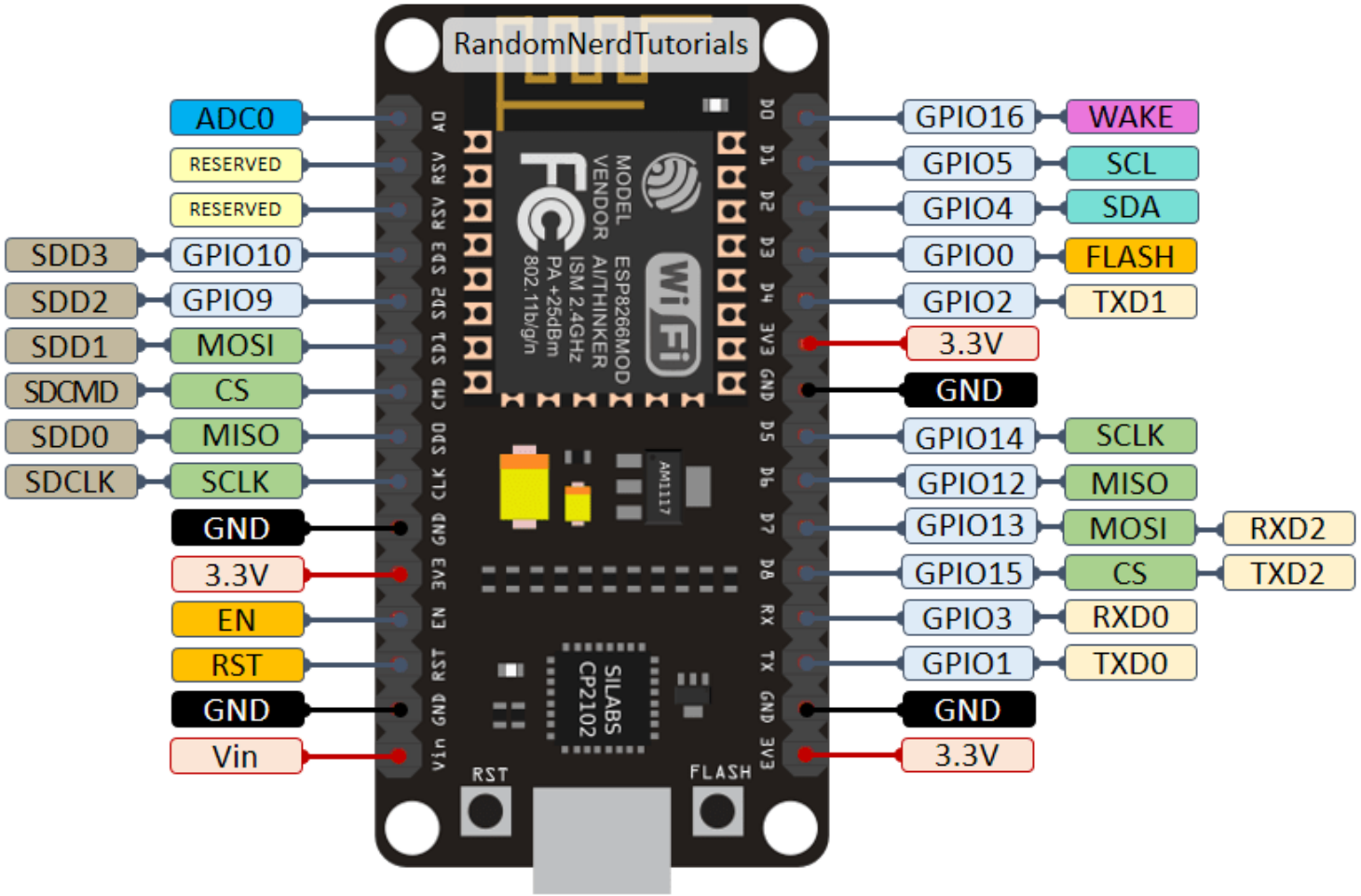

Метеостанція та
пожежна система
на базі
NodeMCU +
Arduino Nano

Використані компоненти. NodeMCU



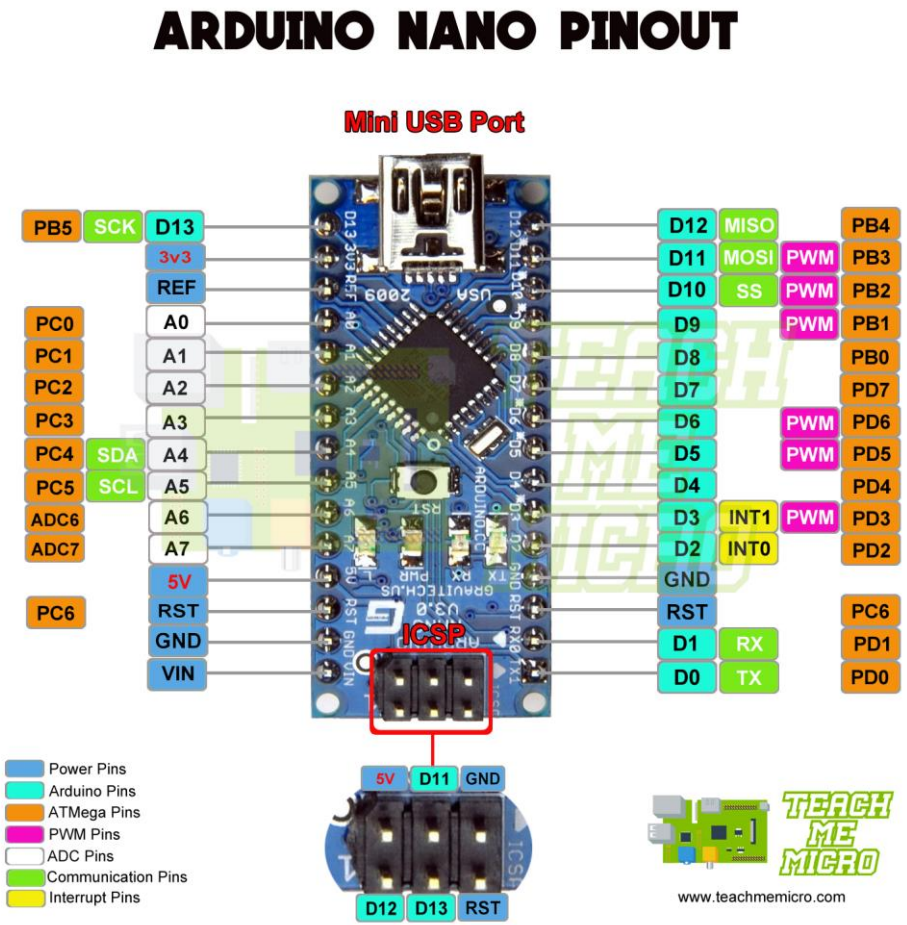
WiFi Module (ESP8266)

