

**Метеостанція**

## **Зміст**

1. Загальна інформація про виконану роботу.....	3 с.
2. Коротка інформація про метеостанцію.....	4 с.
3. Мануал.....	6 с.
4. Загальна схема.....	8 с.
5. Барометр BMP280.....	9 с.
6. LCD2004.....	11 с.
7. RTC.....	13 с.
8. SD card module.....	15 с.
9. Гігрометр DHT11.....	17 с.
10. Esp8266.....	19 с.

# **Загальна інформація про виконану роботу**

## **Поставлена задача**

Створити просту метеостанцію, котра буде вимірювати температуру, тиск і вологість повітря, поточний час та виводитиме покази на графічний дисплей.

## **Виконана робота**

Створено метеостанцію на базі STM32F411, яка вимірює тиск, вологість повітря та температуру. Для роботи з гігрометром DHT11, барометром BMP280 та дисплеєм LCD2004 використовуються три відповідні бібліотеки. Також реалізовано вивід на сервер за допомогою WiFi модуля NodeMCU на базі ESP8266. Дані метеостанція виводить на екран та на сервер (поки локальний), з цього ж сервера можна нею керувати. Сервер, крім актуальних даних, також, за допомогою графіків відображає історію змін за останніх кілька годин після запуску метеостанції.

## **Переваги нашого рішення**

- Економічний аспект:
  - Вартість всього обладнання становить 1000-1500 грн
  - Вартість готового рішення на базі плат Arduino -- від 2700 грн
- Швидкість роботи:
  - вбудований мікроконтролер, STM32F411, здатен реалізовувати складніші алгоритми, зокрема, такі, що потребують більшої обчислювальної потужності, ніж мікроконтролери AVR8, використовувані в Arduino.
  - так як код написаний на C, із використанням лише HAL, кількість проміжних рівнів абстракції менша а ефективність коду помітно вища, ніж із використанням бібліотек Arduino. Зворотною стороною цього є ускладнення розробки.
- Навчання:

- під STM32 написано значно менше бібліотек ніж під Arduino, тому потрібно робити все самому або модифікувати готові варіанти

## **Коротко про метеостанцію**

- Корпус виготовлений з фанери
- Метеостанція може переключатись між трьома режимами роботи(кнопкою або через сервер):
  - Full Power Mode: на екрані включається підсвітка, режим барометра - нормальний, частота зчитування даних з барометра стає 150 Гц.
  - Low Power Mode - вимикається підсвітка на екрані, частота барометра стає 15 Гц.
  - Power off - повністю вимикає дисплей, але зберігає останні символи, барометр переводиться в режим сну, після чого чекає на сигнал WiFi модуля або кнопки.

В такому режимі метеостанція не вимикає WiFi модуль, оскільки за допомогою нього здійснюється керування нею.

## **Нормалізація даних**

Здійснюється нормалізація даних: читається певна кількість вимірів, знаходяться і відкидаються найбільше та найменше із них, після чого шукає середнє значення елементів, які залишилися. Аж після такої обробки дані передаються на LCD дисплей та публікуються на сервері.

# Мануал

Для того щоб запустити метеостанцію Вам необхідно:

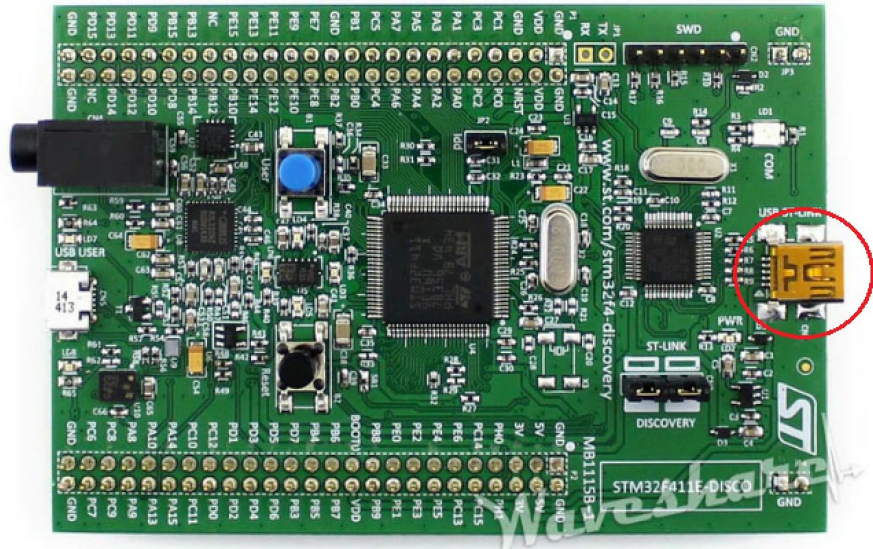
- Замінити батарейки або вставити mini usb у відповідний роз'єм
- Натиснути кнопку ввімкнення
- Підключитися до сервера. Для цього вам необхідно назвати свою точку доступу NodeMCU та ввести пароль вісім одиниць - 11111111
- Включити на телефоні інтернет
- Запустити метеостанція і почекати поки в першому рядку з'явиться ір сервера, якщо воно не змінилося з 0.0.0.0, то ви не правильно створили точку доступу Wi-Fi
- Перейти за даним ір в інтернет браузері
- Ввести необхідний час на сервері
- Насолоджуватись даними

