IP-адресу вашої системи, вписавши в терміналі команду `ifconfig` або зайшовши на сайт <http://www.whatsmyip.org/>. Скопіюйте свій IP-адресу. Тепер відкрийте налаштування роутера і перейдіть до налаштувань переадресації. Введіть дані в поля для службового порту і IP-адреси. Службовий порт - це порт, зазначений у коді Arduino (80): `WiFiServer server (80);` // порт веб-сервера. IP-адреса - це адреса, знайдений вами в абзаці вище. Інші налаштування залиште за умовчанням. Тепер перейдіть в браузер і введіть в адресному рядку наступну адресу: `xxx.xxx.xx.xx:80`. В результаті повинна відкритися сторінка для управління, наприклад світлодіодом або іншим датчиком.

4.3 Контрольні запитання

1. Яке призначення WiFi-трансівера ESP8266?
2. Які технічні характеристики має ESP8266?
3. Які відмінності між ESP8266 і ESP32?
4. Як можливо виконати програмування ESP8266?
5. Як здійснити передачу даних між датчиками і комп'ютером через Інтернет за допомогою WiFi-трансівера ESP8266?

5. ЛАБОРАТОРНА РОБОТА №5 „ДОСЛІДЖЕННЯ У МІКРОЕЛЕКТРОНІЦІ ЗА ДОПОМОГОЮ ВІРТУАЛЬНИХ ВІМІРЮВАЛЬНИХ ПРИЛАДІВ”

Мета роботи – ознайомлення з можливостями та основними принципами роботи вимірювання параметрів приладів електроніки.

Віртуальні вимірювальні прилади мають сприяти формуванню системних знань про об’єкт, що вивчається. Для розв’язання поставленого завдання використовувалася лабораторна платформа NI ELVIS II, віртуальні прилади, що діють у середовищі NI LabVIEW.

Вимірювання напруги здійснюється віртуальним вольтметром з високоомним входом, вимірювання струму робиться шляхом вимірювання напруги на резисторі, включеному послідовно з досліджуваним діодом. Відповідні сигнали подаються на фізичні канали платформи NI ELVIS II. Для перемикання струмових діапазонів у схемі вимірів передбачений магазин резисторів з номіналами 100 Ом, 100 КОм, 100 МОм, для швидкої зміни зразків передбачені чотири контактних ZIF термінали. Перемикання між резисторами і терміналами здійснюються мікрореле шляхом подавання в ланцюг живлення реле логічної одиниці (5 В).

Для початку роботи необхідно включити комп’ютер і запустити віртуальний прилад у середовищі NI LabVIEW (рис. 5.1).

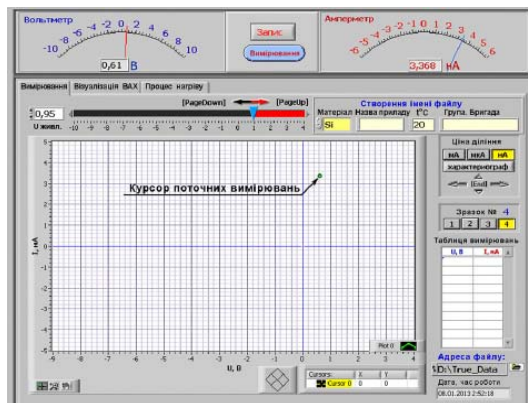


Рисунок 5.1 – Лицева панель віртуального приладу для дослідження діодів (режим вимірювань)

Перемикання між вкладками «Демо ВАХ», «Ім'я файлу», «Блок вимірювань», «Візуалізація ВАХ», «Контроль нагріву» дозволяє вибирати режими роботи віртуального приладу. Після спостереження ВАХ різних діодів формується файлу, у який записуватимуться результати вимірювань (рис. 5.2). Після заповнення полів конструктора (матеріал, тип приладу, температура) створюється «мовляче» ім'я файлу, у якому відбиті умови експерименту. Переміщуючи курсором миші движок напруги на лицевій панелі приладу, можна спостерігати переміщення курсора поточних вимірювань на віртуальному аналізі характеріографа в координатах струм-напруга. Натиснення кнопки «Вимірювання» фіксує пару значень струм напруга. Кожна фіксація супроводжується звуковим сигналом і заповненням рядка в демонстраційній таблиці вимірювань. Закінчення вимірювань натисненням кнопки «Запис» завершує формування файлу з результатами вимірювань.