Web Server

Sensor Collector (master)

Main Logic

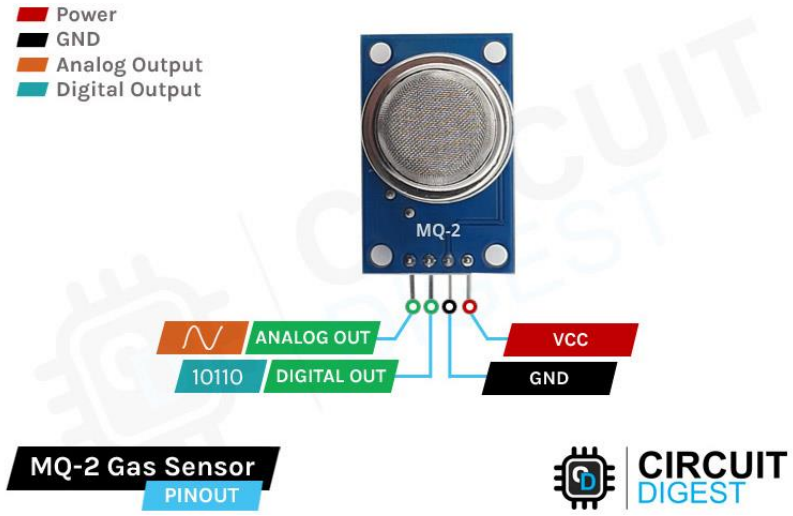
Використані компоненти. Arduino Nano

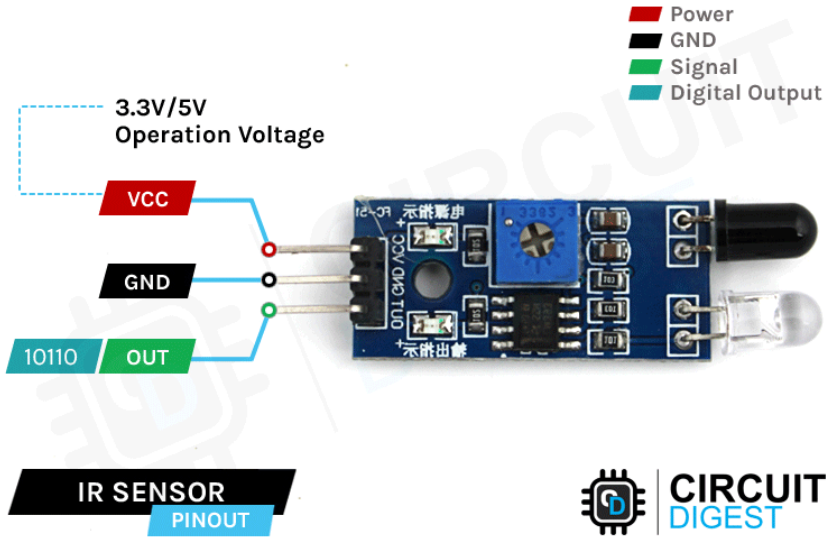


Power (5V)