Для роботи з метеостанцією Вам необхідно:

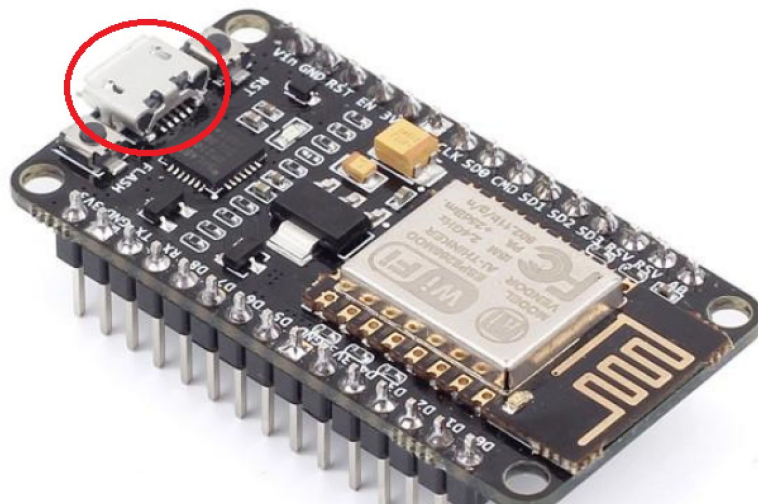
- Середовище System Workbench for STM32 або аналогічне
  - Для того щоб мати можливість користуватися semihosting
    - <https://indrekis2.blogspot.com/2017/10/semihosting.html>
- Для програмування WiFi-модуля знадобиться також середовище Arduino IDE:
  - Виконати всі кроки за наступним посиланням:
    - <https://www.teachmemicro.com/intro-nodemcu-arduino/>
  - Встановити бібліотеку Adafruit ESP8266
- Генератор конфігурацій STM32CubeMX
- В проєкті потрібно **обов'язково** додати підтримку floating-point
- Якщо годинник RTC часто збивається, то замініть батарейку всередині RTC модуля
- Якщо щось не працює відкрийте корпус і перевірте всі з'єднання -- відомо, що переважна кількість проблем електроніки це погані контакти.

- Для того щоб завантажити прошивку на відповідний мікроконтролер, зніміть задню кришку та вставте кабель mini usb(micro usb) в відповідний вихід stm32f411(NodeMCU). Нагадаємо виглядають вони так:

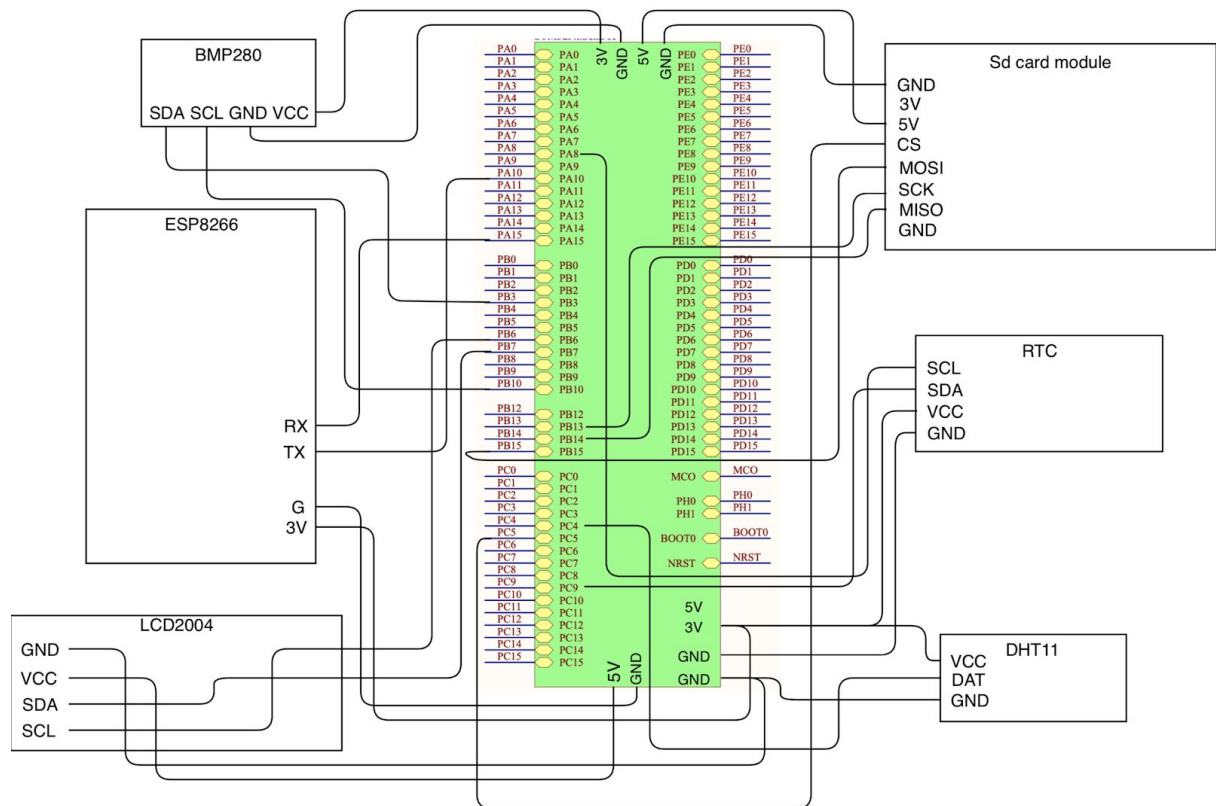
- stm32f411 discovery:



- NodeMCU:

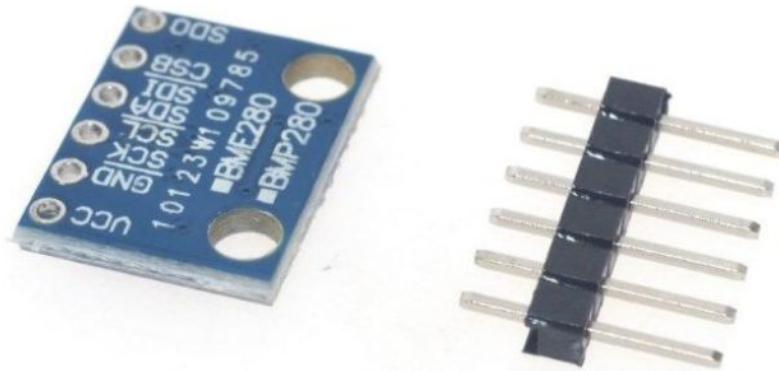


## Загальна схема





## BMP280



### Опис:

BMP280 - датчик тиску та температури повітря. За роботу цього датчику відповідають два файли `bmp280.c` та `bmp280.h`.

### Основні характеристики:

- межі вимірювання тиску 300...1100ГПа, похибка  $\pm 1$ ГПа
- межі вимірювання температури  $-40...+85^{\circ}\text{C}$ , похибка  $\pm 1^{\circ}\text{C}$
- SPI або I2C інтерфейс

### BMP280 має 3 основні режими роботи:

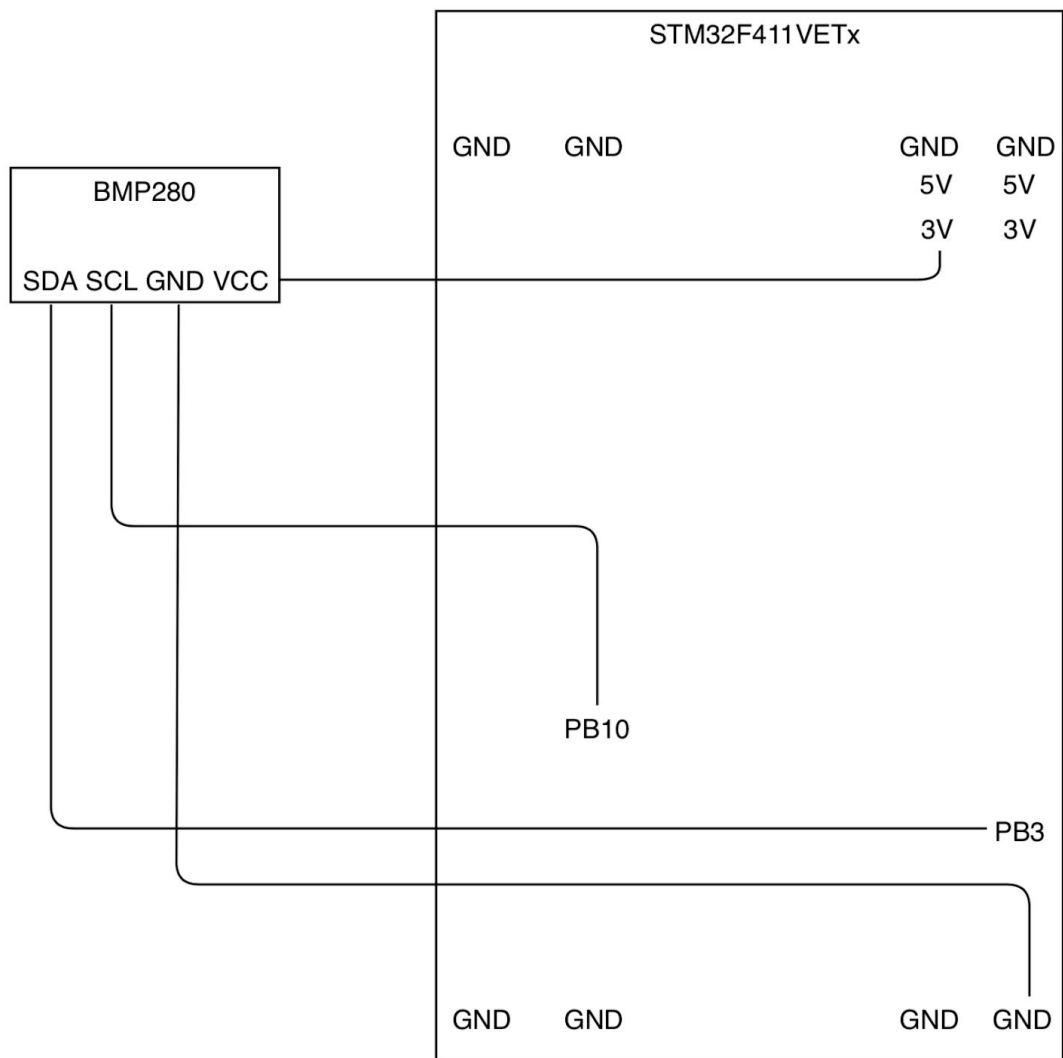
- sleep - режим сну
- forced - режим, при якому дається команда на замір і через певний час зчитуються дані, після чого датчик переходить в режим сну
- normal - циклічний режим, при якому датчик через певний час(його можна вказати) виходить з режиму сну, бере заміри, заносить їх в регістри і знову засинає на вказаний час