мкА	В	Si
01.12.2010	18:38	
t20°C	U (В)	I (мкА)
заголовок	-7,016002	-2,845481
файлу	-7,011485	-1,893760
	-7,010840	-0,896874
	-7,000194	-0,500056
	-6,970513	-0,303260
	-6,542079	-0,083880
	-6,491105	-0,090332

Рисунок 5.2 – Фрагмент файлу вимірювань із заголовком

Перемикання струмового діапазону (мА, мкА, нА) і вибір терміналу із зразком, що вивчається, здійснюється кліком миші на кнопках на панелі віртуального приладу. Результат перемикання автоматично відображається в найменуваннях одиниць вимірювань цифрового амперметра й осі струмів на графіку ВАХ, а також в імені і в заголовку файлу з результатами вимірювань.

Функціональне призначення елементів управління лицеві панелі приладу у вигляді підказки може бути викликане через пункт меню «Help» середовища LabVIEW або гарячими клавішами. Під час зняття ВАХ є можливість ручного управління напругою на $p-n$

переході (двигком на лицевій панелі віртуального приладу), що дозволяє сконцентрувати увагу учня на русі поточних координат вимірювань, варіювати їх крок, робити частішу фіксацію точок на ділянках характеристики з різкими змінами.

Інтерфейс, реалізований за допомогою віртуального приладу, дозволяє в реальному часі безпосередньо спостерігати рух курсору в координатах струм-напруга за треком ВАХ. Стрілочна і цифрова інтерпретація вимірювальних приладів (вольтметр і амперметр), а також таблична інтерпретація результатів вимірювання збільшують наочність процесу вимірювань і формування результатів. Крім того, це зв'язує інтерфейс віртуального лабораторного приладу з попередньою методикою, доступною раніше на звичайних стрілочних вимірювальних приладах. В аналітичному блоці роботи надається можливість засобами віртуального приладу робити порівняння раніше знятих ВАХ. Наразі можна простежити вплив таких чинників, як ширина забороненої зони, температура, площа p - n переходу. Визначення впливу температури на напругу відкриття p - n переходу представлено на рис. 5.3. ВАХ Ge і Si p - n переходів у різних струмових діапазонах представлені на рис. 5.4 і 5.5.

