

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,  
управління та адміністрування

Кафедра інформаційних  
технологій

**Бакалаврська кваліфікаційна робота**

на тему: Розробка IoT-проекту метеостанції

Виконав студент 4 курсу групи К-41

Напряму 6.050101 2 комп'ютерні  
науки,

Киріак Дмитро Євгенович

---

Керівник асистент

Штефан Наталія Зінов'ївна

Консультант: к.т.н., доцент

Великодний Станіслав Сергійович

Рецензент: к.георг.н., доцент

Бояринцев Євген Львович

Одеса 2019

## ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ.....	6
ВСТУП.....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1 Опис предметної області.....	11
1.2 Характеристика об'єкту розробки.....	12
1.3 Аналіз існуючих аналогів.....	12
1.3.3 Дисплей OLED 128x64 I2C.....	18
1.3.4 Мікроконтролер.....	20
1.3.5 Додаткове обладнання.....	22
1.4 Інструментальні засоби розробки.....	22
2 ПРОЕКТНА ЧАСТИНА.....	27
2.1 Моделювання системи.....	27
2.2 Налаштування середовища розробки Arduino IDE.....	31
2.2.1 Установка середовища розробки.....	31
2.2.2 Управління бібліотеками.....	33
2.2.3 Монітор послідовного порту.....	35
2.2.4 Проектування схем пристрою.....	37
3 ПРОГРАМНА ЧАСТИНА.....	40
3.1 Тестове підключення плати.....	40
3.2 Підключення датчика температури і вологості DHT11.....	42
3.3 Підключення датчика тиску BMP180.....	46
3.4 Робота з рідкокристалічним дисплеєм OLED 128x64 I2C.....	49
3.5 Зборка метеостанції.....	52
3.6 Налаштування Wi-Fi модуля.....	55
ВИСНОВКИ.....	58
ПЕРЕЛІК ПОСИЛАНЬ.....	59

## **СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ**

АЦП – Аналого-цифровий перетворювач

OLED – Organic Light-Emitting Diode

LCD – Liquid Crystal Display

UML – Unified Modeling Language

USB – Universal Serial Bus

Терміни

Піни – роз’єми на платі Ардуіно

Шілд – Shield уніфіковані сторонні модулі

Adafruit GFX – бібліотека для роботи з OLED у Arduino IDE

Adafruit SSD1306 – бібліотека для роботи з OLED у Arduino IDE

Arduino – торгова марка апаратно-програмних засобів для побудови простих систем автоматики і робототехніки

Arduino IDE – середовище розробки

Arduino Uno R3 – модель мікроконтролера

BMP – сімейство датчиків атмосферного тиску

COM – послідовний порт

C/C ++ – мови програмування

DHT – сімейство датчиків для визначення температури

IoT – інтернет речей

I2C – інтерфейс мікроконтролера

Fritzing – програмне забезпечення з відкритим кодом для віртуального моделювання електричних ланцюгів, схем і електронного обладнання

GND – пін мікроконтролера (земля);

GPIO1-GPIO16 – виходи на платі, які відповідають цифровим 0-15

Arduino

NodeMCU – мікроконтролер на базі чипу ESP8266

Leonardo – модель мікроконтролера Arduino

OzOLED – бібліотека для роботи з OLED у Arduino IDE

SDA – пін мікроконтролера NodeMCU

SCL – пін мікроконтролера NodeMCU

Seeed OLED Display 128 \* 64 library – бібліотека для роботи з OLED у

Arduino IDE

SPI – інтерфейс мікроконтролера

VCC – пін мікроконтролера NodeMCU

Vin – висновок роз'єму живлення, «повітря»

Yun – модель мікроконтролера Arduino

Use case – варіанти використання у мові UML

Wire – бібліотека Ардуіно

Zero – модель мікроконтролера Arduino

Умовні позначення

В – Вольт, напруга

гПа – гектопаскаль, тиск

Мм – мілі метр, довжина

мкА – мікро ампер, ток

С – Гра́дус Це́льсія, одиниця температури

## ВСТУП

Розвиток мікроелектроніки та широке її застосування в промисловості, пристроях і системах контролю та управління різноманітними об'єктами і процесами є в нинішній час одним з головних напрямків науково-технічного прогресу.

Сучасні мікропроцесори значно зменшилися в розмірах у порівнянні зі своїми минулими аналогами і стали дешевше. Ці пристрої тепер розташовуються на одній друкованій платі, завдяки чому стало можливим їх вставляти (вбудовувати) в різні місця, раніше для комп'ютерів недоступні. Тоді і з'явився термін вбудованих систем.

Платформа Arduino являє собою сімейство мікроконтролерів на базі різних процесорів. Всі мікроконтролери програмуються на мові C/C ++ в середовищі розробки Arduino IDE. Платою можна керувати з комп'ютера, або запрограмувати її й після від'єднання від комп'ютера вона буде працювати автономно. Модель Arduino Uno R3 вважають базовою платою Arduino [1]<sup>1)</sup>.

Проте існують і інші моделі Arduino (Leonardo, Zero, Yun), а також інші пристрої для програмування котрих також використовується мова Arduino.

Велика частина плат Arduino (крім маленьких, таких як micro, pro mini) мають ідентичне розташування висновків (пинов, pins) і дозволяють підключати уніфіковані сторонні модулі, звані Шілд (Shield). На всіх платах є набір цифрових і аналогових пинов, а так само інтерфейси SPI і I2C. Для роботи зі сторонніми модулями в середовищі розробки є менеджер бібліотек, куди зібрані найбільш часто використовувані для Arduino бібліотеки [2]<sup>2)</sup>.

---

<sup>1)</sup> [1] Інформаційний портал з Ардуіно. URL: <https://arduino.ua/art2-istoriya-sozdaniya-arduino>. (дата звернення 13.03.2019)

<sup>2)</sup> [2] Офіційний сайт Arduino. URL: <https://www.arduino.cc/>.(дата звернення 13.03.2019)

Пристрої на базі Arduino стають все більш популярними, наприклад, розробляють механічні руки, гірокоптери, датчики серцебиття та багато іншого. Пристрої на базі Arduino застосовуються в різних галузях, як в лікувальних, так і в розважальних.

Темою диплому обрана розробка IoT-проекту метеостанції на базі Arduino. Вона буде визначати температуру, тиск і вологість.

Загальні характеристики кваліфікаційної роботи:

- повний обсяг сторінок пояснювальної записки – 60
- кількість рисунків – 31
- кількість таблиць – 0
- кількість посилань – 19

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Опис предметної області

Більшість мікроконтролерів мають підтримку USB (вбудовану або винесену в окремий перетворювач) і підключаються до операційної системи як послідовний порт. Послідовний порт (COM порт) – спеціальний порт для послідовної передачі даних між пристроями. Може бути апаратним (спеціальний COM роз'єм на материнській платі ПК), або емулюється поверх іншого апаратного протоколу (наприклад, поверх USB, як у випадку з Arduino).

Послідовний порт використовується для завантаження програми на мікроконтролер, а так само може використовуватися для взаємодії ПК і програми для мікроконтролера (в Arduino IDE по іншому ця програма називається "скетч") [2]<sup>1)</sup>.

Скетч для мікроконтролера Arduino являє собою файл з розширенням .ino, що містить код на мові C / C ++ /Код складається з декількох основних блоків:

- підключення заголовних файлів бібліотек;
- оголошення глобальних змінних, констант, макроозначень;
- визначення власних функцій, структур і класів [3]<sup>2)</sup>.

Функція `setup()` – викликається один раз на початку роботи програми, в ній повинно здійснюватися ініціалізація контролера - настройка пинов, запуск послідовного порту, SPI і I2C інтерфейсів.

Функція `loop()` – викликається мікро контролером в нескінченному циклі, в ній виробляється основна робота.

---

<sup>1)</sup> [2] Офіційний сайт Arduino. URL: <https://www.arduino.cc/>.(дата звернення 13.03.2019)

<sup>2)</sup> [3] Інформаційний довідник з Ардуіно. URL: <http://arduino-diy.com/>.(дата звернення 13.03.2019)



Після завантаження на мікроконтролер програма зберігається навіть після відключення Arduino, до тих пір, поки вона не буде переписана новою програмою [3]<sup>1)</sup>.

Мета роботи – розробка метеостанції на базі мікроконтролера Arduino.

## **1.2 Характеристика об'єкту розробки**

Об'єкт розробки являє собою метеостанцію за базі мікроконтролера Arduino. До системи поставлені наступні вимоги:

- вимірювання температури і вологості повітря у приміщенні;
- вимірювання атмосферного тиску;
- можливість відображення отриманих показників на жидкокристалічний дисплей;
- до пристрою потрібно підключити і налаштувати (якщо нема вбудованого у саму плату) Wi-Fi модуль для подальшої передачі інформації для обробки користувачем;
- архітектура системи, апаратне і програмне забезпечення повинні забезпечувати подальшу розширюваність системи для додавання нових датчиків і нових можливостей.

## **1.3 Аналіз існуючих аналогів**

На даний час існує багато різновидів домашніх метеостанцій в рамках IoT-проектів. Під час огляду предметної області і аналогів проекту проектування було встановлено, що більшість з них побудовані на базі Arduino Uno, що не є актуальним, так як зараз існує багато інших мікроконтролерів, які менші за розмірами і більш функціональні.

---

<sup>1)</sup> [3] Інформаційний довідник з Ардуіно. URL: <http://arduino-diy.com/>.(дата звернення 13.03.2019)

Крім того, якщо потрібно, щоб метеостанція мала зв'язок з сервером і передавала дані, для таких пристроїв використовують додатковий WiFi-модуль [4]<sup>1)</sup>.

Під час вибору основи для проекту – мікроконтролера враховувались такі фактори, як ціна, розміри, функціональність, можливість програмування у Arduino IDE, вбудований WiFi-модуль. Було прийнято рішення використовувати мікроконтролер NodeMCU на базі WiFi-модуля ESP8266.

Для того, щоб оптимально скомпонувати пристрій, необхідно провести порівняльний аналіз між шилдами (додатковими модулями) до плати Arduino (і її аналогів), які будуть потрібні для реалізації проекту.

### 1.3.1 Датчики температури

На сьогоднішній день існує безліч датчиків і модулів, які можна використовувати для вимірювання температури та інших показників, пов'язаних з підтриманням оптимальної життєдіяльності людини, а також інших речей і організмів. Їх можна використовувати в самих простих метеостанціях, в різних системах контролю за кліматом і в розумному будинку, для підтримки необхідної температури в приміщеннях, на виробництві і в багатьох інших випадках.

Датчики сімейства DHT є найпопулярнішими в колі Arduino. Важливими критеріями тут є простота у використанні і написанні програмного коду, так і відносно недорога вартість. У сімействі DHT виділяють три найпоширеніших датчика: DHT11, DHT22 і DHT21 [5]<sup>2)</sup>.

Перші два датчика зовні трохи схожі один на одного, до того ж підключаються вони теж однаково (рис.1). DHT21 має інший вигляд (рис. 2). Відмінність цього модуля від перших двох полягає в тому, що він має

<sup>1)</sup> [4] Інформаційний блог RoboCraft. URL: <http://robocraft.ru/blog/arduino/>. (дата звернення 17.03.2019)

<sup>2)</sup> [5] Інформаційний ресурс по шилдам до Ардуіно. URL: <https://voltiq.ru/dht11-dht22-and-dht21/>. (дата звернення 19.03.2019)

захисний корпус, що дозволяє використовувати його на вулиці, де цей корпус захистить його від пилу, бруду і дощу.

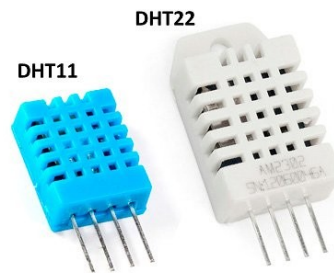


Рисунок 1 – Датчики температури DHT11 і DHT22



Рисунок 2 – Датчик температури DHT21

Тепер слід порівняти модулі за основними показниками: точність і діапазон вимірювання, а також за ціною.

Датчик DHT11:

- визначення вологості в діапазоні 20-80% з точністю  $\pm 5\%$ ;
- визначення температури від  $0^{\circ}\text{C}$  до  $+50^{\circ}\text{C}$  з точністю  $\pm 2^{\circ}\text{C}$ ;
- частота опитування 1 раз в секунду.

Датчик DHT22:

- визначення вологості в діапазоні 0-100% з точністю  $\pm 2\%$ ;
- визначення температури від  $-40^{\circ}\text{C}$  до  $+125^{\circ}\text{C}$  з точністю  $\pm 0.5^{\circ}\text{C}$ ;
- частота опитування 1 раз в 2 секунди.

Датчик DHT21:

- визначення вологості в діапазоні 0-100% з точністю  $\pm 2\%$ ;
- визначення температури від  $-40\text{ }^{\circ}\text{C}$  до  $+80\text{ }^{\circ}\text{C}$  з точністю  $\pm 0.5\text{ }^{\circ}\text{C}$  [5]<sup>1)</sup>.

Порівнюючи ціни на дані модулі, можна відразу виділити низьку ціну на датчик DHT11. Це пов'язано з високим попитом на дані модулі і з їх простим пристроєм.

DHT21 і DHT22 на тлі першого легко можна віднести до більш дорогим: пов'язано це з більшою точністю показань, великим діапазоном вимірювання температур, до того ж у датчика DHT22 є захисний корпус, який оберігає його від забруднення і вологи, що теж грає вагомую роль в складанні ціни [6]<sup>2)</sup>.

Найбільш оптимальним датчиком для домашньої метеостанції буде DHT11, оскільки він дешевше, займає менше місця, надійний і простий в експлуатації і не вимагає від творця вимірювати рекордно низькі або високі температури, підтримуючи стабільність протягом довгого часу.

### 1.3.2 Датчики тиску

Найбільш поширеними і доступними є датчики тиску від фірми BOSCH: це BMP085, BMP180, BMP280 і інші. Перші два дуже схожі між собою, BMP280 – це більш новий і вдосконалений датчик.

Датчик BMP280 (рис. 3) створений спеціально для додатків, де потрібні малі розміри і знижене споживання енергії. До таких додатків ставляться навігаційні системи, прогноз погоди, індикація вертикальної швидкості та інші [7]<sup>1)</sup>.

---

<sup>1)</sup> [5] Інформаційний ресурс по шилдам до Ардуіно. URL: <https://volti.ru/dht11-dht22-and-dht21/>. (дата звернення 19.03.2019)

<sup>2)</sup> [6] Інформаційний довідник з Ардуіно. URL: <http://radioschema.ru/el-komponenty/spravochnik>. (дата звернення 20.03.2019)

<sup>1)</sup> [7] Інформаційний ресурс по шилдам до Ардуіно. URL: <https://arduinomaster.ru/datchiki-arduino/datchiki-atmosfernogo-davleniya-bmp280-bmp180-bme280/>. (дата звернення 20.03.2019)



Рисунок 3 – Датчик атмосферного тиску BMP280

Датчик має високу точність, хорошою стабільністю і лінійністю.