### Принцип роботи:

Спочатку зчитуються дані з 16 регістрів. Всі дані типу `int16_t`, крім першого регістру для температури та першого для барометра, які мають тип `uint16_t`. Після цього спеціальна функція перетворює вміст регістрів у floating-point, про яку можна прочитати в офіційній інструкції(datasheet). Для роботи з цими даними використовуються кілька структур, за допомогою яких перевіряється чи ми працюємо з

потрібним датчиком, яка його адреса на шині, час останніх замірів і т.д.

Схема підключення(в нашому випадку використовується інтерфейс i2c):



## LCD2004



Опис:

LCD2004 - LCD дисплей.

Основні характеристики:

- можливість вивести максимум 4 стрічки по 20 символів на екран
- LED підсвітка. Якщо цифри або букви видно нечітко, то покрутіть бігунок, який знаходиться на мікросхемі
- I2C інтерфейс

Принцип роботи:

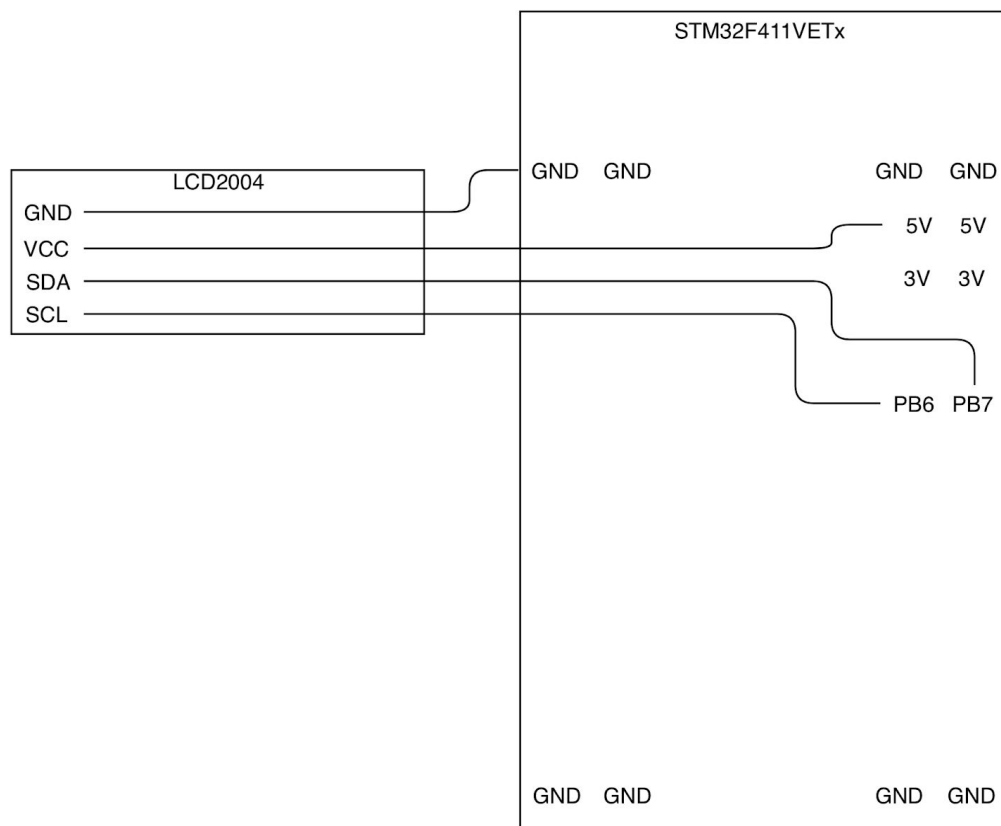
Принцип роботи з даним девайсом достатньо простий. Використовується інтерфейс I2C з мікросхемою PCF8574. Для роботи з дисплеєм було написано 2 бібліотеки, більшість функцій спільні, відрізняється лише принцип запису даних. Одна бібліотека виводить дані посимвольно, проте, коли перший раз було виведено 1000, а потім 999 і отримано 1999, ми зрозуміли, що це незручно, тому при роботі з нею необхідно кожен раз очищати дисплей і ставити затримку, оскільки останні дані в 4 рядку не встигатимуть виводитись. Друга ж бібліотека зберігає попередній вміст і перед тим як вивести нові дані на дисплей звіряється, чи вони однакові, якщо так - вона пропускає поточний символ. Першу бібліотеку Ви можете знайти в папці Additional libraries. Для роботи з цим дисплеєм спочатку вам треба ініціалізувати його, тобто визначити