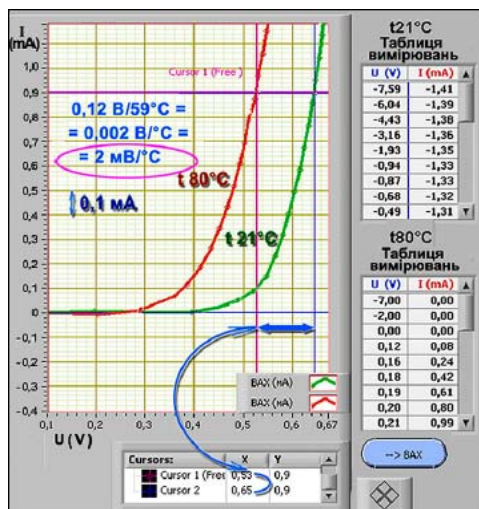


Рисунок 5.3 – Вплив температури на напругу відкриття p - n переходу

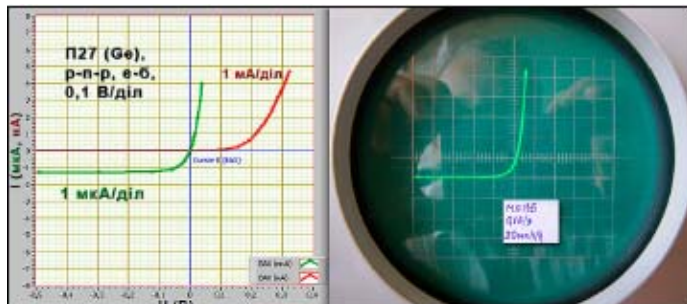


Рисунок 5.4 – Порівняння ВАХ Ge діоду в різних струмових діапазонах

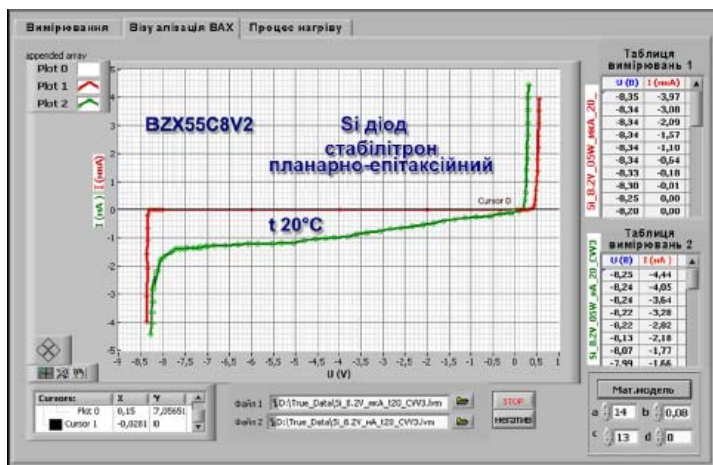


Рисунок 5.5 – ВАХ кремнієвого стабілітрона в діапазоні струмів «мкА» і «нА»

Важливим доповненням до порівнюваних характеристик є можливість представлення теоретичної ВАХ, керованої в динамічному режимі за допомогою зміни параметрів моделі (ширина забороненої зони, площа $p-n$ переходу, температура), що відразу ж відбивається на вигляді теоретичної характеристики.

6. ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Поняття вимірювальних технологій. Основні тренди сучасних вимірювальних тих-нологій. Причини виникнення комп'ютерних вимірювальних технологій. Призначення і переваги КСВ. Області застосування КСВ. Основа КСВ.

2. Апаратна основа КСВ. Вимірювальні системи. Підходи до реалізації вимірювальних систем. Приладно-модульні, функціонально-модульні і гібридні системи. Приклади ін-терфейс.

3. Інтерфейси приборно-модульних вимірювальних систем. Інтерфейс КОП (GPIB). Основні характеристики. Відмінні особливості.

4. Інтерфейси приборно-модульних вимірювальних систем. Інтерфейси RS-232, RS-423, 422, 485. Основні характеристики. Гідності й недоліки. Порівняння характеристик.

5. Інтерфейси функціонально-модульних вимірювальних систем. Інтерфейси VME і VXI. Фізичне виконання пристроїв. Призначення, основні характеристики. Області застосування.

6. Інтерфейси функціонально-модульних вимірювальних систем. Інтерфейси CompactPCI, PXI і PXIe. Фізичне виконання пристроїв. Призначення, основні характеристики. Основні області застосування.

7. Проблеми розробки вимірювальних систем. Інтерфейс LXI. Вимоги базової спеці-ції LXI. Конструктивне виконання пристроїв. Характеристики, розширені можли-ності і переваги LXI. Області застосування.

8. Інтерфейс ATCA і AXIe. Конструктивне виконання пристроїв. Основні характери-стики, особливості та розширення AXIe.

9. Технології збору даних. Узагальнена структура вимірювальної системи на основі DAQ. Конструктивне виконання пристроїв DAQ. Компоненти, інтерфейси і функціональні можливості.

10. Технології збору даних. Програмне забезпечення DAQmx. Сполучення DAQ і засобів розробки. NI MAX.

11. Програмне забезпечення КСВ. Завдання програмного забезпечення КСВ. Вимоги до ПЗ КСВ. Лідери ринку ПЗ КСВ.