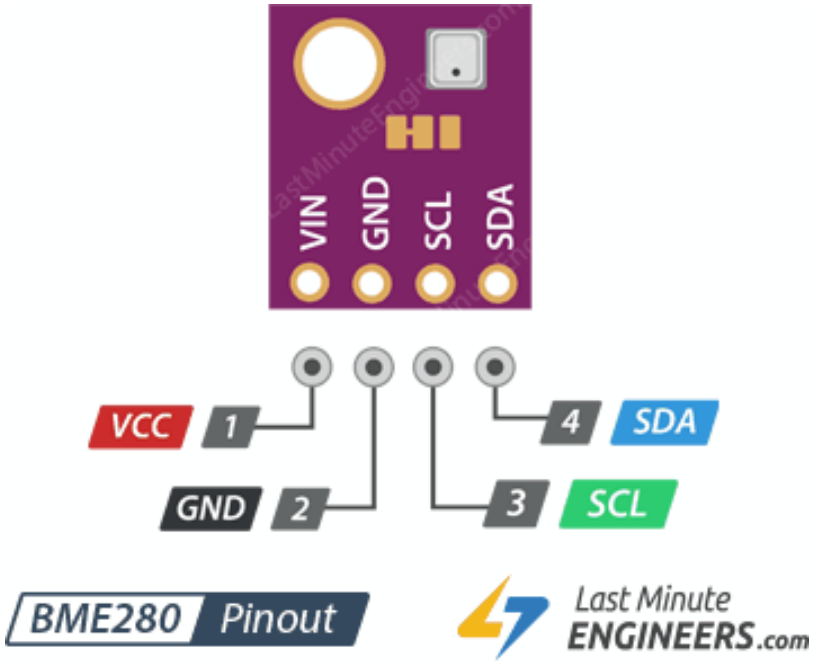
I2C Led Screen Handler

Використані компоненти. Сенсори

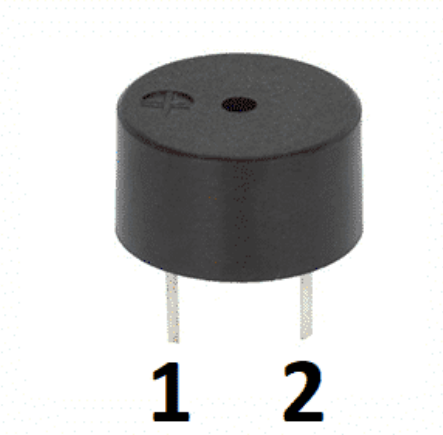




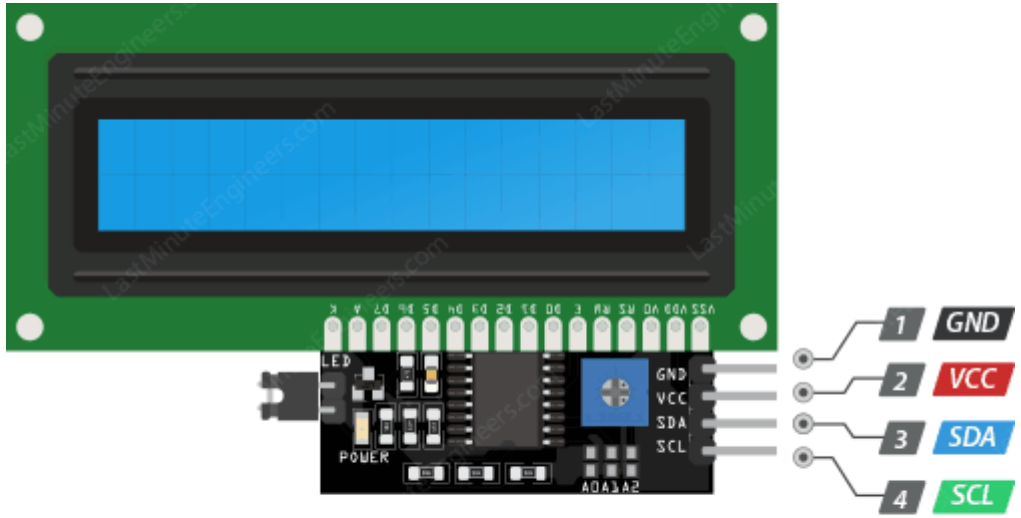
Використані компоненти. Сенсори



Використані компоненти. Актори

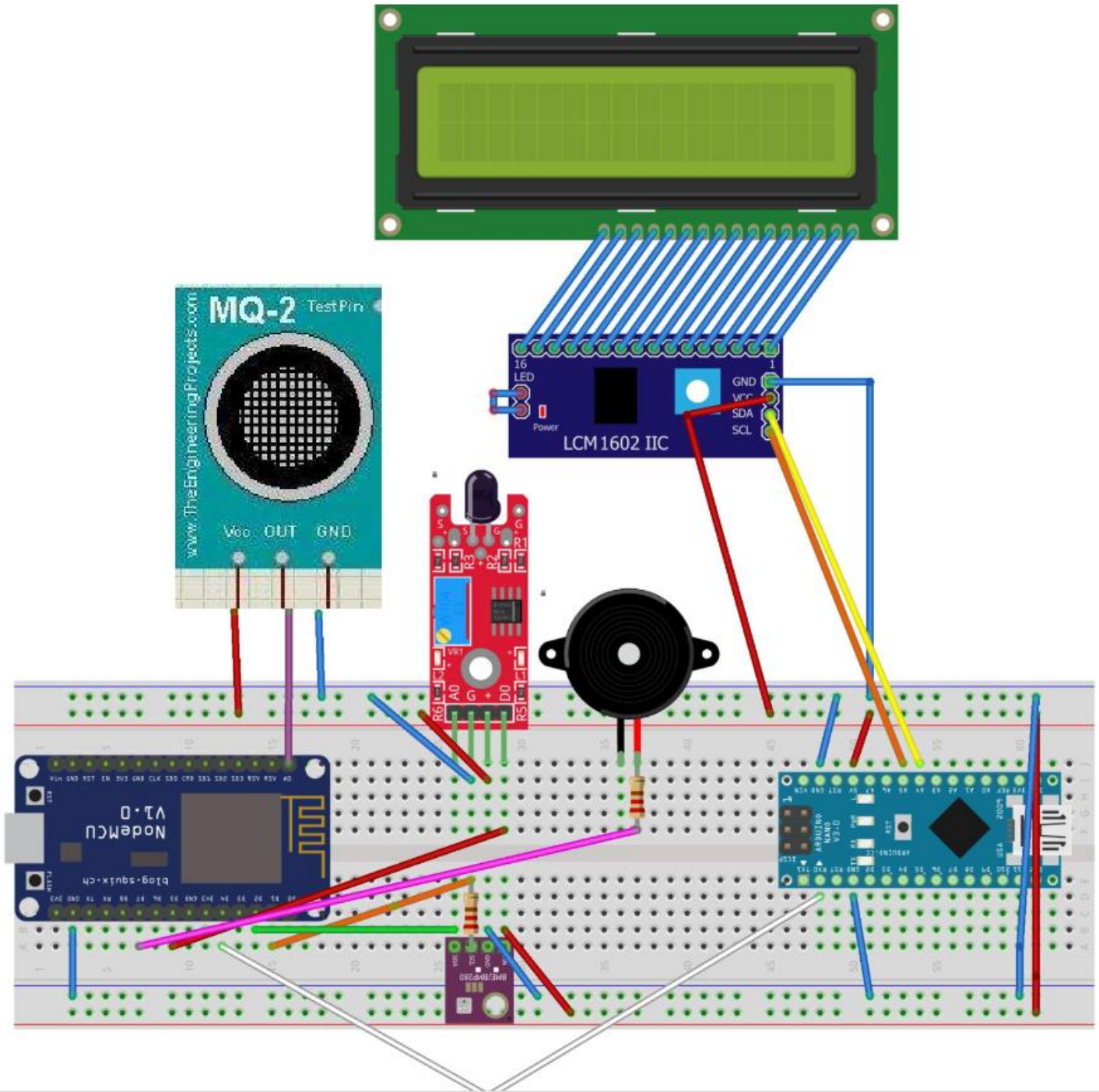