номер I2C, його адресу на шині, очистити необхідні, для роботи, масиви, а також сам дисплей, встановити поточну адресу, куда будуть записуватися дані. Дисплей приймає дані посимвольно, тому кожен символ потрібно надсилати окремо. Після чого необхідно надіслати дані у новий масив і викликати функцію LCD\_Print() - на дисплей виведуться дані. За рахунок другої бібліотеки ви можете очищувати тільки потрібний вам рядок або ж і символ і екран змінить тільки його.

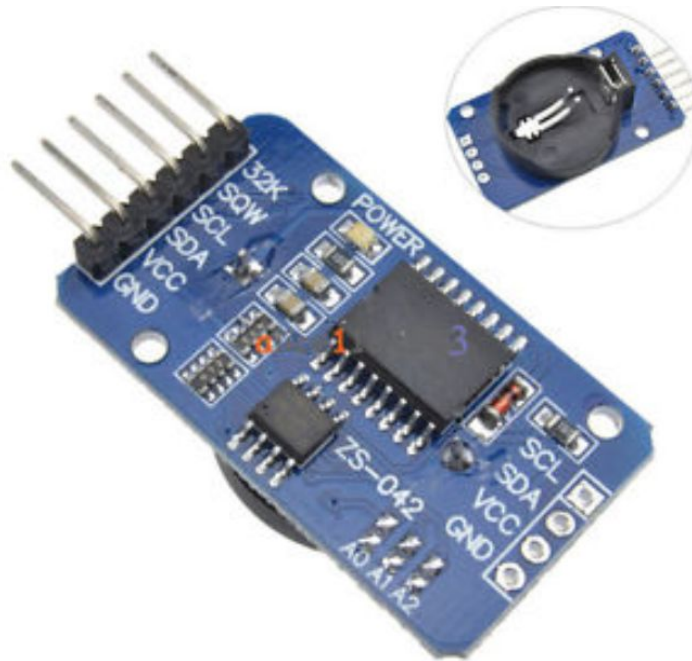
Декілька слів про передачу даних. Якщо на піні RS встановлено 0, то дані сприймаються як команди. Даний пін знаходиться на самому екрані, а не на платі I2C. Дані передаються посимвольно, за допомогою Hal.

Можна дописати багато функцій, які дозволять писати справа наліво і тд.

Схема підключення:



## RTC



Опис:

RTC(real-time clock) - годинник реального часу.

Основні характеристики:

- рахує секунди, хвилини, години, день місяця, місяць, день, тиждень і рік(включаючи високосний до 2100р)
- I2C інтерфейс
- експлуатується в умовах від -40 до +85°C
- для роботи потрібна батарейка CR2032