Технічні характеристики датчика BMP280:

- габарити 2 x 2,5 x 0,95 мм.
- тиск 300-1100гПа;
- температури від 0С до 65 С;
- підтримка інтерфейсів I2C і SPI;
- напруга живлення 1,7В – 3,6;
- середній струм 2,7мкА;
- 3 режими роботи: режим сну, режим FORCED (проведення вимірювання, зчитування значення, перехід в сплячий режим), режим NORMAL (переклад датчика в циклічну роботу – тобто пристрій самостійно через встановлений час виходить з режиму сну, проводить вимірювання, зчитує показання, зберігає виміряні значення і переходить знову в режим сну).

Датчик атмосферного тиску BMP085 (рис. 4) – датчик фірми Bosch.

Його характеристики:

- вимірюється тиск: від 300гПа до 1100 гПа (від 9000 до – 500 метрів над рівнем моря);
- напруга живлення: від 1.8 до 3.6В;
- робочий струм: 5мкА при 1 опитуванні в секунду;
- точність: 0.5м в найшвидшому режимі і 0.25м в самому точному;

- час вимірювання: до 3мс;
- вбудований термометр.

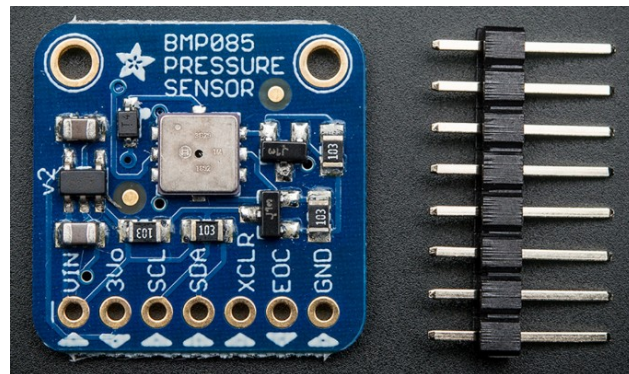


Рисунок 4 – Датчик атмосферного тиску BMP085

Датчик BMP180 (рис. 5) – це дешевий і простий в застосуванні сенсорний датчик, який вимірює атмосферний тиск і температуру [6]<sup>1)</sup>. Використовується зазвичай для визначення висоти і в метеостанціях. Складається пристрій з пьезо-резистивного датчика, термодатчика, АЦП, незалежної пам'яті, ОЗУ і мікроконтролера.

Технічні характеристики датчика BMP180:

- межі вимірюваного тиску 225-825 мм рт. ст.
- напруга живлення 3,3 – 5В;
- струм 0,5мА;
- підтримка інтерфейсу I2C;
- час спрацювання 4,5мс;
- розміри 15 x 14 мм [7]<sup>1)</sup>.

<sup>1)</sup> [6] Інформаційний довідник з Ардуіно. URL: <http://radioschema.ru/el-komponenty/spravochnik>. (дата звернення 20.03.2019)

<sup>1)</sup> [7] Інформаційний ресурс по шилдам до Ардуіно. URL: <https://arduino-master.ru/datchiki-arduino/datchiki-atmosfernogo-davleniya-bmp280-bmp180-bme280/>. (дата звернення 20.03.2019)



Рисунок 5 – Датчик атмосферного тиску BME180

Для об'єкту розробки будемо використовувати датчик атмосферного тиску BME 180.

### 1.3.3 Дисплей OLED 128x64 I2C

У OLED (Organic Light-Emitting Diode) дисплеях використовується технологія в якій світлодіоди самі випромінюють світло без додаткового підсвічування як наприклад в LCD (Liquid Crystal Display) дисплеях.

Дисплей OLED складається з тонкої багатошарової органічної плівки, вміщеній між анодом і катодом. OLED має високий потенціал застосування практично для всіх типів дисплеїв і розглядається в якості кінцевої технології для наступного покоління плоских дисплеїв.

Особливості OLED I2C дисплея (рис. 6):

- не потрібно підсвічування дисплея;
- висока роздільна здатність: 128x64 пікселів;
- кут огляду: більше 160 градусів;
- повністю сумісний з Arduino;
- ультра-низьке енергоспоживання;
- робоча напруга: 3V ~ 5 В постійного струму;

- I2C / ІС інтерфейс, потрібно тільки 2 дроти;
- розмір плати: 2.7см x 2.8см;
- розмір дисплея: 2.7см x 1,95см (0,96 "дюйма) [8]<sup>1)</sup>.

Підключення OLED I2C дисплея:

- VCC – + 5v (+5 вольт);
- GND – GND (земля);
- SDA – pin SDA (pin A4 для Arduino nano V3; pin D2 для NodeMCU V3);
- SCL – pin SCL (pin A5 для Arduino nano V3; pin D1 для NodeMCU V3).

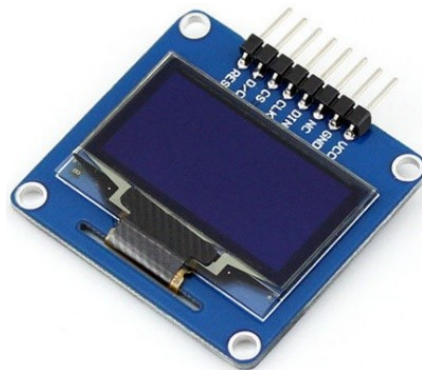


Рисунок 6 – Дисплей OLED I2C

Бібліотеки для OLED I2C дисплея існують від різних авторів, ось кілька:

- від «Adafruit» (потрібна установка обох бібліотек) – «Adafruit GFX» і «Adafruit SSD1306»;
- від Oscar Liang – «OzOLED»;
- «Seeed OLED Display 128 \* 64 library» від Seeed-Studio [8]<sup>1)</sup>.

---

<sup>1)</sup> [8] Інформаційний портал з Ардуіно. Характеристики світодіодів. URL: <https://arduino.ua/cat14-kraniisvetodiodi>. (дата звернення 22.03.2019)

<sup>1)</sup> [8] Інформаційний портал з Ардуіно. Характеристики світодіодів. URL: <https://arduino.ua/cat14-kraniisvetodiodi>. (дата звернення 22.03.2019)



### 1.3.4 Микроконтролер

Плата NodeMCU побудована на базі модуля WiFi ESP8266 (рис. 8) для створення різних пристроїв інтернету речей (IoT). Модуль вміє відправляти і отримувати інформацію в локальну мережу або в інтернет за допомогою Wi-Fi. Недорогий модуль часто використовується для створення систем розумного будинку або роботів Arduino, керованих на відстані [9]<sup>1)</sup>.

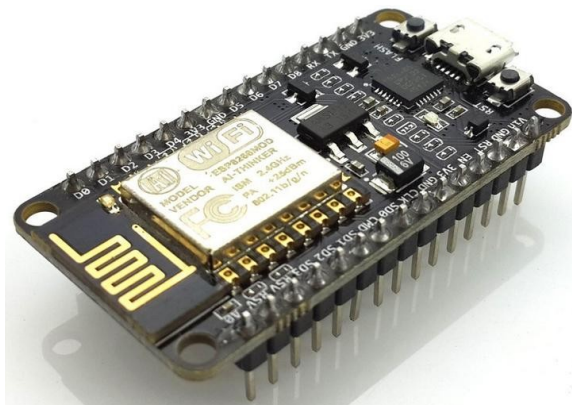


Рисунок 8 – Плата NodeMCU

Для роботи з цією платою була додана підтримка середовища розробки Arduino IDE. Практично всі бібліотеки для звичайної Arduino так само працюють і для цієї плати, однак її терморегулятори відрізняється від терморегулятори плат Arduino. Так, на платі є один аналоговий вхід (A0), на якому працює 10-бітний АЦП. Цифровим пінам 0-15 Arduino відповідають виходи, позначені GPIO1-GPIO16.

Наприклад, якщо в Arduino IDE використовується пін 12, йому буде відповідати контакт, позначений на платі як D6 (GPIO12). Піни GPIO1-GPIO5, GPIO10, GPIO12-GPIO15 підтримують ШІМ [10]<sup>2)</sup>.

<sup>1)</sup> [9] Інформаційний портал з плат Ардуіно. URL: <https://arduinomaster.ru/platy-arduino/esp8266-nodemcu-v3-lua/>. (дата звернення 24.03.2019)

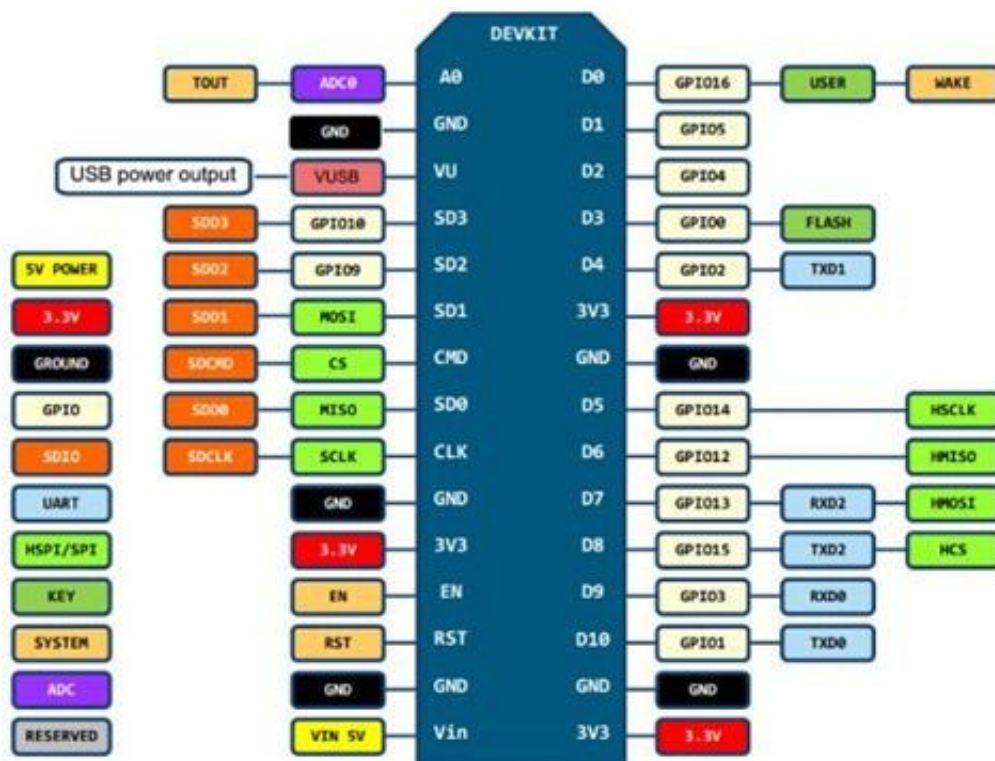
<sup>2)</sup> [10] Інформаційний портал з плат Ардуіно. URL: <https://arduino.ua/prod1804-plata-adaptera-node-mcu-esp8266-v3>. (дата звернення 24.03.2019)

Характеристики плати:

- вбудований стек TCP / IP;
- вбудовані регулятори, і система управління живленням<sup>4</sup>
- номінальна напруга: 3,3 В;
- вхідна напруга: 3,7-20 В;
- максимальний споживаний струм: 220 мА.

Распінровка плата NodeMCU представлено на рисунку 9. Модуль V3 має 11 контактів введення-виведення загального призначення. Крім цього деякі з висновків володіють додатковими функціями:

- D1-D10 – висновки з широтно-імпульсною модуляцією;
- D1, D2 – висновки для інтерфейсу I<sup>2</sup>C / TWI;
- D5-D8 – висновки для інтерфейсу SPI;
- D9, D10 – UART;
- A0 – вхід з АЦП.



## Рисунок 9 – Распіновка NodeMCU

Основні відмінності Ардуіно від ESP8266:

- ESP8266 має більший обсяг флеш-пам'яті, при цьому у ESP8266 відсутня незалежна пам'ять;
- процесор ESP8266 швидше, ніж у Ардуіно;
- наявність Wi-Fi у ESP8266;
- ESP8266 потребує більше струму, ніж для Ардуіно [10]<sup>1)</sup>.

### 1.3.5 Додаткове обладнання

Для розробки простих електронних схем часто використовуються макетні плати. Вони бувають двох основних видів – для пайки і для безпаячного монтажу.

Контакти цієї плати з'єднані всередині особливим чином, як показано на рисунку 10:

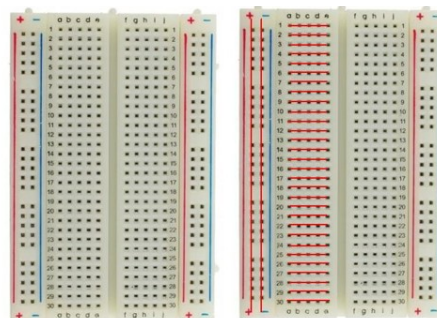


Рисунок 10 – Макетна плата

## 1.4 Інструментальні засоби розробки

<sup>1)</sup> [10] Інформаційний портал з плат Ардуіно. URL: <https://arduino.ua/prod1804-plata-adaptera-node-mcu-esp8266-v3>. (дата звернення 24.03.2019)

Середовище розробки Ардуіно (рис. 11) складається з вбудованого текстового редактора програмного коду, області повідомлень, вікна виведення тексту (консолі), панелі інструментів з кнопками часто використовуваних команд і декількох меню. Для завантаження програм і зв'язку середовище розробки підключається до апаратної частини Ардуіно.

Програма написана в середовищі Arduino IDE, називається “скетч”. Скетч пишеться в текстовому редакторі, що має інструменти вирізки / вставки, пошуку / заміни тексту. Під час збереження і експорту проекту в області повідомлень з'являються пояснення, також можуть відображатися помилки.

Вікно виведення тексту(консоль) показує повідомлення Ардуіно, що включають повні звіти про помилки та іншу інформацію. Кнопки панелі інструментів дозволяють перевірити і записати програму, створити, відкрити і зберегти скетч, відкрити моніторинг послідовної шини [11]<sup>1)</sup>.