12. Основи побудови ПЗ КСВ в LabVIEW. Основні концепції програмування в LabVIEW. Основні алгоритмічні структури.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Войтенко С.С., Герасимов С.В. Використання комп'ютерних засобів вимірювання при підготовці фахівців-метрологів. / Системи обробки інформації. 2009. № 6(80). С. 156-158.
2. Закон України «Про метрологію та метрологічну діяльність» / Відомості Верховної Ради (ВВР), 2014, № 30, ст.1008. (Із змінами, внесеними згідно із Законом № 124-VIII від 15.01.2015, ВВР, 2015, № 14, ст.96).
3. Проекування комп'ютеризованих вимірювальних систем і комплексів: навчальний посібник / М. Паламар, М. Стрембіцький, А. Паламар. – Тернопіль : ТНТУ, 2018. – 150 с.
4. Виртуальные средства измерений : учеб.-метод. пособие / А. Л. Гурский, В. Т. Ревин. – Минск : БГУИР, 2016. – 67 с.
5. Дорожовець М. та ін. Основи метрології та вимірювальної техніки: Підручник у 2 т. / М.Дорожовець, В.Мотало, Б.Стадник, В.Василук, Р.Борек, А.Ковальчик, за ред. Б.Стадника. – Львів: Видавництво національного університету «Львівська політехніка». 2005, - т.1. Основи метрології. – 532 с.
6. Дорожовець М. та ін. Основи метрології та вимірювальної техніки: Підручник у 2 т. / М.Дорожовець, В.Мотало, Б.Стадник, В.Василук, Р.Борек, А.Ковальчик, за ред. Б.Стадника. – Львів: Видавництво національного університету «Львівська політехніка». 2005, - т.2. Вимірювальна техніка. – 656 с.
7. Раннев Г.Г. Информационно-измерительная техника и электроника: учебник для вузов / Г.Г. Раннев. – М.: издательский центр «Академия», 2006. – 512 с.
8. Ширяев В.В. Компьютерные измерительные средства (КИС): учеб. пособие / В.В. Ширяев. – Томск: ТПУ, 2008. – 190 с.
9. PXI. Высокопроизводительная модульная платформа, предназначенная для создания автоматизированных и испытательных комплексов / National Instruments [Электронный ресурс]. – 2015. – Режим доступа: ni.com.
10. UART и USART. COM-порт. Часть 1. – [Електронний ресурс] режим доступу до ресурсу: http://www.rotr.info/electronics/mcu/arm_usart.htm
11. Описание шины CAN – [Електронний ресурс] режим доступу до ресурсу: http://itt-ltd.com/reference/ref_can.html

12. Розподілені мікропроцесорні системи: конспект лекцій [Електронний ресурс]: для підготовки докторів філософії в галузі знань 17 Електроніка та телекомунікація за спеціальністю 171 Електроніка за спеціалізацією «Електронні системи» / КПІ ім. Ігоря Сікорського ; уклад.: Т. О. Терещенко – Електронні текстові данні (1 файл:5544 кбайт). – Київ : КПІ ім. Ігоря Сікорського, 2018. – 192 с.

13. Universal serial bus [Електронний ресурс] – режим доступу до ресурсу: <http://www.usb.org>

14. Автоматизація виробничих процесів [Текст] : підручник / І. В. Ельперін, О. М. Пупена, В. М. Сідлецький, С. М. Швед ; Нац. ун-т харч. технол. – 2-ге вид., випр. – К. : Ліра-К, 2015. – 378 с.

15. Промислові мережі та інтеграційні технології в автоматизованих системах [Текст] : навч. посіб. / О. М. Пупена, І. В. Ельперін, Н. М. Луцька, А. П. Ладанюк. — К. : Ліра-К, 2011. – 552 с.

16. Промислові контролери [Текст] : навч. посіб. / І. В. Ельперін ; МОН України, НУХТ. – К. : НУХТ, 2003. – 320 с.

17. Boyer, Stuart A. (2010). SCADA Supervisory Control and Data Acquisition. USA: ISA – International Society of Automation. p. 179. ISBN 978-1-936007-09-7.

18. SCADA системи [Електронний ресурс] режим доступу до ресурсу:<http://www.scadasystems.net/>

19. Як вибрати SCADA систему [Електронний ресурс] режим доступу до ресурсу:<http://readonline.com.ua/items/47871-yak-vibrati-scada-sistemu/>

20. Воеводін С. В. Віртуальні вимірювальні прилади у лабораторному практикумі з твердотільної електроніці. / Комп'ютер у школі та сім'ї . 2016. № 1. С. 31-35

Додаток А**Приклад оформлення титульної сторінки**

Міністерство освіти та науки України

Національний університет «Запорізька політехніка»

Кафедра МiНЕ

ЗВІТ

З ЛАБОРАТОРНОЇ РОБОТИ №__

(тема роботи)

з дисципліни „КОМП'ЮТЕРНІ СИСТЕМИ ВИМІРЮВАНЬ“

Виконав:

ст. гр. РТ_

(Ініціали, Прізвище)

Прийняв:

(посада)

(Ініціали, Прізвище)

20__ р.