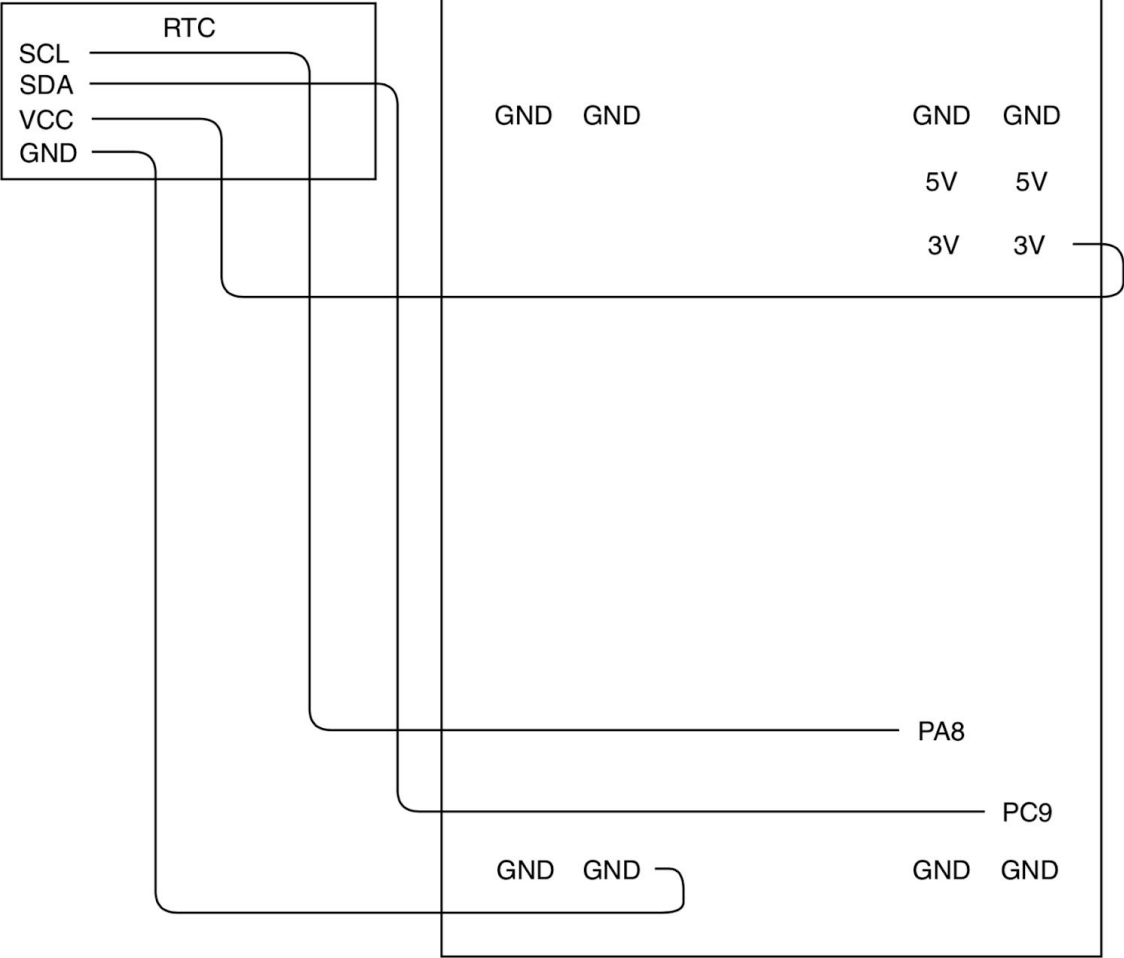
Принцип роботи:

Найпростіший з усіх девайсів. За допомогою вбудованої функції зчитує регістри з годинника, а потім переводить з бінарного в десятковий вигляд.

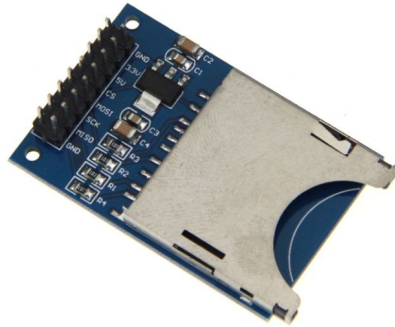
**УВАГА!** Годинник **не** вміє показувати рік у форматі “2019”, він розуміє тільки як 19.

Вводити час можна за допомогою сервера.

Схема підключення:



## SD card module



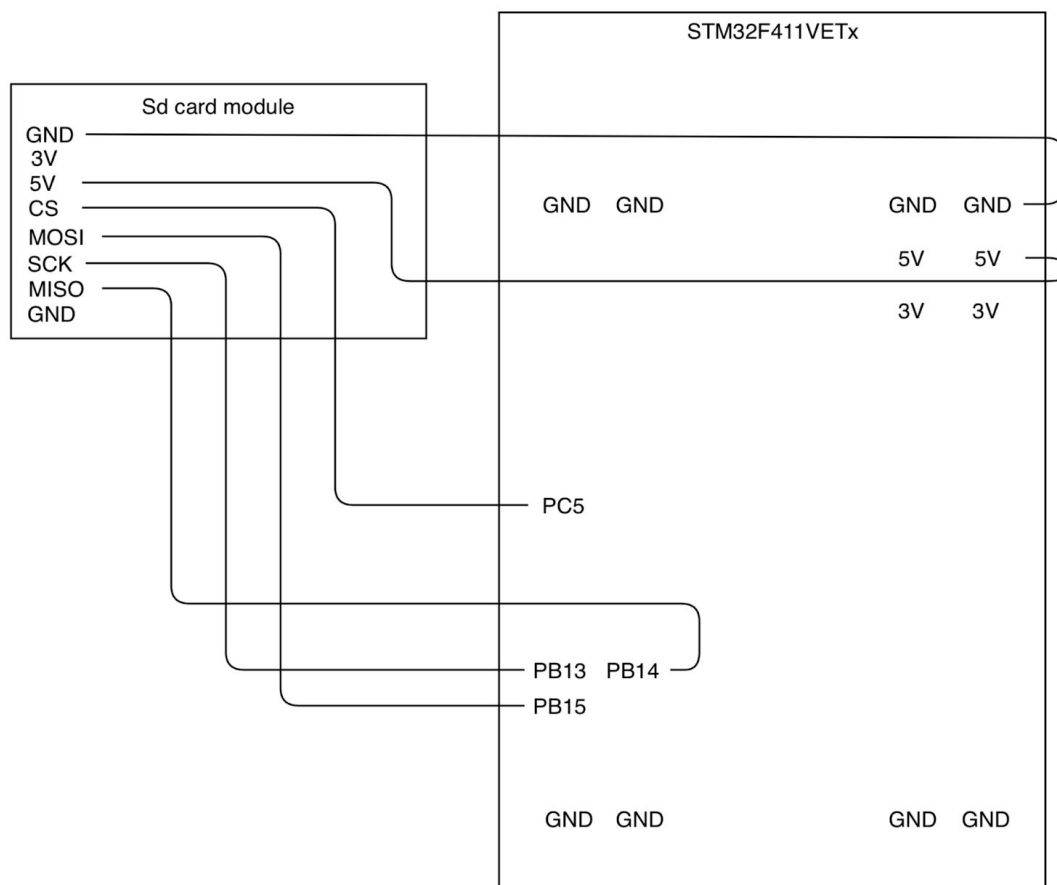
Опис:

Модуль SD картки для запису та зчитування даних.