---

<sup>1)</sup> [11] Інформаційний блог. URL: <https://soltau.ru/index.../465-kakie-sushchestvuyut-sredy-razrabotki-ide-dlya-arduino/>. (дата звернення 28.03.2019)

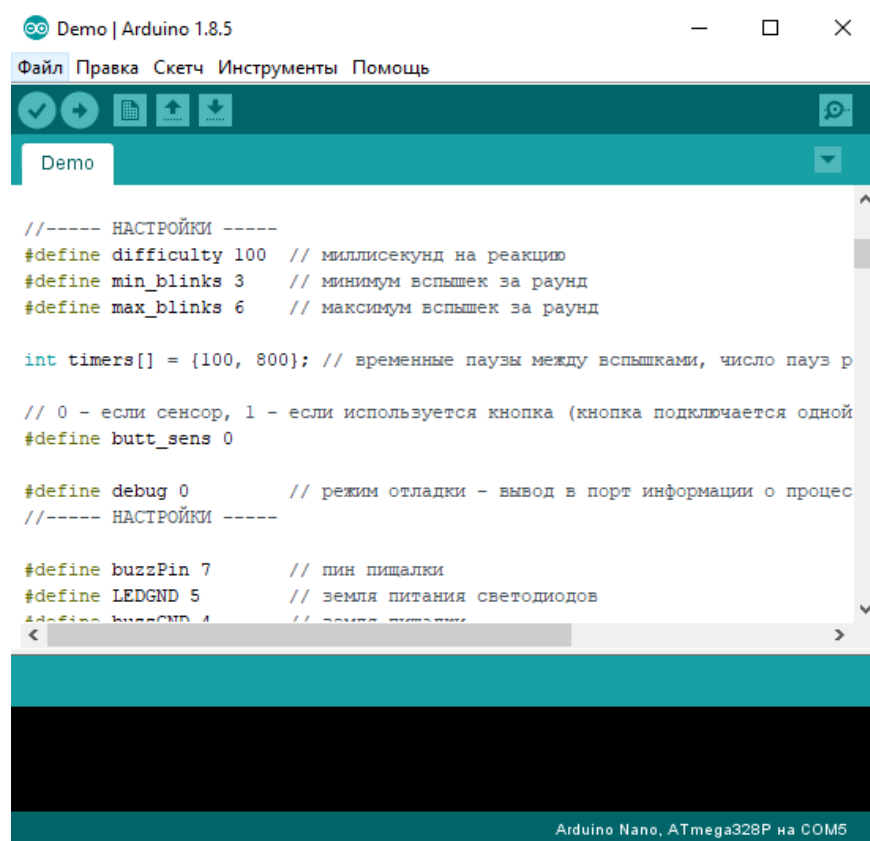






Рисунок 11 – Середовище розробки “Arduino IDE”

Середовище розробки містить такі основні елементи: текстовий редактор для написання коду, область для виведення повідомлень, текстова консоль, панель інструментів з традиційними кнопками та головне меню.

Дане програмне забезпечення дозволяє комп’ютеру взаємодіяти з Arduino як для передачі даних, так и для прошивки коду в контролер:

-  (Перевірка / Компіляція) – перевірка програмного коду на помилки, компіляція. Якщо компіляція пройшла успішно, Arduino IDE видасть нам таке повідомлення (рис. 12);
-  (Створити новий скетч) – створення нового скетчу;
-  (Відкрити) – відкриття меню доступу до всіх скетчів в блокноті. Відкривається натисканням в поточному вікні;

-  (Завантажити) – компілює програмний код і завантажує його в пристрій Ардуіно. Опис завантаження наведено нижче [11]<sup>1)</sup>.

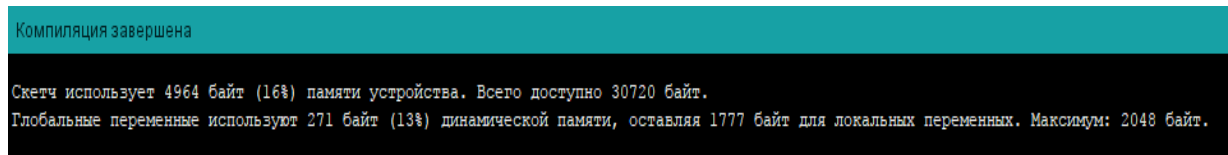


Рисунок 12 – Успішна компіляція

Додаткові команди згруповані в п'ять меню: File, Edit, Sketch, Tools, Help. Перед записом рекомендується перевірити правильність вибору платформи з меню. При використанні AVR ISP необхідно вибрати програматору порт з меню «Serial Port». Не мало важною частиною програмування Ардуіно є бібліотеки. Бібліотеки додають додаткову функціональність скетчам, наприклад, при роботі з апаратною частиною або при обробці даних. Для використання бібліотеки необхідно вибрати меню Sketch> Import Library.

Інші можуть бути завантажені з різних ресурсів. Для установки викачаних бібліотек необхідно створити директорію «libraries» в папці блокнота і потім розпакувати архів у необхідній директорії. Наприклад, для установки бібліотеки «DateTime» її файли повинні знаходитися в папці «libraries / DateTime» папки блокнота [11]<sup>1)</sup>.

Fritzing є чудовим інструментом розробника з відкритим початковим кодом для навчання, прототипування і обміном проектами на базі Arduino. Fritzing дозволяє розробити принципову схему пристрою (рис. 13), і

<sup>1)</sup> [11] Інформаційний блог. URL: <https://soltau.ru/index.../465-kakie-sushchestvuyut-sredy-razrabotki-ide-dlya-arduino/>. (дата звернення 28.03.2019)

<sup>1)</sup> [11] Інформаційний блог. URL: <https://soltau.ru/index.../465-kakie-sushchestvuyut-sredy-razrabotki-ide-dlya-arduino/>. (дата звернення 28.03.2019)

створити її представлення у вигляді з'єднання макетів елементів, які виглядають дуже навіть професійно.

Він також дає можливість розробити друковану плату для її подальшого виготовлення. На відміну від інших систем проектування, у Fritzing простий інтерфейс, який робить розробку електронних схем інтуїтивно зрозумілою [12]<sup>2)</sup>.

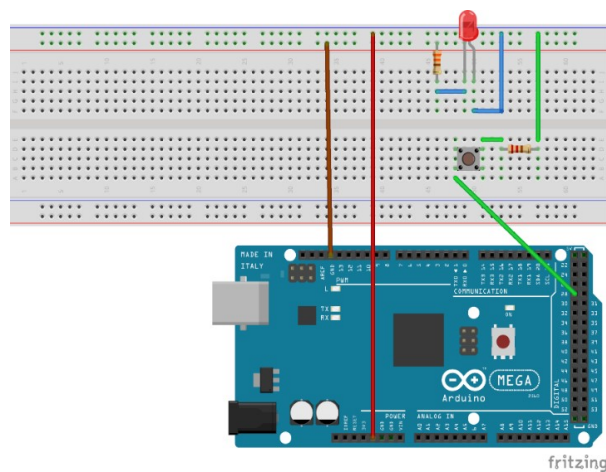


Рисунок 13 – Приклад схема з'єднань у Fritzing  
На вкладці «Макетна плата» можна побачити наступний екран (рис. 14):

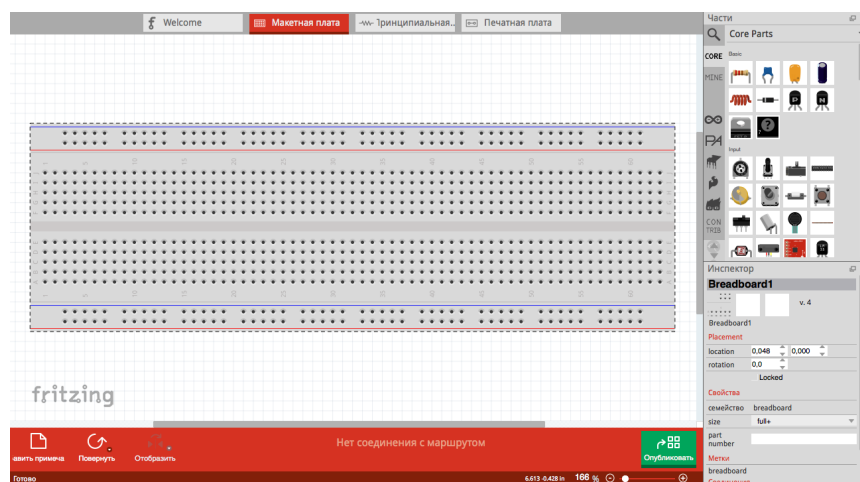


Рисунок 14 – Робоче вікно вкладки «Макетна плата»

У правій частині екрану знаходиться панель інструментів з усіма елементами і опціями (рис. 15). Якщо компонент настроюється, то в нижній

<sup>2)</sup> [12] Офіційний сайт Fritzing. URL: [fritzing.org/](http://fritzing.org/). (дата звернення 02.04.2019)

частині панелі інструментів відображаються параметри, що настраюються, для цього компонента [13]<sup>1)</sup>:

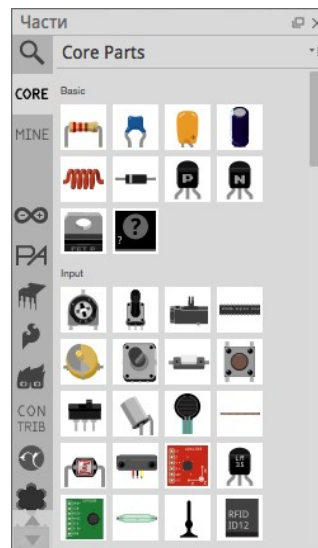


Рисунок 15 – Меню компонентів

## 2 ПРОЕКТНА ЧАСТИНА

### 2.1 Моделювання системи

UML (англ. Unified Modeling Language) – уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Вона є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, яка називається UML-моделлю [14]<sup>1)</sup>.

UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

<sup>1)</sup> [13] Інформаційний портал з Ардуіно. URL:<https://arduinomaster.ru/biblioteki-arduino/skachat-biblioteki-arduino/>. (дата звернення 05.04.2019)

<sup>1)</sup> [14] Інформаційний портал з розробки програмного забезпечення. URL:<https://pro-prof.com/archives/2594>. (дата звернення 23.04.2019)



UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки прикладних програм. Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем.

Діаграма прецедентів – в UML, діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання. Ця діаграма є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямо-кутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання [15]<sup>2)</sup>.

Суть даної діаграми полягає в наступному: проектована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання.

Варіант використання (англ. use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором. При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

Так, для об'єкту розробки діаграма прецедентів буде мати наступний вид (рис. 16):

---

<sup>2)</sup> [15] Інформаційний портал з UML-моделювання систем. URL: <https://www.lucidchart.com/pages/uml-use-case-diagram>. (дата тзвернення 23.04.2019)

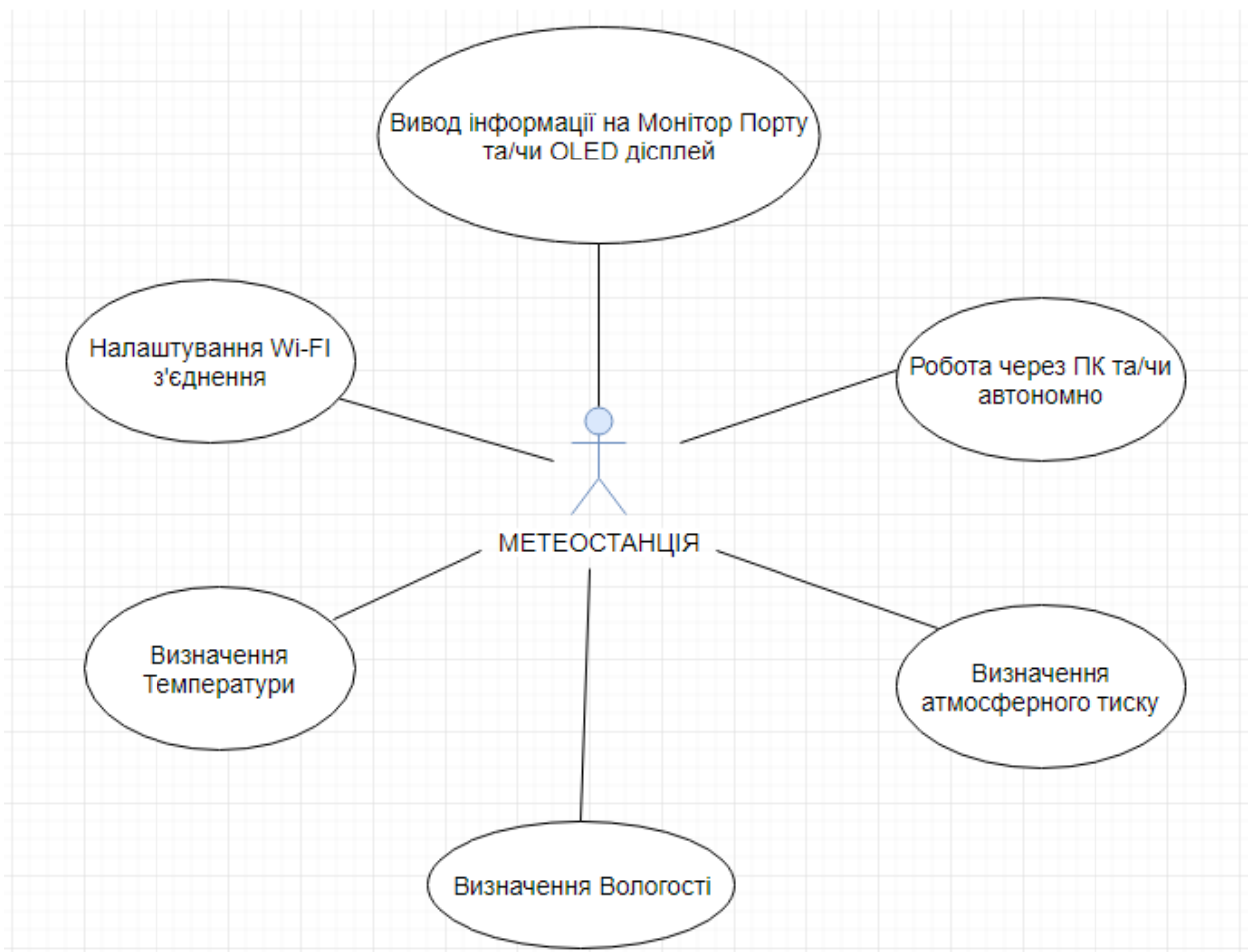


Рисунок 16 – Діаграма USE-CASE

Діаграма діяльності – в UML, візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій.

Дія є фундаментальною одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів. Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в

діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності.

Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності. Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку [16]<sup>1)</sup>.

Ці діаграми широко використовуються в описі поведінки, що включає велику кількість паралельних процесів. Кожний стан на діаграмі діяльності відповідає виконанню деякої елементарної операції, а перехід в наступний стан виконується тільки після завершення цієї операції.

Таким чином, діаграму діяльності можна вважати приватним випадком діаграми станів.