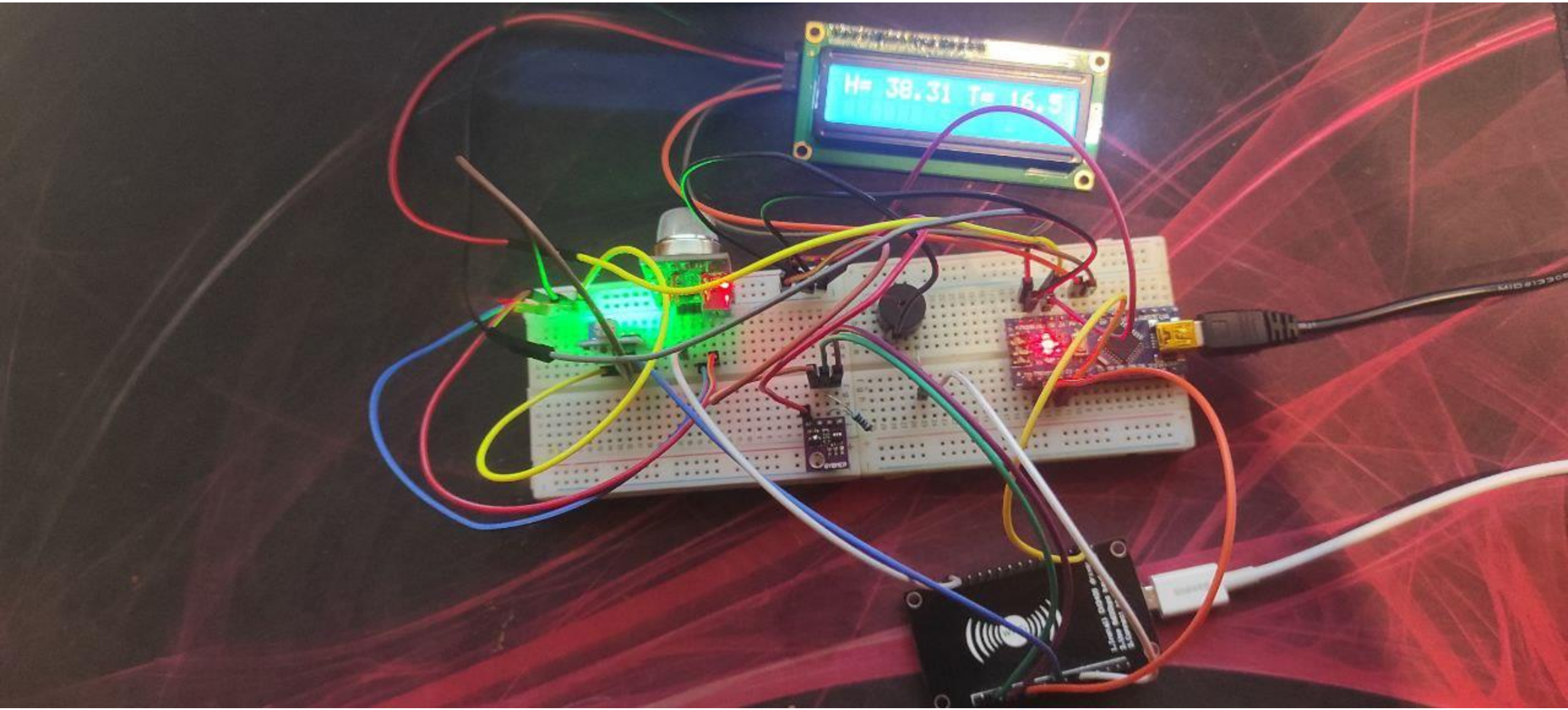
Pin1 - VCC
Pin2 - GND



I2C LCD Pinout

Last Minute ENGINEERS.com


MQ-2
TestPin
www.TheEngineeringProjects.com
Vcc OUT GND
LCM1602 IIC
16
LED
Power
1
GND
VCC
SDA
SCL
NodeMCU
V1.0
blog-squix-ch
A0 G + D0
VR1
R1
R2
R3
R5
R6

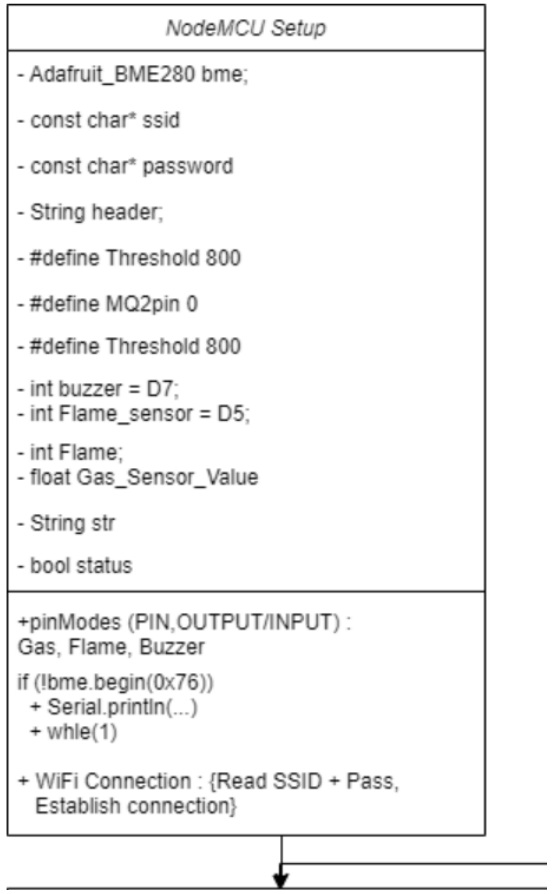

H= 38.31 T= 16.9

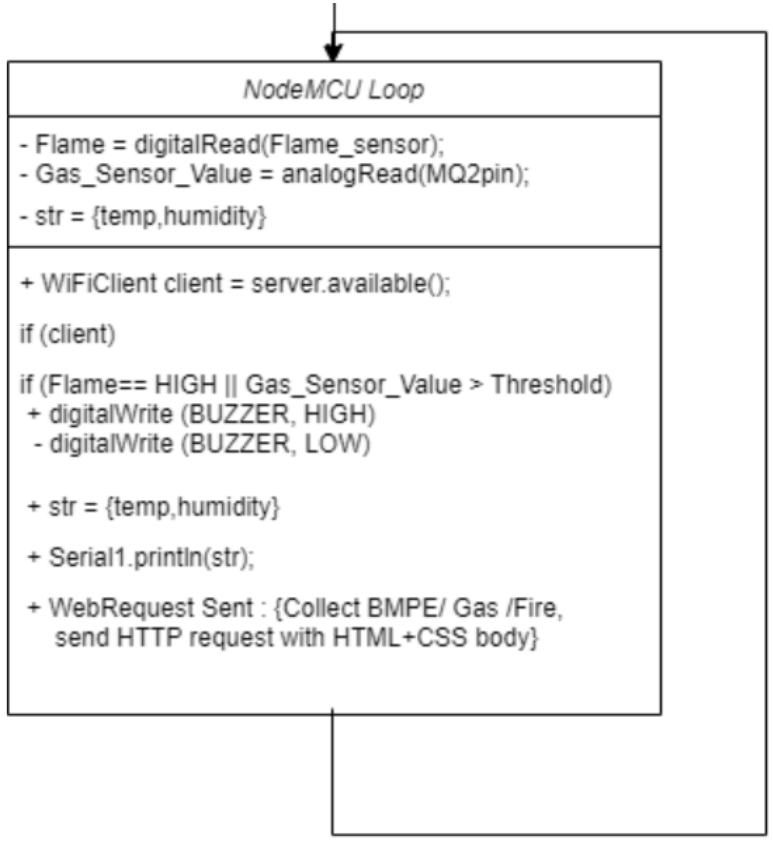
Веб Інтерфейс

Weather Station on BME280 and NODEMCU

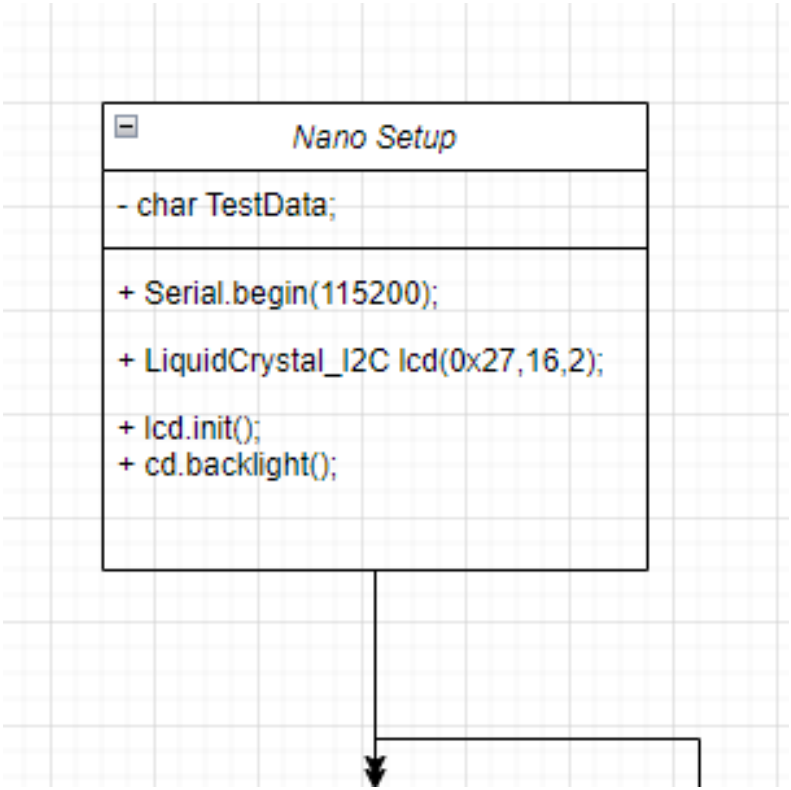
Name	Value
Temperature	**18.70 *C**
Pressure	**992.04 hPa**
Level above the sea line	**178.14 m**
Humidity	**40.95 %**
Gas Level in the Air	**675.00 CO2 Level**
Flame?	**0 Flame_detected**

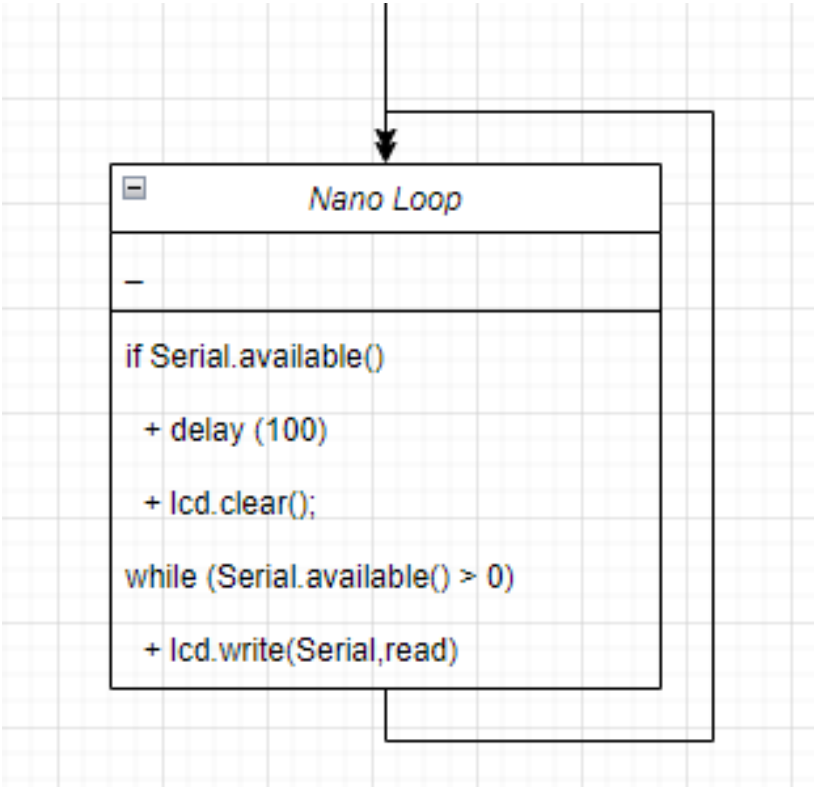
Діаграма коду. NodeMCU





Діаграма коду. Arduino





Відкриття серверу на прослуховування

```
if (!bme.begin(0x76)) {                                  // Проверка инициализации датчика
  Serial.println("Could not find a valid BME280 sensor, check wiring!"); // Печать, об ошибки инициализации.
  while (1);                                             // Зацикливаем
}

Serial.print("Connecting to ");                           // Отправка в Serial port
Serial.println(ssid);                                    // Отправка в Serial port
WiFi.begin(ssid, password);                              // Подключение к WiFi Сети
while (WiFi.status() != WL_CONNECTED) {                  // Проверка подключения к WiFi сети
  delay(500);                                            // Пауза
  Serial.print(".");                                     // Отправка в Serial port
}

Serial.println("");                                      // Отправка в Serial port
Serial.println("WiFi connected.");                       // Отправка в Serial port
Serial.println("IP address: ");                          // Отправка в Serial port
Serial.println(WiFi.localIP());                          // Отправка в Serial port
server.begin();


oid loop(){

WiFiClient client = server.available();                  // Получаем данные, посылаемые клиентом
```

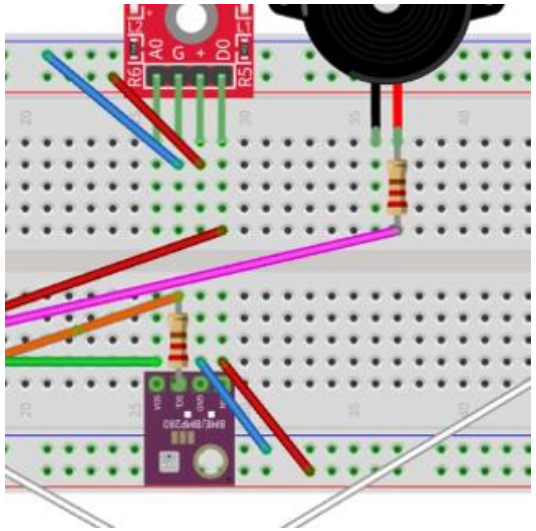
Опис запиту до кліента

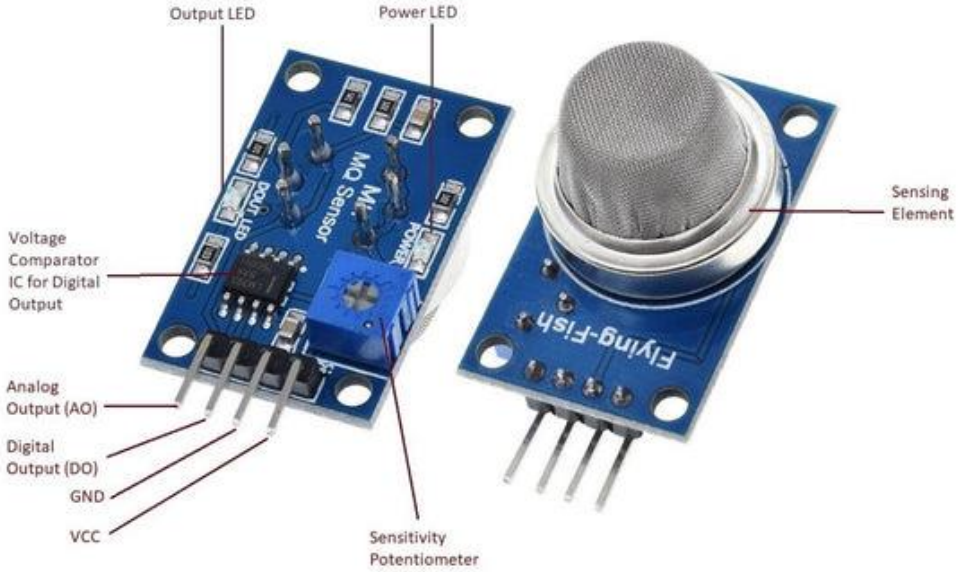
```
Serial.write(c);                                // Отправка в Serial port
header += c;
if (c == '\n') {                                // Вывод HTML страницы
  if (currentLine.length() == 0) {
    client.println("HTTP/1.1 200 OK");          // Стандартный заголовок HT
    client.println("Content-type:text/html ");
    client.println("Connection: close");        // Соединение будет закрыто после завершения ответа
    client.println("Refresh: 10");              // Автоматическое обновление каждые 10 сек
    client.println();

    client.println("<!DOCTYPE html><html>");
    client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">");
    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: transparent ; padding: 1px; }");
```

Опис запиту до кліента

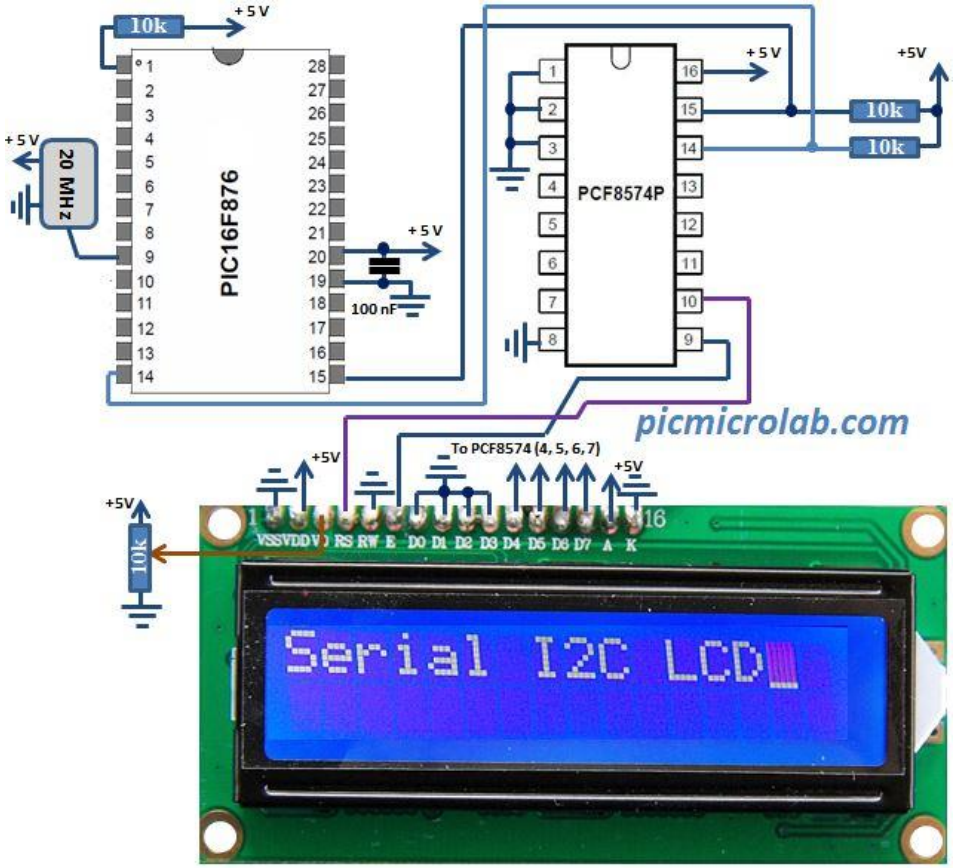
```
client.println("</style></head><body><h1>Weather Station on BME280 and NODEMCU</h1>");
client.println("<table><tr><th>Name</th><th>Value</th></tr>");
client.println("<tr><td>Temperature</td><td><span class=\"sensor\">");
client.println(bme.readTemperature());
client.println(" *C</span></td></tr>");
client.println("<tr><td>Pressure</td><td><span class=\"sensor\">");
client.println(bme.readPressure() / 100.0F);
client.println(" hPa</span></td></tr>");
client.println("<tr><td>Level above the sea line</td><td><span class=\"sensor\">");
client.println(bme.readAltitude(SEALEVELPRESSURE_HPA));
client.println(" m</span></td></tr>");
client.println("<tr><td>Humidity</td><td><span class=\"sensor\">");
client.println(bme.readHumidity());
client.println(" %</span></td></tr>");
client.println("<tr><td>Gas Level in the Air</td><td><span class=\"sensor\">");
client.println(Gas_Sensor_Value);
client.println(" CO2 Level</span></td></tr>");
client.println("</body></html>");
client.println("<tr><td>Flame?</td><td><span class=\"sensor\">");
client.println(Flame);
client.println(" Flame_detected</span></td></tr>");
client.println("</body></html>");
```

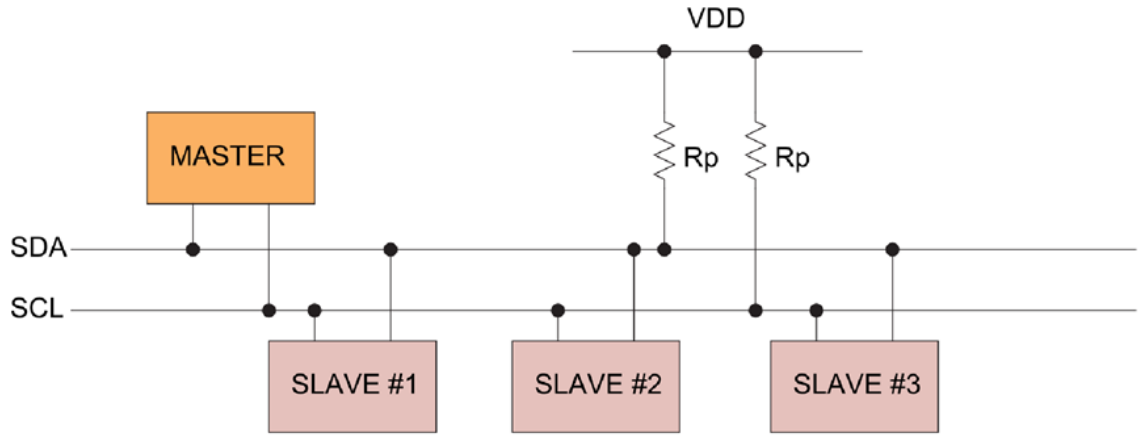
Апаратний захист. Резистори та потенціометри





Апаратний захист. Стандарт I2C





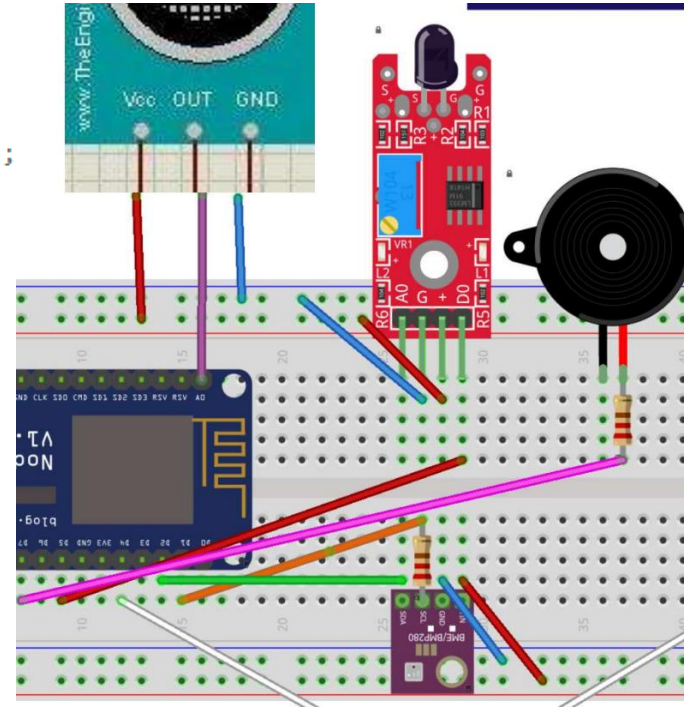
Програмний захист. Реакція на сенсори

```
WiFiClient client = server.available();

if (client) {

  Flame = digitalRead(Flame_sensor);
  Gas_Sensor_Value = analogRead(MQ2pin);

  if (Flame== HIGH)
  {
  digitalWrite(buzzer, HIGH);
  }
  else
  {
  digitalWrite(buzzer, LOW);
  }
```

```
if (!bme.begin(0x76)) {
  Serial.println("Could not find a valid BME280 sensor, check wiring!");
  while (1);
}
```

```
06:22:29.227 -> I2C device found at address 0x27  !
06:22:29.266 -> done
```

Flame?	1 Flame_detected



Інший функціонал