Основні характеристики:

- може працювати на швидкості від 78.125 KBits/S до 10 MBits/S
- працює за доп. вбудованої бібліотеки FATFS та протоколом SPI

Схема підключення:

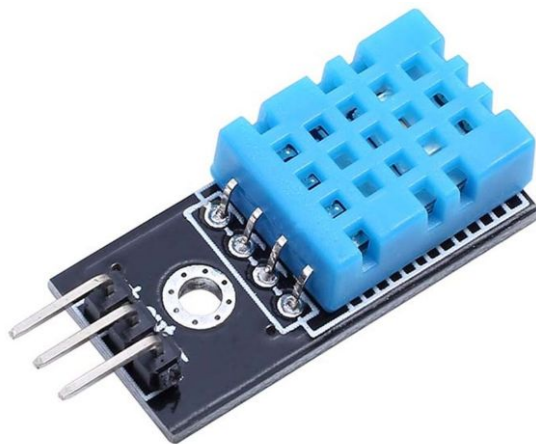


Принцип роботи:

Спочатку ініціалізується сама sd картка, її тип, швидкість передачі даних та сам spi. Зчитування або запис на неї здійснюються наступним чином: спочатку надсилається відповідна команда на sd карту, а потім самі дані. Після запису і зчитування файл закривається, а сама sd картка переходить в безпечний для витягнення режим.



## DHT11



Опис:

Гігрометр - пристрій для вимірювання вологості повітря.

DHT11 також вимірює температуру повітря, що потрібно для вимірювання самої вологості.

Основні характеристики:

- використовується сенсор “модуль для Arduino”
- температура в градусах Цельсія
- вологість у відсотках
- похибка температури 1 градус Цельсія

Отримання даних:

Загалом сенсор передає 40 біт даних, які в свою чергу розбиваються на 5 байт. Старший біт в байті йде першим.

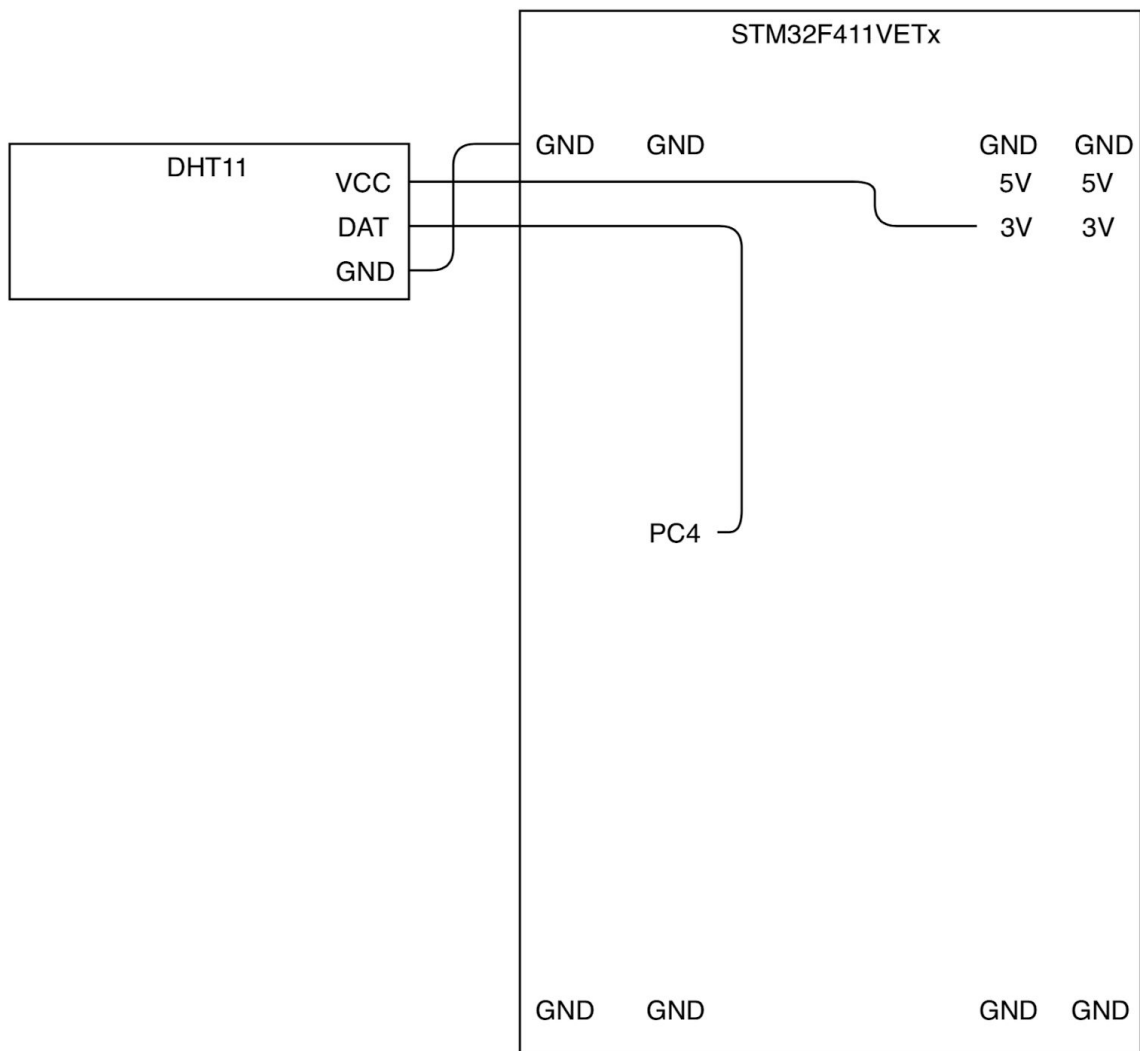
1. Ціла частина вологості.
2. Дробова частина вологості.
3. Ціла частина температури.
4. Дробова частина температури.
5. Контрольна сума. (8 біт, які виведені на те, щоб перевірити попередні 32 біти).