Основним напрямком використання діаграми діяльності є візуалізація особливостей реалізації операцій класів, коли необхідно надати алгоритми їх виконання [17]<sup>2)</sup>.

Наступна діаграма демонструє основні етапи розробки метеостанції (рис. 17):

---

<sup>1)</sup> [16] Інформаційний ресурс з моделювання на UML. URL: <http://book.uml3.ru/>. (дата звернення 26.04.2019)

<sup>2)</sup> [17] Офіційний сайт UML. URL: <https://www.uml.org/>. (дата звернення 26.04.2019)

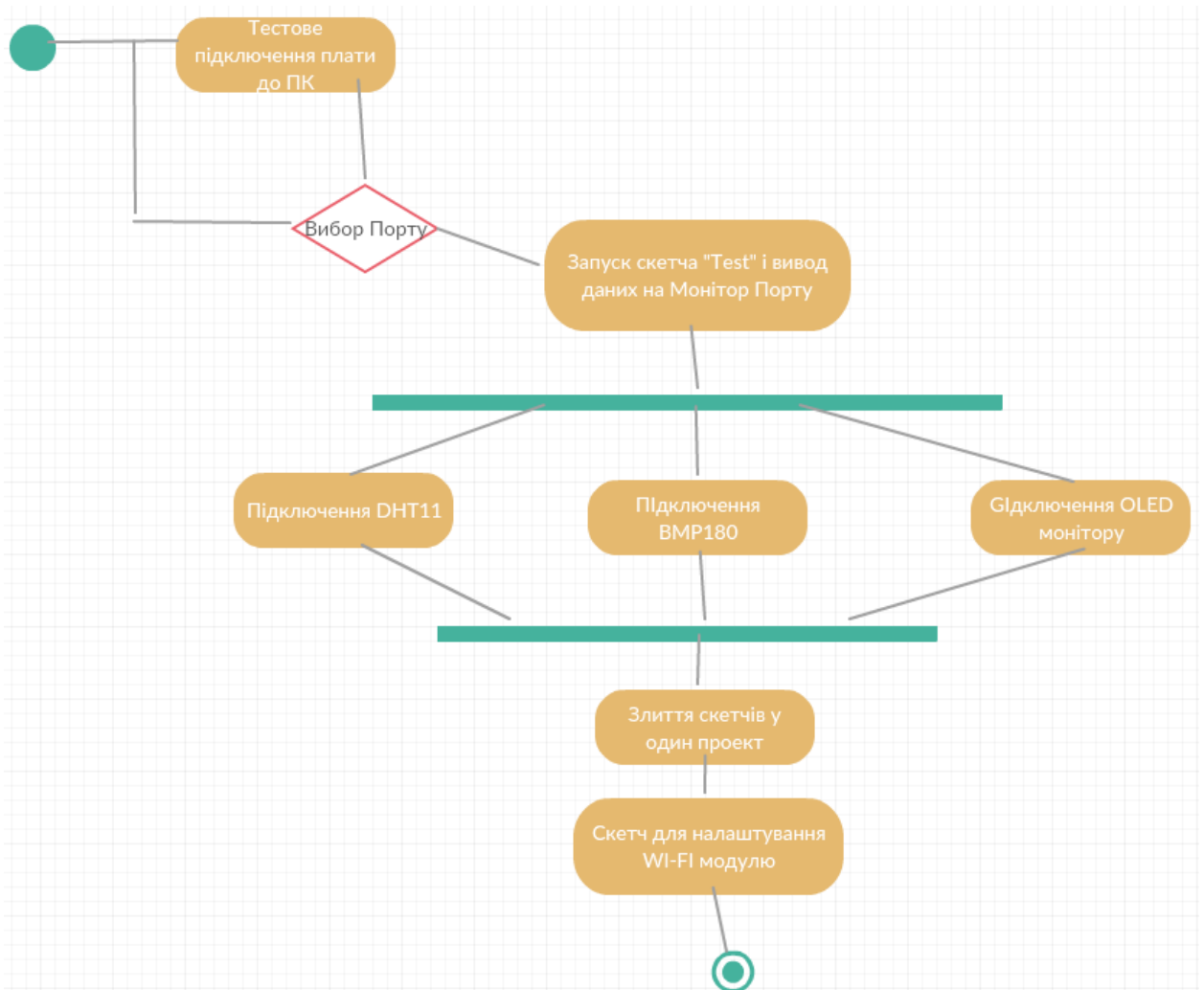


Рисунок 17 – Діаграма діяльності «Розробка метеостанції»

## 2.2 Налаштування середовища розробки Arduino IDE

### 2.2.1 Установка середовища розробки

Скетч для мікроконтролера Arduino являє собою файл з розширенням «.іпо». Після завантаження на мікроконтролер програма зберігається навіть після відключення Arduino, до тих пір, поки вона не буде переписана новою програмою.

Потрібно на ресурсі «<https://www.arduino.cc/en/Main/Software>» встановити потрібну версію середовища розробки під свою операційну систему. Далі, підключаємо плату до ПК USB кабелем.

У середовищі розробки вказуємо потрібну плату (на платі вказано її найменування) (рис. 18):

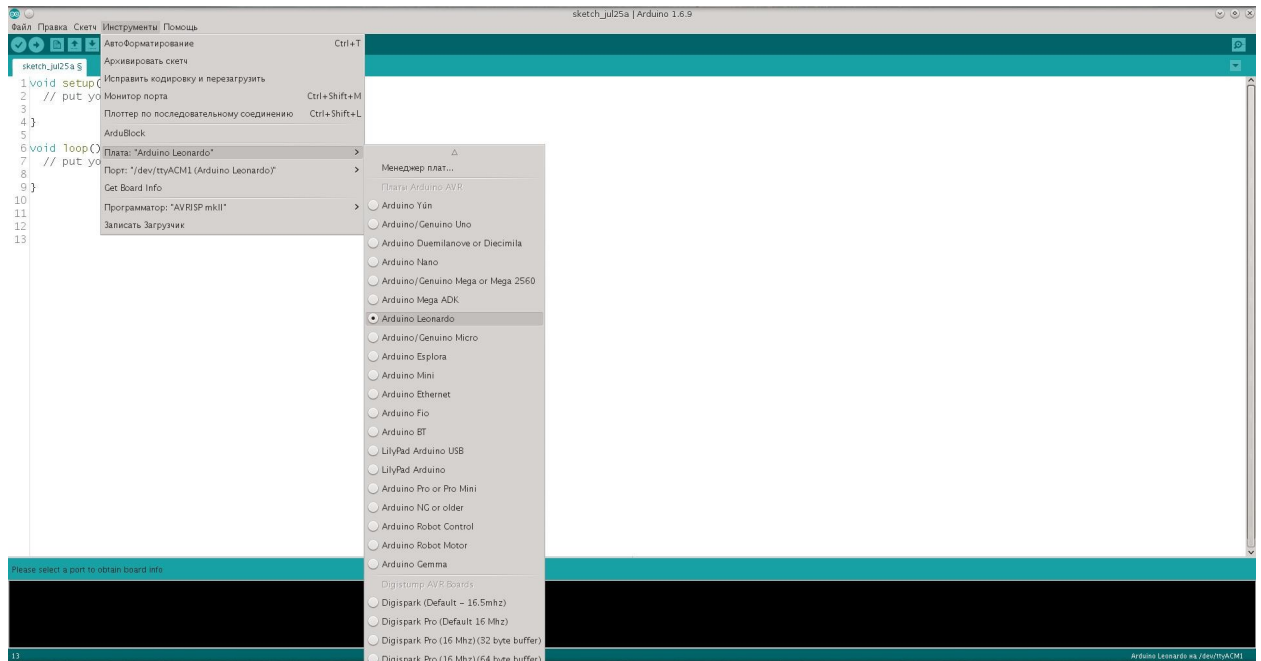


Рисунок 18 – Підключення плати до IDE Arduino

В ОС Windows порти можуть позначатися як COM1 або COM2 (для плати послідовної шини) або COM4, COM5, COM7 і вище (для плати USB). Визначення порту USB проводиться в поле Послідовною шини USB Диспетчера пристроїв Windows.

Після вибору порту і платформи необхідно натиснути кнопку завантаження на панелі інструментів або вибрати пункт меню «File» Upload to I/O Board». Сучасні платформи Arduino перезавантажуються автоматично перед завантаженням.

На наступному кроці вказуємо порт мікроконтролера і таким чином перевіряємо коректність підключення плати (рис. 19).

Викликавши пункт меню «Інструменти → Get board info», можна перевірити, що на обраному порту дійсно знаходиться потрібна плата.

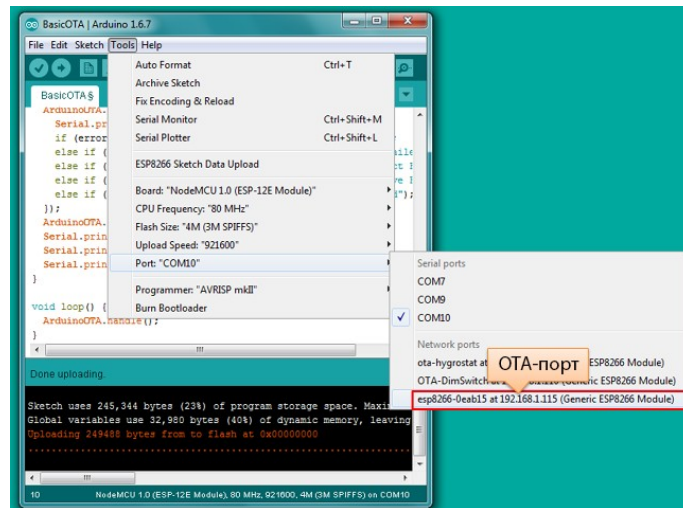


Рисунок 19 – Підключення порта плати

## 2.2.2 Управління бібліотеками

Бібліотеки додають додаткову функціональність скетчам, наприклад, при роботі з апаратною частиною або при обробці даних. Для використання бібліотеки необхідно вибрати меню «Sketch» Import Library». Одна або кілька директив «#include» будуть розміщені на початку коду скетчу з подальшою компіляцією бібліотек і разом зі скетчем. Завантаження бібліотек вимагає додаткового місця в пам'яті Arduino. Невикористані бібліотеки можна видалити з скетчу прибравши директиву «#include».

На Arduino.cc є список бібліотек. Деякі бібліотеки включені в середу розробки Arduino. Інші можуть бути завантажені з різних ресурсів. Для установки викачаних бібліотек необхідно створити директорію «libraries» в папці блокнота і потім розпакувати архів [18]<sup>1)</sup>.

Наприклад, для установки бібліотеки DateTime її файли повинні знаходитися в папці / libraries / DateTime папки блокнота.

Іноді в процесі роботи виникає необхідність підключення до скетчу сторонніх бібліотек, наприклад, для роботи з різними датчиками. Бібліотека

<sup>1)</sup> [18] Інформаційний ресурс. URL:[http://arduino.ru/Arduino\\_environment](http://arduino.ru/Arduino_environment). (дата звернення 02.05.2019)

являє собою набір заголовних і файлів (з розширенням .h) і файлів з вихідним кодом (розширення .c або .cpp) [6]<sup>1)</sup>.

При підключенні бібліотеки в скетч автоматично додаються підключення потрібних заголовків файлів (такі рядки `#include <...>`) [19]<sup>2)</sup>.

Для прикладу, можна підключити одну зі стандартних бібліотек Wire. Для цього слід обрати у меню «Скетч → Підключити бібліотеку → Wire».

При цьому IDE автоматично додасть зміни в скетч, необхідні для підключення бібліотеки.

У верху файлу скетчу повинна з'явитися рядок:

```
#include <Wire.h>
```

Відразу після установки в середовищі розробки вже доступний базовий набір бібліотек (в тому числі Wire з прикладу вище). Решта бібліотеки спочатку потрібно завантажити в IDE, для цього використовується менеджер бібліотек [3]<sup>3)</sup>.

Вибираємо пункт меню «Скетч → Підключити бібліотеку → Управління бібліотеками».

У менеджері (рис. 20) присутня безліч бібліотек для роботи з різними модулями / протоколами і т.п. Після завантаження бібліотеки вона стає доступна для підключення до скетчу.

У верхній частині менеджера знаходяться два поля фільтрації за типом і темі, а так само поле текстового пошуку по назві і опису бібліотеки. При виборі бібліотеки в списку стає доступна кнопка «Установка», для деяких бібліотек так само можна вибрати версію, якщо є кілька версій.

Бібліотеки для роботи з датчиками температури:

---

<sup>1)</sup> [6] Інформаційний довідник з Ардуіно. URL: <http://radioschema.ru/el-komponenty/spravochnik>. (дата звернення 20.03.2019)

<sup>2)</sup> [19] Інформаційний портал з Ардуіно. URL: <https://arduinomaster.ru/biblioteki-arduino/skachat-biblioteki-arduino/>. (дата звернення 05.04.2019)

<sup>3)</sup> [3] Інформаційний довідник з Ардуіно. URL: <http://arduino-diy.com/>. (дата звернення 13.03.2019)

- Adafruit SHT31;
- Adafruit Si7021;
- Arduino Learning Board.

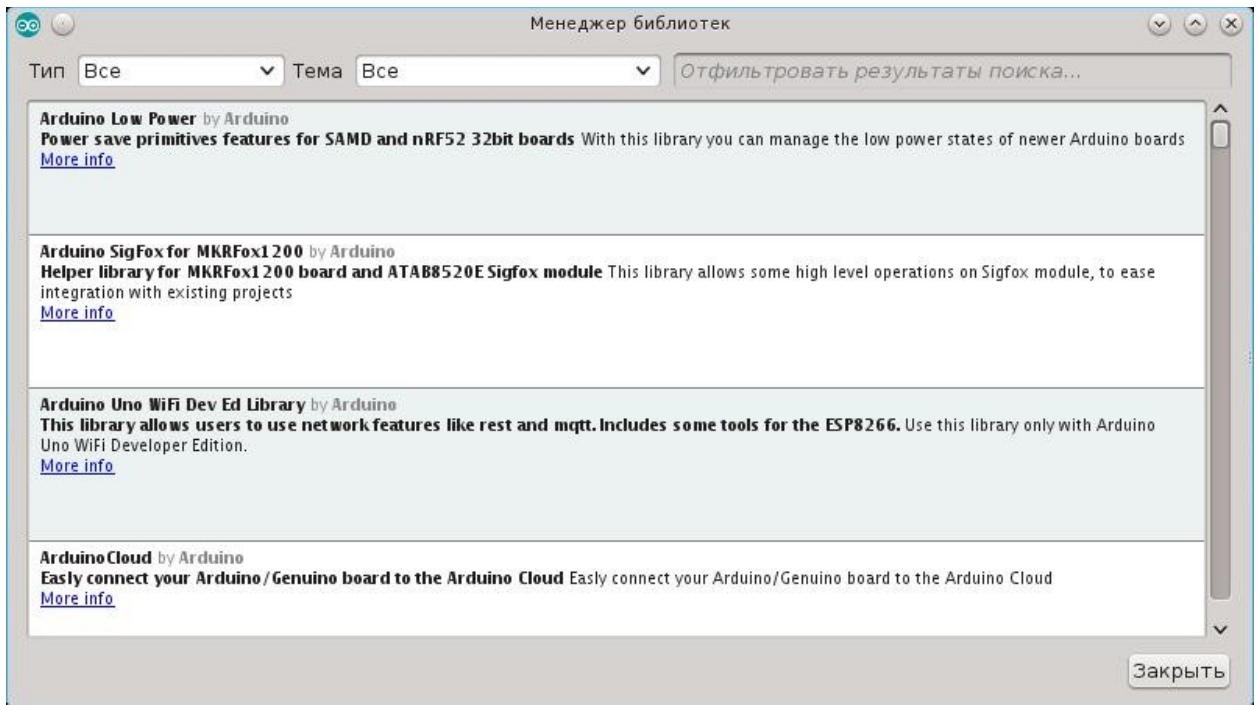



Рисунок 20 – Вікно менеджера бібліотек IDE Arduino

### 2.2.3 Монітор послідовного порту

У середовищі Arduino IDE зв'язок комп'ютера з контролером обгорнута в графічну оболонку - Монітор порту. Для відкриття якого клацаємо «Сервіс => Монітор порту», або ж комбінація клавіш «Ctrl + Shift + M».

Відкриваємо монітор послідовного порту (Інструменти -> Монітор порту або кнопка  вгорі праворуч). З'являється наступне вікно (рис. 21).



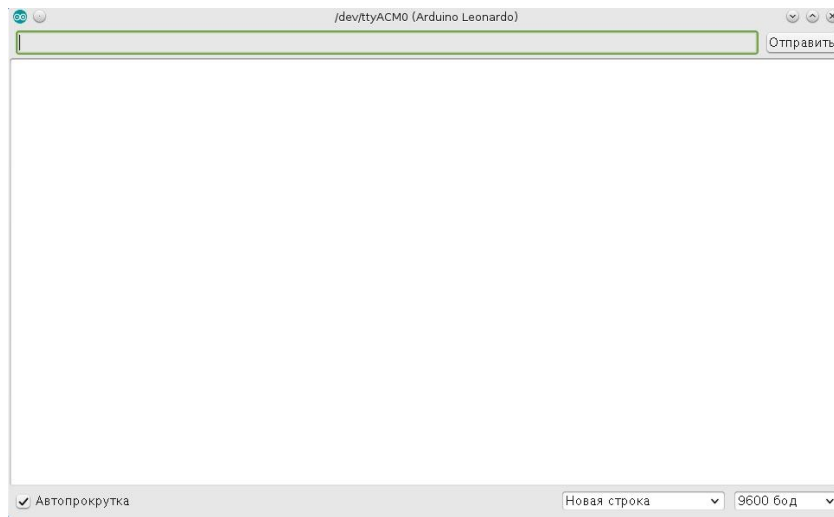


Рисунок 21 – Вікно послідовного порту IDE Arduino

В даному вікні виводиться інформація, яка надходить від контролера. У верхньому рядку зліва від кнопки "Відправити" можна писати текст, який відправляється на контролер.

У нижньому правому вікні налаштовується швидкість передачі даних і вказується, чи буде до тексту додаватися символ перекладу рядка NL, CR або обидва відразу, або текст буде відправлений як є. Для даної роботи потрібно виставити режим «Немає кінця рядка» [13]<sup>1)</sup>. Для коректної роботи з портом слід дотримуватися двох умов:

- правильно обраний COM порт яким визначилася плата Arduino;
- швидкість роботи в скетчі повинна збігатися зі швидкістю обраної в Моніторі порту.

Перше необхідно встановити вручну «Сервіс => Послідовний порт => COM», друге ж задається в скетчі функцією:

```
Serial.begin(); // Задаємо швидкість роботи монітор порту
```

---

<sup>1)</sup> [13] Інформаційний портал. URL: <https://volti.ru/wiki/arduino-ide-port-monitor/>. (дата звернення 10.04.2019)

При відкритті «Монітор порту» в ньому по дефолту встановлена швидкість 9600, в зв'язку з цим, в скетчах Arduino можна часто зустріти цю швидкість. Сама ж передача даних в порт задається функцією:

```
Serial.println(); // Вивод повідомлення на монітор порту
```

Результат наведено на рисунку 22:

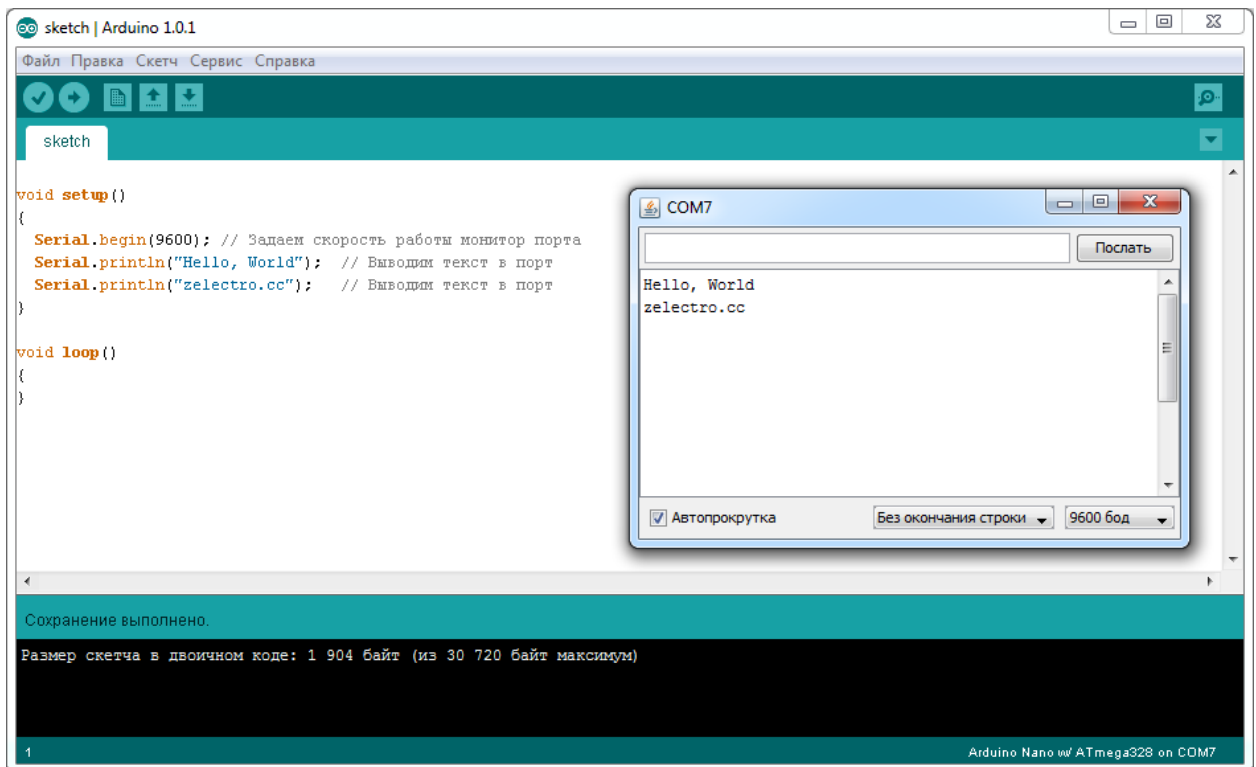


Рисунок 22 – Тестування роботи послідовного порту IDE Arduino

## 2.2.4 Проектування схем пристрою

Схема для метеостанції буде будуватись поетапно, послідовно підключаючи до неї необхідні елементи (шилди) і створюючи окремі програмні скетчі для її роботи. Так буде зручніше протестувати роботи всіх елементів ще на стадії проектування.

За результатами успішного тестування всі скетчі будуть з'єднані у єдиному проекті.

На першому етапі до обраної плати (NodeMCU) слід спроектувати схему підключення датчику BMP180 для визначення атмосферного тиску і температури. Так як цей елемент не входить до стандартної бібліотеки Fritzing, його слід завантажити додатково з інтернет-ресурсу [20]<sup>1)</sup>.

Приклад розміщення схеми на макетній платі представлено на рисунку 23, він не є обов'язковим, так як при нарощенні елементів до схеми розміщення може бути змінено.

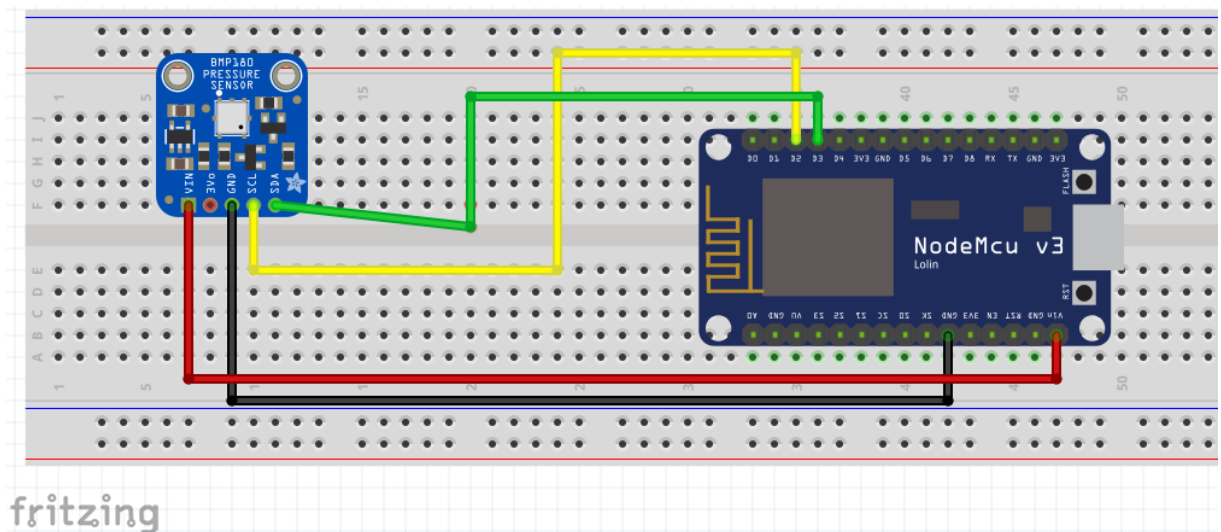


Рисунок 23 – Схема підключення датчику BMP180 до плати NodeMCU

Далі у програмі Fritzing побудуємо схему для підключення датчику DHT11 (його так само слід завантажити додатково) для визначення рівня вологості (рис. 24):

<sup>1)</sup> [20] Бібліотека елементів до програми Fritzing. URL: <https://github.com/adafruit/Fritzing-Library/blob/master/parts/Adafruit%20BMP180.fzpz>. (дата звернення 08.05.2019)

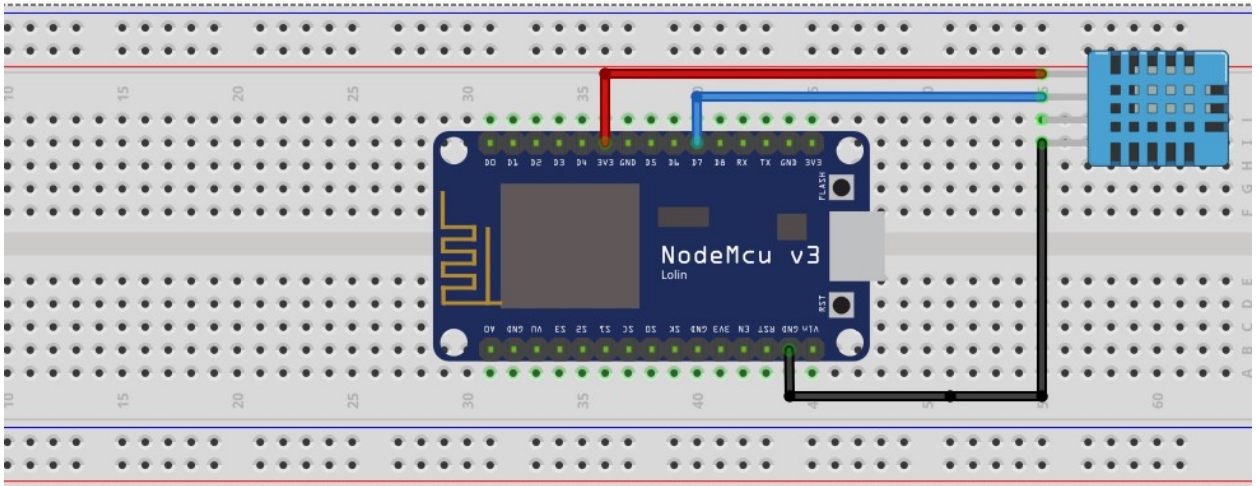


Рисунок 24 – Схема підключення датчика DHT11 до плати NodeMCU

Крім вищезазначених датчиків до схеми за метою роботи слід підключити екран для зручного виводу інформації. Після об'єднання всіх елементів у одній схемі, отримаємо результат, представлений на рисунку 25.

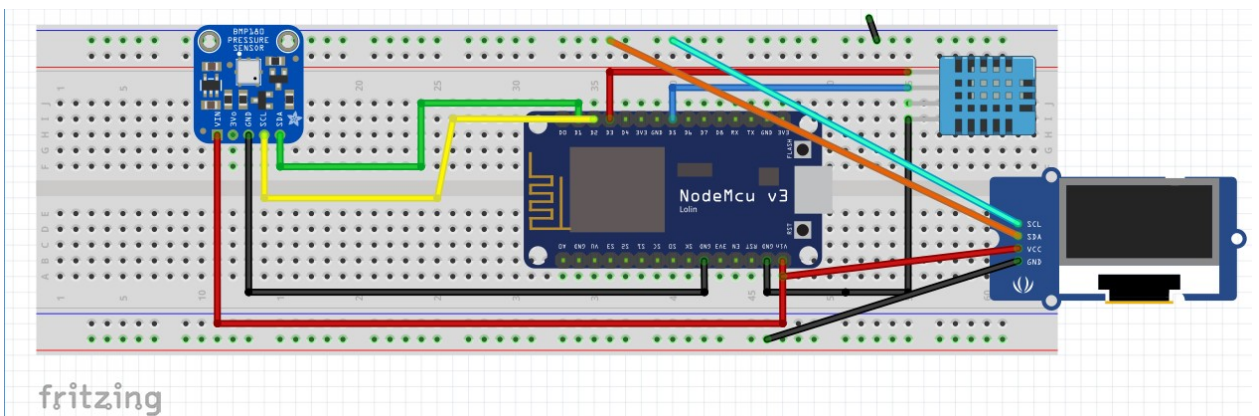


Рисунок 25 – Схема для метеостанції

## 3 ПРОГРАМНА ЧАСТИНА

### 3.1 Тестове підключення плати

Спочатку потрібно підключити плату до ПК USB кабелем. У середовищі розробки для цього вказуємо назву необхідної плати (рис. 26), після чого обираємо необхідний порт.

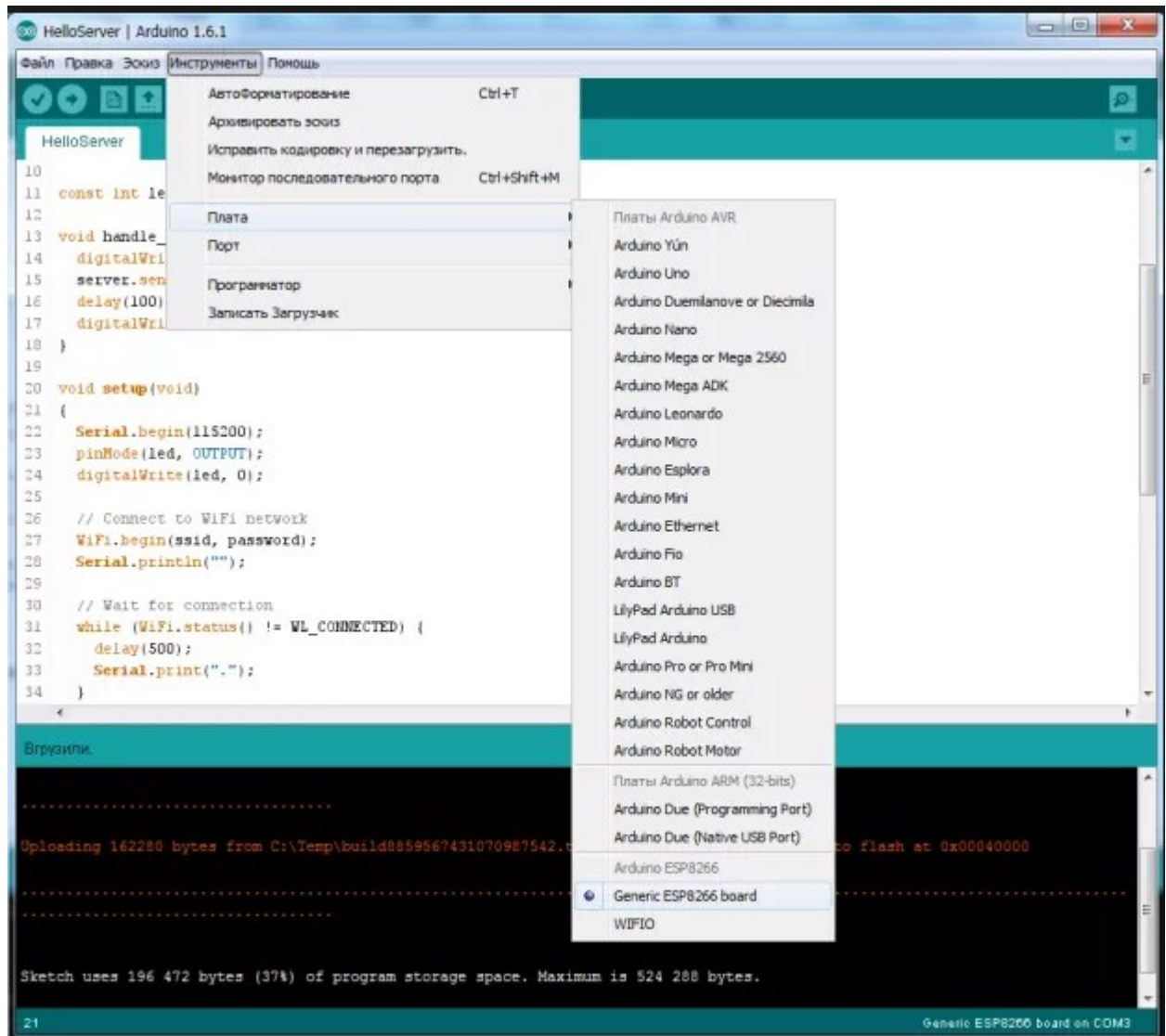



Рисунок 26 – Вибір необхідної плати

Викликавши пункт меню «Інструменти →Get board info», можна перевірити, що на обраному порту дійсно знаходиться потрібна плата.

Для пробного тесту роботи плати використовується приклад роботи світлодіода.

Вибираємо з бібліотеки прикладів скетч Blink (Файл -> Приклади -> 0.1Basics -> Blink). Перевіряємо обраний порт і плату. Натискаємо кнопку Завантажити () або вибираємо «Скетч -> Завантаження».

Після завантаження скетчу на платі повинен почати мигати світлодіод раз в секунду.

Розглянемо основні блоки скетчу для тесту: функція setup() запускається один раз, при підключенні плати до комп'ютера:

```
void setup() {
```

ініціалізація пінов LED\_BUILTIN як вихідних

```
    pinMode(LED_BUILTIN_TX, OUTPUT);
    pinMode(LED_BUILTIN_RX, OUTPUT);
}
```

Функція loop() має наступний вигляд:

```
void loop() {
```

вмикаємо світлодіод (високий рівень напруги) і очікуємо одну секунду:

```
    digitalWrite(LED_BUILTIN_TX, HIGH);
    delay(1000);
```

вимикаємо світлодіод, роблячи напругу LOW і знов очікуємо секунду:

```
    digitalWrite(LED_BUILTIN_TX, LOW);
    delay(1000);
```

Тепер вмикаємо правий світоід з урахуванням паузи, як і в випадку лівого світодіода:

```

        digitalWrite(LED_BUILTIN_RX, HIGH);

    delay(1000);
    digitalWrite(LED_BUILTIN_RX, LOW)
    delay(1000); // wait for a second
}

```

### 3.2 Підключення датчика температури і вологості DHT11

Спочатку необхідно установити бібліотеку для роботи з датчиком DHT11. Для цього створюємо новий скетч і зберігаємо його з назвою «DHTTest».

Перевіряємо, чи правильно обрана плата і порт. Відкриваємо менеджер бібліотек і знаходимо в списку бібліотек дві:

- «Adafruit Unified Sensor»;
- «DHT sensor library».

Встановлюємо обидві. Підключаємо до проекту бібліотеку «DHT sensor library». На початку скетчу повинні з'явитися рядки:

```

#include <DHT.h>

#include <DHT_U.h>

```

Наступним кроком роботи є зборка установки і розробка скетчу для Arduino для зчитування показань датчика DHT11.

Підключаємо датчик DHT до мікроконтролера «-» до gnd, «+» до 3.5V, середній роз'єм «out» до цифрового піну 13.

Допрацьовуємо скетч. До функції setup () створюємо глобальну змінну dht, вказуючи номер цифрового піна 13 і тип датчика DHT11:

```
DHT_Unified dht(13,DHT11);
```

У функції `setup()` ініціалізуємо послідовний порт і датчик:

```
Serial.begin(9600);
  dht.begin();
```

Далі модифікуємо функцію `loop ()`, в ній виробляємо наступні дії: спочатку створюємо змінну під значення датчика і зчитуємо в неї значення температури. Якщо значення вірне, виводимо в `Serial` температуру, в іншому випадку виводимо повідомлення про помилку;

```
sensors_event_t event;

dht.temperature().getEvent(&event);

if(isnan(event.temperature))
  Serial.println("Error reading temperature!");
else
{
  Serial.print("Temperature: ");

  Serial.print(event.temperature);

  Serial.println(" *C");
}
```

Далі, повторюємо аналогічно для вологості.

```
dht.humidity().getEvent(&event);

if(isnan(event.relative_humidity))
  Serial.println("Error reading humidity!");
else
{
  Serial.print("Humidity: ");

  Serial.print(event.relative_humidity);

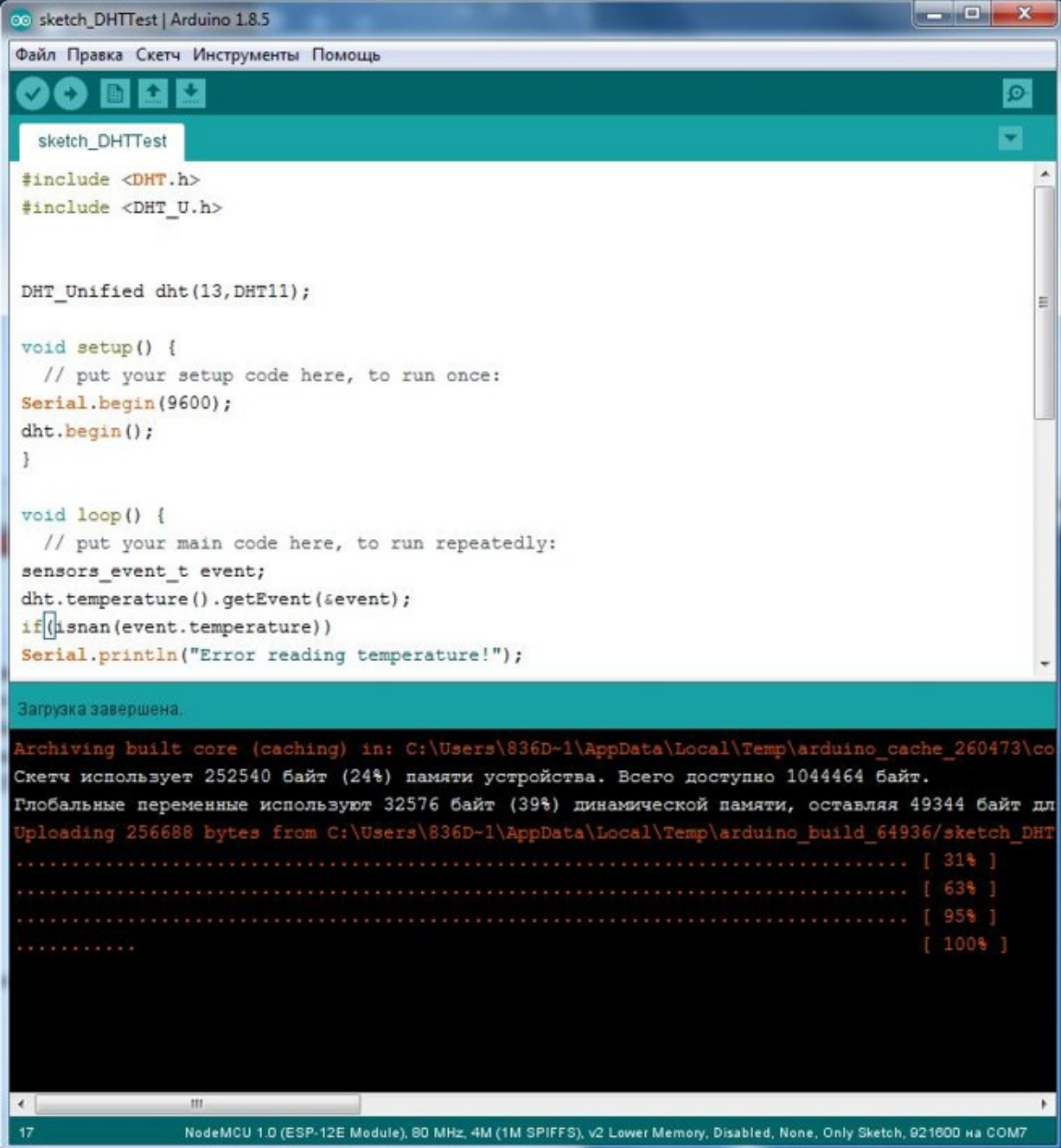
  Serial.println("%");
}
```



Далі слід зробити зупинку на пів секунди, а саме:

```
delay(500);
```

Після цього запускаємо скетч на компіляцію для перевірки наявності помилок (рис. 27).



```
sketch_DHTTest | Arduino 1.8.5
Файл Правка Скетч Инструменты Помощь
sketch_DHTTest
#include <DHT.h>
#include <DHT_U.h>

DHT_Unified dht(13,DHT11);

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  // put your main code here, to run repeatedly:
  sensors_event_t event;
  dht.temperature().getEvent(&event);
  if(!isnan(event.temperature))
  Serial.println("Error reading temperature!");
}

Загрузка завершена.
Archiving built core (caching) in: C:\Users\836D-1\AppData\Local\Temp\arduino_cache_260473\co
Скетч использует 252540 байт (24%) памяти устройства. Всего доступно 1044464 байт.
Глобальные переменные используют 32576 байт (39%) динамической памяти, оставляя 49344 байт дл
Uploading 256688 bytes from C:\Users\836D-1\AppData\Local\Temp\arduino_build_64936/sketch_DHT
..... [ 31% ]
..... [ 63% ]
..... [ 95% ]
..... [ 100% ]

17 NodeMCU 1.0 (ESP-12E Module), 80 MHz, 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 921600 на COM7
```

Рисунок 27 – Процес компіляції скетчу IDE Arduino

У разі вдалого компілювання слід відкрити монітор порту для перевірки наявності даних з датчику (рис. 28). Результат підключення модуля DHT11 представлено на рисунку 29.

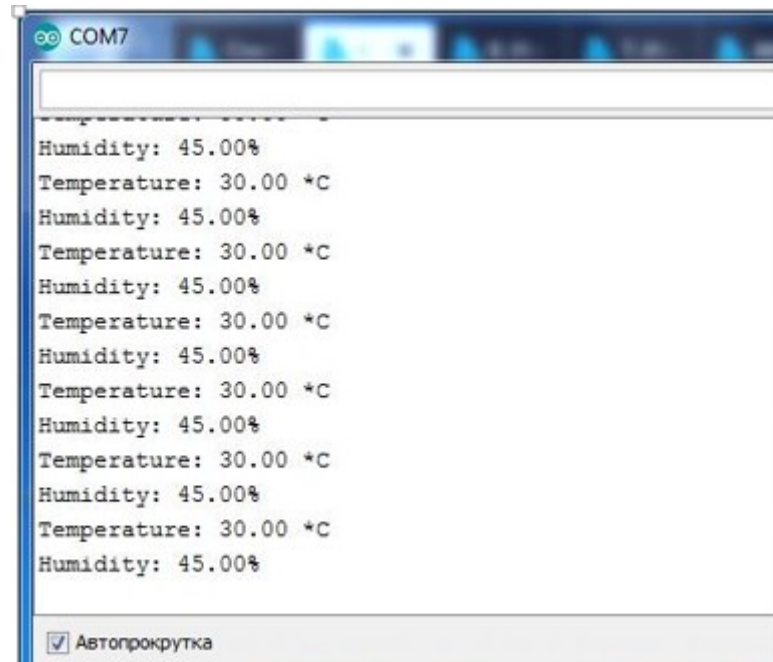


Рисунок 28 – Результати роботи на моніторі порту IDE Arduino

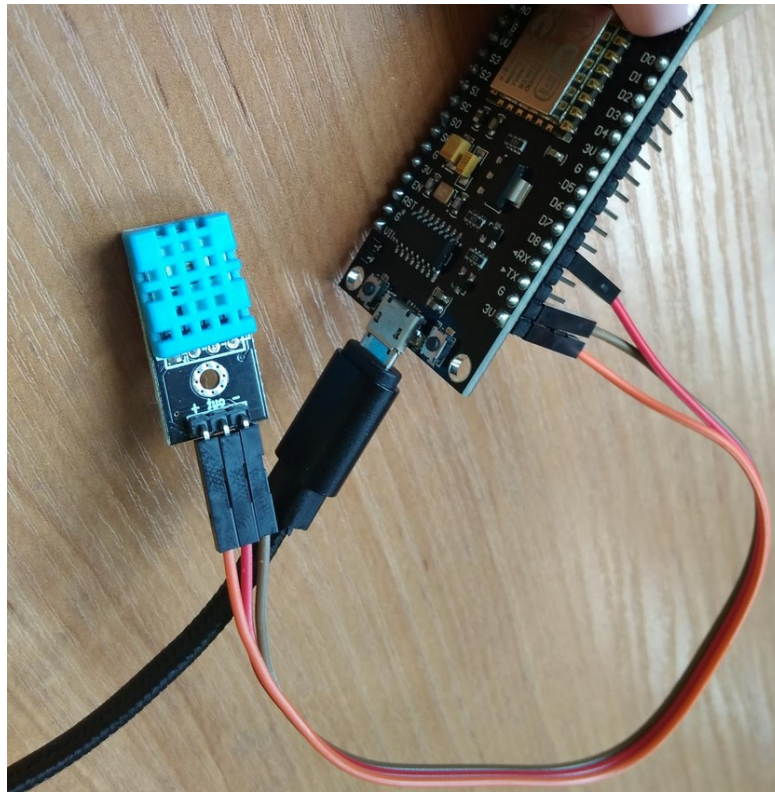


Рисунок 29 – Результат підключення модуля DHT11

### 3.3 Підключення датчика тиску BMP180

BMP180 – суміщений датчик для вимірювання атмосферного тиску і температури. Залежно від виробника існує кілька готових модулів датчиків BMP180, що розрізняються числом висновків і напругою живлення. У даній роботі використовується модуль «gy-68» з 4 висновками і напругою живлення 3.3 вольта.

На першому етапі слід установити бібліотеку для роботи з датчиком BMP180. Створюємо новий скетч і зберігаємо його з назвою «BMP180Test». Перевіряємо, чи правильно обрана плата і порт.

Завантажуємо бібліотеку для роботи з датчиком BMP180 «Adafruit BMP085 Unified» через менеджер бібліотек.

```
Adafruit_BMP085_Unified bmp(10085);
```

У функції `setup ()` запускаємо послідовний порт і ініціалізуємо датчик BMP180:

```
Serial.begin (9600);  
bmp.begin ();
```

Далі в функції `loop ()` зчитуємо значення тиску:

```
sensors_event_t event;  
bmp.getEvent (& event);
```

Виводимо тиск в hPa:

```
Serial.print ("Pressure (hPa):");  
Serial.print (event.pressure);
```

Прочитуємо значення температури:

```
float temperature;  
bmp.getTemperature (& temperature);
```

і виводимо його значення на екран монітора:

```
Serial.print ("Temp (* C):");  
Serial.println (temperature);
```

Обов'язково необхідно додати затримку

```
delay (1000);
```

Результат роботи скетчу представлено на рисунку 30:

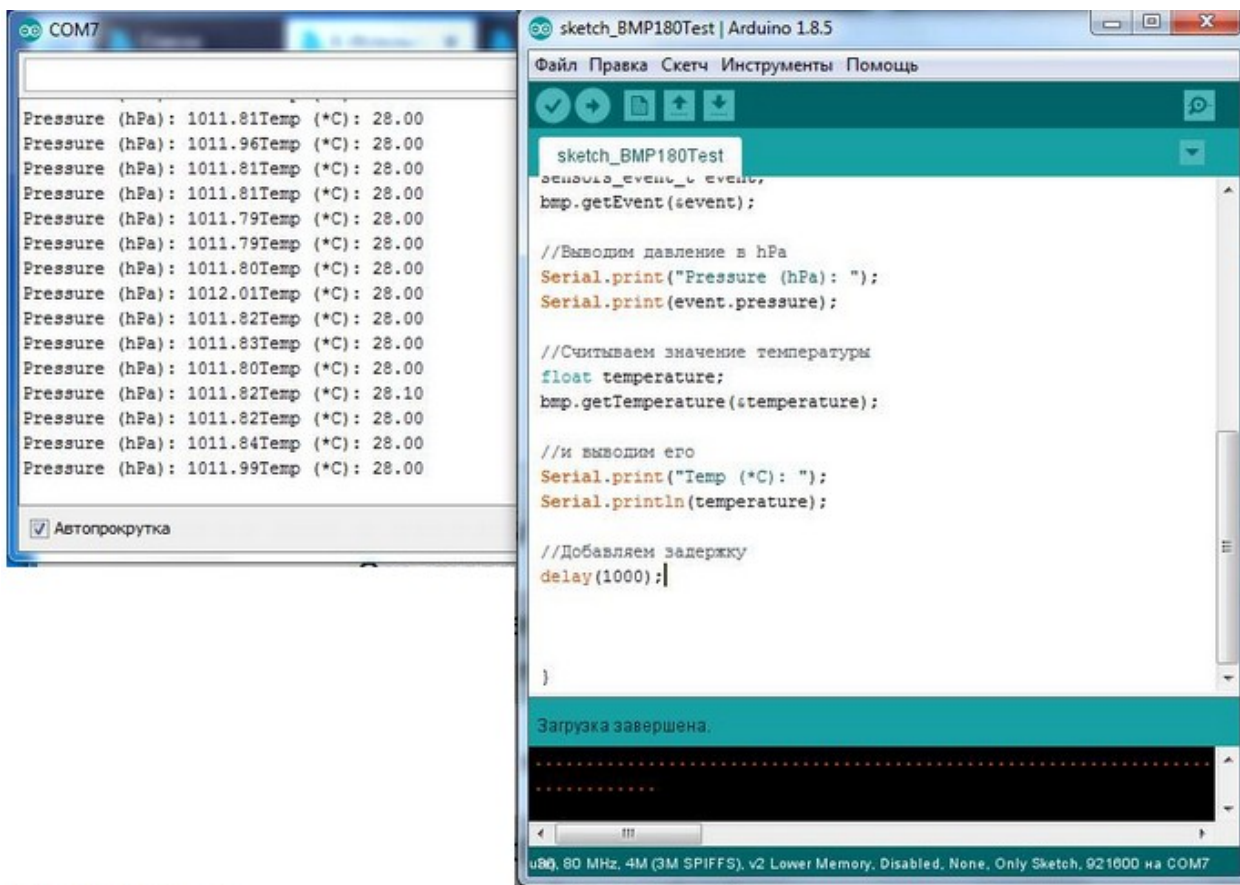


Рисунок 30 – Результат работы датчику BMP180

Результат підключення BMP180 до плати Arduino представлено на рисунку 31:

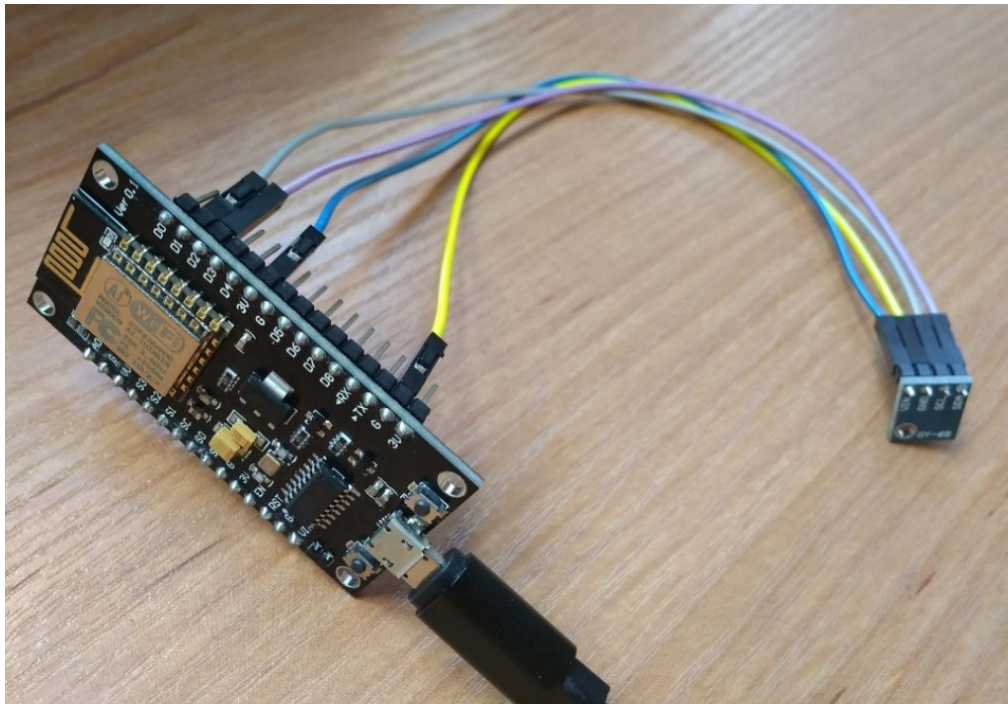


Рисунок 31 – Підключення BMP180 до плати Arduino

### 3.4 Робота з рідкокристалічним дисплеєм OLED 128x64 I2C

Існує декілька бібліотек для OLED I2C дисплея від різних авторів, ось кілька:

- від Adafruit (потрібна установка обох бібліотек) – Adafruit GFX і Adafruit SSD1306;
- від Oscar Liang – OzOLED;
- «Seeed OLED Display 128\*64 library» від Seeed-Studio.

Для даного проекту буде використовувана остання.

Створюємо новий скетч і зберігаємо його з назвою «OledAndSensors».

Перевіряємо, чи правильно обрана плата і порт.

Далі завантажуюмо архів з бібліотекою для роботи з дисплеєм за посиланням [15]<sup>1)</sup>.

---

<sup>1)</sup> [15] Інформаційний блог з програмування Github. URL: «[https://github.com/Seeed-Studio/OLED\\_Display\\_128X64/archive/master.zip](https://github.com/Seeed-Studio/OLED_Display_128X64/archive/master.zip)». (дата звернення 10.03.2019)

Встановлюємо бібліотеку через менеджер бібліотек: вибираємо пункт меню «Скетч» -> Управління бібліотеками -> Додати .ZIP бібліотеку» і знаходимо скачаний архів OLED\_Display\_128X64-master.zip.

Підключаємо бібліотеку для роботи з дисплеєм «Grove – OLED Display 0.96».

У функції setup () ініціалізуємо дисплей, очищаємо його і виводимо текст привітання.

```
// Init OLED Display
```

```
SeeedOled.sendCommand (0xA8); // Multiplex ratio
SeeedOled.sendCommand (0x3F); // ^ від 16 до 64
```

```
SeeedOled.sendCommand (0xD3); // Зсув зображення на дисплеї
                                (Offset)
```

```
SeeedOled.sendCommand (0x00); // ^ Offset
```

Далі слід установити початковий рядок в RAM:

```
SeeedOled.sendCommand (0x40);
SeeedOled.sendCommand (0xA1); // Розгортка: A0-справа/   наліво;
                                A1 -зліва/ направо.
```

«0xC8» – дані виводяться через підрядник зверху вниз, 0xC0 – від низу до верху:

```
SeeedOled.sendCommand (0xC8);
SeeedOled.sendCommand (0xDA); // Конфігурація COM
```

```
SeeedOled.sendCommand (0x12);
SeeedOled.sendCommand (0x81); // установить яскравість
```

```
SeeedOled.sendCommand (0x7F); // Яскравість
```

```
SeeedOled.sendCommand (0xA4); // output ram to display
SeeedOled.sendCommand (0xA6); // none inverted normal display
                                mode
```

```
SeeedOled.sendCommand (0xD5); // Частота оновлення екрану
```

```

SeeedOled.sendCommand (0x80); // ^ Частота

    SeeedOled.sendCommand (0x8D); // Управління внутрішнім
                                перетворювачем
SeeedOled.sendCommand (0x14); //^ 0x10 - відключити (Vcc
зовнішнє); 0x14 - включити (внутрішній DC / DC)

SeeedOled.sendCommand (0xAF); // Включити дисплей

SeeedOled.init(); // initialize SEEED OLED display

```

Метод `clearDisplay()` очищає екран і встановить початкову позицію у верхній лівий кут :

```
SeeedOled.clearDisplay();
```

Встановити режим відображення у звичайному режимі (тобто не обернений режим):

```
SeeedOled.setNormalDisplay();
```

Встановить режим адресації на режим сторінки наступний метод:

```
SeeedOled.setPageMode();
```

Встановить курсор на X-строке, Y-стовпчику:

```
SeeedOled.setTextXY (0,0);
SeeedOled.putString ( "Sensors data");
```

В останій строчці у лапках можна написати любий тестс текст.

Далі в функції `loop()` зчитуємо значення тиску і температури з датчика BMP180, виводимо їх використовуючи функції позиціонування початку виведення і власне виведення рядка на дисплей, приклад (значення з датчиків необхідно отримати раніше):



```

SeeedOled.setTextXY (2,0);
SeeedOled.putString ( "[BMP180]");
SeeedOled.setTextXY (3,0);
SeeedOled.putString (( "Pressure:" + String (event.pressure)). C_str ());

SeeedOled.setTextXY (4,0);
SeeedOled.putString (( "Temp, * C:" + String (temperature)). C_str ());

```

Прочитуємо значення вологості і температури з датчика DHT, виводимо їх аналогічно даними з датчика BMP180, тільки нижче і додаємо затримку:

```
delay (1000);
```

Результат виводу інформації представлено на рисунку 31:

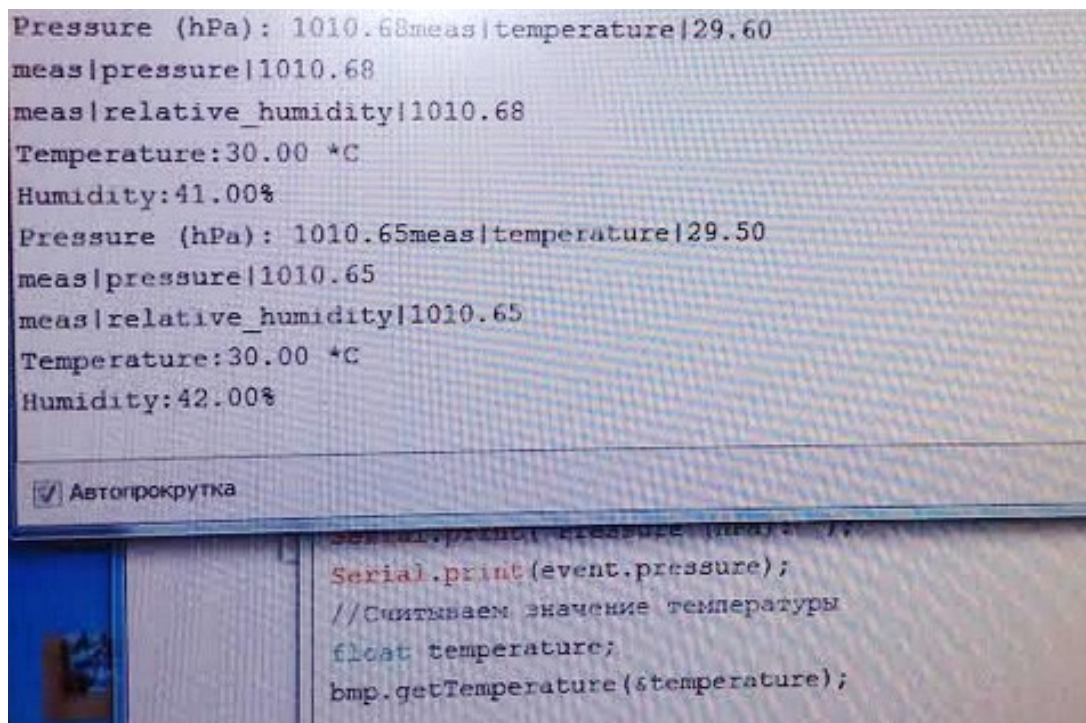


Рисунок 31 – Вивод показників на монітор порту

### 3.5 Зборка метеостанції

Після повної зборки метеостанції (всіх її компонентів) опишемо головний скетч. Спочатку підключаються необхідні бібліотеки:

```
#include <DHT.h>
#include <DHT_U.h>
#include <Adafruit_BMP085_U.h>
```

Далі: описуємо 11 пін:

```
Adafruit_BMP085_Unified bmp(10085);
DHT_Unified dht(13,DHT11);
```

Перейдемо до методу loop (): спочатку слід зчитати значення тиску.

```
void loop() {
  sensors_event_t event;
  bmp.getEvent (& event);
```

Тиск будемо виводити в hPa:

```
Serial.print ( "Pressure (hPa):");
Serial.print (event.pressure);
```

Прочитуємо значення температури:

```
float temperature;
bmp.getTemperature (& temperature);
```

Далі зчитуємо значення тиску і температури з датчика BMP180, виводимо їх використовуючи функції позиціонування початку виведення і власне виведення рядка на дисплей, приклад (значення з датчиків необхідно отримати раніше:

```
SeedOled.setTextXY (2,0);
```

У вищенаведеної строчці вказуємо номер рядка і перший символ, з якого починається висновок інформації.

```

SeeedOled.putString ( "[BMP180]");
SeeedOled.setTextXY (3,0);
SeeedOled.putString (("Pres, hPa:" + String (event.pressure)). C_str ());
SeeedOled.setTextXY (4,0);
SeeedOled.putString (( "Temp, * C:" + String (temperature)). C_str ());

```

Обов'язково враховуємо паузу виведення інформації на екран:

```

delay (5000);

dht.temperature (). getEvent (& event);

    if (isnan (event.temperature))
        Serial.println ("Error reading temperature!");
    else
        {
            Serial.print ("Temperature:");
            Serial.print (event.temperature);
            Serial.println ("* C");
        }

    dht.humidity (). getEvent (& event);

    if (isnan (event.relative_humidity))

        Serial.println ("Error reading humidity!");
    else
        {
            Serial.print ("Humidity:");
            Serial.print (event.relative_humidity);
            Serial.println ("%");
        }

```

Переходимо до наступного датчика виводу температури та рівня вологості:

```

SeeedOled.setTextXY (5,0);
SeeedOled.putString ("[DHT11]");
SeeedOled.setTextXY (6,0);
SeeedOled.putString      (("Humidity,%:" + String
(event.relative_humidity)). C_str ());
SeeedOled.setTextXY (7,0);

```

```
SeedOled.putString ("Temp, * C:" + String
(temperature)). C_str ());
```

Прочитуємо значення вологості і температури з датчика DHT, виводимо їх аналогічно даними з датчика BMP180, тільки нижче:

```
delay (5000);
}
```

Результат повної зборки схеми для метеостанції і її роботи, а саме вивод інформації на екран, представлено на рисунку 32:

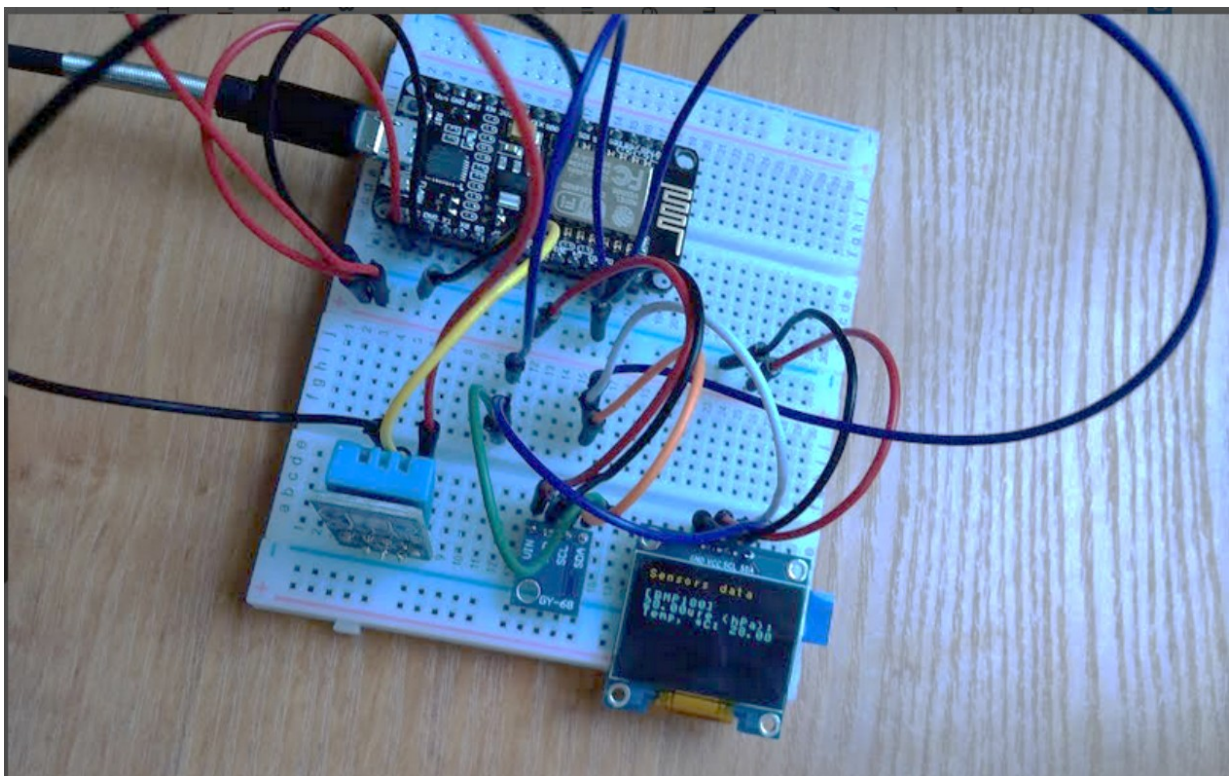


Рисунок 32 – Робота метеостанції

### 3.6 Налаштування Wi-Fi модуля

Скетч для роботи з Wi-Fi використовуємо бібліотеки, аналогічні стандартної бібліотеці Wi-Fi для звичайних Arduino. При роботі зі звичайною

Arduino і будь-яким WiFi шідд будуть підключатися аналогічні заголовки, тільки без префікса «ESP8266».

Підключаємо заголовки:

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
```

Вказуємо параметри WiFi мережі

```
const char * essid = "WIFI_ESSID";
const char * key = "WIFI_KEY";
```

WIFI\_ESSID і WIFI\_KEY замінюємо на ім'я і пароль використовуваної WiFi мережі. У функції setup () запускаємо Serial і підключаємося до WiFi

```
Serial.begin (9600);
WiFi.begin (essid, key);
while (WiFi.status () != WL_CONNECTED)
{
  delay (500);
  Serial.print ( ".");
}
Serial.println ( "WiFi connected");
```

У функції loop виконуємо GET запит на WEB-сторінку віддаленого сервера. Для цього підключаємося до сервера

```
WiFiClient client;
if (! client.connect ( "wl.unn.ru", 80))
{
  Serial.println ( "connection failed");
  return;
}
```

Відправляємо згенерований вручну GET запит

```
client.print ( "GET / laboratory /? page = 1 HTTP / 1.1 \ r \ nConnection:
close \ r \ n \ r \ n");
```

Чекаємо відповіді від сервера

```
unsigned long timeout = millis ();
while (client.available () == 0)
{
  if (millis () - timeout > 5000)
  {
    Serial.println ( ">>> Client Timeout!");
    client.stop ();
    return;
  }
}
```

Виводимо відповідь і спимо 10 секунд

```
while (client.available ())
{
  String line = client.readStringUntil ( '\ r');
  Serial.print (line);
}
delay (10000);
```

Завантажуємо скетч на плату і відкриваємо менеджер порту. Там раз в десять секунд повинен з'являтися код веб-сторінки, що прийшов у відповідь від сервера [w1.unn.ru](http://w1.unn.ru).

## ВИСНОВКИ

Отже, Arduino – це зручна платформа для реалізації проектів різної складності. Вона прийнятна як початківцям, які ще не мають навичок у сфері робототехніки, так і досвідченим користувачам.

Платформа Arduino за технічним оснащенням максимально підходить для навчального процесу з проектування різноманітних автоматизованих технічних систем та роботів, завдяки сприйнятливому середовищу програмування, можливості спостереження фізичних процесів у реальному часі.

Під час аналізу предметної області було:

- розглянуті аналоги мікроконтролерів і додаткових шилдов для них реалізації проекту;
- обрана середа проектування схем Fritzing і середа розробки скетчу IDE Arduino.

Також, під час роботи над проектом отримані навички розробки схем за допомогою програми Fritzing, завдяки якій спрощується етап розробки і тестування схеми за рахунок поетапного доповнення схеми додатковими датчиками.

Під час реалізації проекту «Метеостанція» були створені скетчі з урахуванням підключення відповідних бібліотек для датчику температури і тиску BMP180, датчику вологості DHT11 і підключено екран для виводу отриманих даних.

Даний проект планується в подальшому розвинути шляхом написання мобільного додатку, який дозволить:

- передавати дані на сервер для подальшої роботи;
- контролем і управлінням метеостанції дистанційно.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Інформаційний портал з Ардуіно. URL: <https://arduino.ua/art2-istoriya-sozdaniya-arduino>. (дата звернення 13.03.2019)
2. Офіційний сайт Arduino. URL: <https://www.arduino.cc/>.(дата звернення 13.03.2019)
3. Інформаційний довідник з Ардуіно. URL: <http://arduino-diy.com/>.(дата звернення 13.03.2019)
4. Інформаційний блог RoboCraft. URL: <http://robocraft.ru/blog/arduino/>. (дата звернення 17.03.2019)
5. Інформаційний ресурс по шилдам до Ардуіно. URL: <https://voltiq.ru/dht11-dht22-and-dht21/>. (дата звернення 19.03.2019)
6. Інформаційний довідник з Ардуіно. URL: <http://radioschema.ru/el-komponenty/spravochnik>. (дата звернення 20.03.2019)
7. Інформаційний ресурс по шилдам до Ардуіно. URL: <https://arduinomaster.ru/datchiki-arduino/datchiki-atmosfernogo-davleniya-bmp280-bmp180-bme280/>. (дата звернення 20.03.2019)
8. Інформаційний портал з Ардуіно. Характеристики світодіодів. URL: <https://arduino.ua/cat14-kraniisvetodiodi>. (дата звернення 22.03.2019)
9. Інформаційний портал з плат Ардуіно. <https://arduinomaster.ru/platy-arduino/esp8266-nodemcu-v3-lua/>. (дата звернення 24.03.2019)
10. Інформаційний портал з плат Ардуіно. URL: <https://arduino.ua/prod1804-plata-adaptera-node-mcu-esp8266-v3>. (дата звернення 24.03.2019)
11. Інформаційний блог. URL: <https://soltau.ru/index.../465-kakie-sushchestvuyut-sredy-razrabotki-ide-dlya-arduino/>. (дата звернення 28.03.2019)



12. Офіційний сайт Fritzing. URL: [fritzing.org/](http://fritzing.org/). (дата звернення 02.04.2019)
13. Інформаційний портал з Ардуіно. URL:<https://arduinomaster.ru/biblioteki-arduino/skachat-biblioteki-arduino/>. (дата звернення 05.04.2019)
14. Інформаційний портал з розробки програмного забезпечення. URL: <https://pro-prof.com/archives/2594>. (дата звернення 23.04.2019)
15. Інформаційний портал з UML-моделювання систем. URL: <https://www.lucidchart.com/pages/uml-use-case-diagram>. (дата тзвернення 23.04.2019)
16. Інформаційний ресурс з моделювання на UML. URL: <http://book.uml3.ru/>. (дата звернення 26.04.2019)
17. Офіційний сайт UML. URL: <https://www.uml.org/>. (дата звернення 26.04.2019)
18. Інформаційний ресурс. URL:[http://arduino.ru/Arduino\\_environment](http://arduino.ru/Arduino_environment). (дата звернення 02.05.2019)
19. Інформаційний портал з Ардуіно. URL:<https://arduinomaster.ru/biblioteki-arduino/skachat-biblioteki-arduino/>. (дата звернення 05.04.2019)
20. Бібліотека елементів до програми Fritzing. URL: <https://github.com/adafruit/Fritzing-Library/blob/master/parts/Adafruit%20BMP180.fzpz>. (дата звернення 08.05.2019)