Байти 2 та 4 завжди нульові, тому ‘єдиний знак після коми буде 0 (23.0C, 69%).

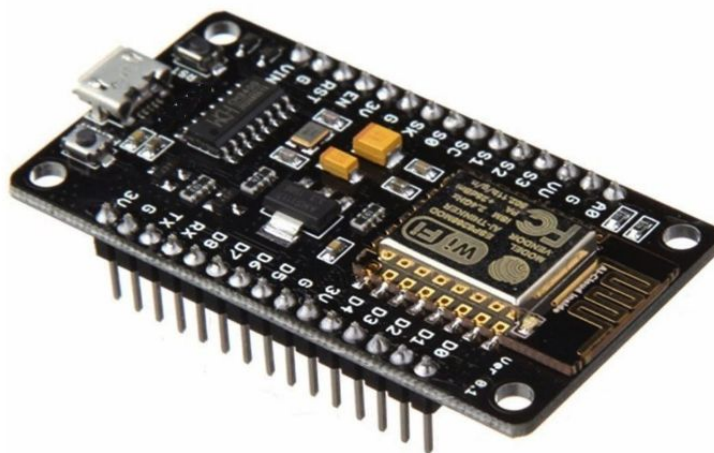
Принцип роботи:

Для роботи з цим датчиком використовувалась бібліотека. На початку роботи необхідно ініціалізувати датчик, після чого слід почекати 2 секунди, для належної роботи датчика. Після цього ми можемо зчитувати кожну секунду дані. Датчик передає імпульси, після чого ми їх побітово зчитуємо і переводимо в ті що нам потрібно. Останній біт нам не потрібен.

Схема підключення:



## ESP8266



Опис: вай-фай модуль.

Основні характеристики:

- працює за протоколом UART
- швидкість передачі даних 115200 Bits/s. **Важливо** встановити цю частоту як на stm32f4, так і на NodeMCU. Якщо дані приходять “погано” -- містять некоректні значення, дані взагалі не приходять, або в аналізаторі сигналів вибиває framing error, перевірте чи правильні піни використовуєте. Можна також спробувати можете змінити їх.
- для живлення потребує 3 V.
- пам’ятайте, що TX слід під’єднати до RX, а RX до TX.

Принцип роботи:

Спочатку на stm32f4 ініціалізується з’єднання по UART. Після чого stm32f4 в потрібний момент починає передачу даних по уарту, а NodeMCU приймає їх. При цьому NodeMCU та stm32f4 постійно перевіряють чи можуть вони прийняти якісь дані. Часто буває, що дані приходять пошкоджені або неповні. Щоб уникнути цього ми використовуємо сертифікат. Зазвичай ми посилаємо дані масивом uint8\_t, який складається з трьох елементів. Перший id елемента, який унікальний для кожного елемента, наприклад якщо відправляємо цілу частину температури з гігрометра, ми вказуємо, що id = 0, для секунди 10 і т.д. Далі йдуть два значення чому рівна та сама температура в даний момент, або чому рівна секунда. Вони йдуть два рази для того, щоб якщо прийшла помилка, то одне значення не дорівнювало іншому і NodeMCU знала, що це помилка. Також на сервері є три кнопки, за допомогою яких можна міняти режим праці stm32f4, або встановлювати необхідну годину. Є ще чотири кнопки, за допомогою яких ми можемо вибрати, графік чого саме ми хочемо бачити.

Схема підключення:

