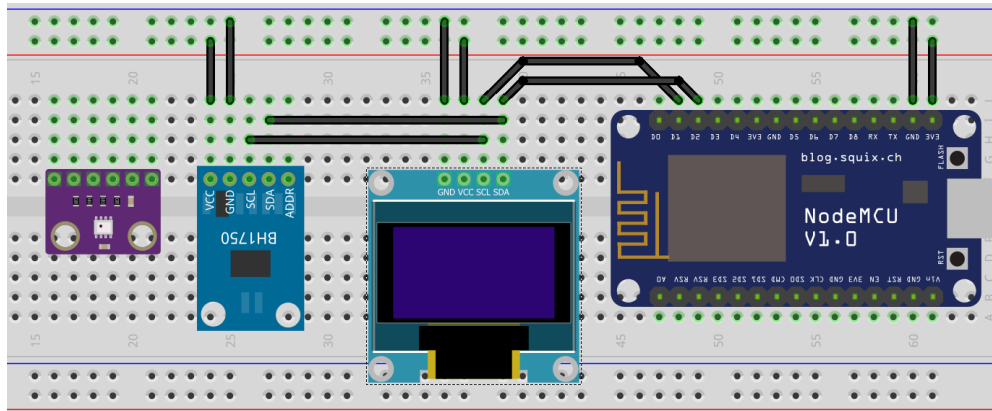


**Zad.1.** W środowisku Arduino IDE należy dodać następujące biblioteki:

- ESPAsyncTCP
- ESPAsyncWebServer
- ESPDash



**Zad.2.** Korzystając z układu z zadania 1 należy sprawdzić działanie poniższego programu.

```
#include<ESP8266WiFi.h>
#include<ESPAsyncTCP.h>
#include<ESPAsyncWebServer.h>
#include<Wire.h>
#include<BH1750.h>
#define BAUDRATE 115200

const char* ssid = "WSG-Mobilna pracownia";
const char* pass = "M0b1l1naPracown1a";

AsyncWebServer server(80);

BH1750 light_sensor;
double lux = 0;

const char index_html[] PROGMEM = R"webpage(
<!DOCTYPE html><html>
<head>
  <meta name='viewport' content='width=device-width,
initial-scale=1.0'>
```

```
<title>ESP8266 Lux meter</title>
<link
href='https://fonts.googleapis.com/css?family=Open+Sans:300,400
,600' rel='stylesheet'>
<style>
  html { font-family: 'Open Sans', sans-serif; display: block;
margin: 0px auto; text-align: center;color: #444444;}
  body{margin: 0px;}
  h1 {margin: 50px auto 30px;}
  .side-by-side{display: table-cell;vertical-align:
middle;position: relative;}
  .text{font-weight: 600;font-size: 19px;width: 200px;}
  .reading{font-weight: 200;font-size: 50px;padding-right:
25px;}
  .light.reading{color: #F29C1F;}
  .superscript{font-size: 17px;font-weight: 600;position:
absolute;top: 10px;}
  .data{padding: 10px;}
  .container{display: table;margin: 0 auto;}
  .icon{width:65px}
</style>
<script>
setInterval(updateLux,1000);
function updateLux() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if(this.readyState == 4 && this.status == 200) {
      document.getElementById("light_int").innerHTML =
        this.responseText; } };
  xhttp.open("GET","light", true);
  xhttp.send(); }
</script>
</head>
<body>
  <h1>ESP8266 Lux meter</h1>
  <div class='container'>
    <div class='data light'>
      <div class='side-by-side icon'>
        <svg xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" version="1.1"
width="60" height="60" viewBox="0 0 60 60" xml:space="preserve">
          <g transform="matrix(0.39 0 0 0.39 29 29)"><path style="stroke:
rgb(238,238,40);
```

```
stroke-width: 8; stroke-dasharray: none; stroke-linecap: butt;
stroke-dashoffset: 0;
stroke-linejoin: miter; stroke-miterlimit: 10; fill:
rgb(238,238,40); fill-rule: evenodd; opacity: 1;"
transform=" translate(-59.83, -59.83)" d="m 0 59.82677 1
27.527777 -8.654488 1 -19.512508 -21.258898 1 28.167 6.268881 1
-6.268881 -28.167 1 21.258898 19.512508 1 8.654488 -27.527777 1
8.654491 27.527777 1 21.258896 -19.512508 1 -6.2688828 28.167 1
28.167 -6.268881 1 -19.512512 21.258898 1 27.527779 8.654488 1
-27.527779 8.654491 1 19.512512 21.258896 1 -28.167 -6.2688828 1
6.2688828 28.167 1 -21.258896 -19.512512 1
-8.654491 27.527779 1 -8.654488 -27.527779 1 -21.258898
19.512512 1 6.268881 -28.167 1 -28.167 6.2688828 1 19.512508
-21.258896 z" stroke-linecap="round" />
</g></svg>
</div>
<div class='side-by-side text'>Light intensity</div>
<div class='side-by-side reading'>
<span id = "light_int"> 0 </span>
<span class='superscript'>lx</span></div>
</div>
</body></html>
)webpage";

void setup() {
Serial.begin(BAUDRATE);
WiFi.begin(ssid,pass);
Serial.print("Connecting to "); Serial.print(ssid);
Serial.print("with password "); Serial.println(pass);
Serial.print("Start connection .");
while(WiFi.status()!=WL_CONNECTED) {
delay(500);
Serial.print("."); }
Serial.print("ESP8266 board IP ");
Serial.print(WiFi.localIP());

Wire.begin();
light_sensor.begin();

//web page
server.on("/", HTTP_GET, [] (AsyncWebServerRequest *request){
request->send_P(200, "text/html", index_html);
});
```

```
server.on("/light", HTTP_GET, [] (AsyncWebServerRequest
*request) {
    request->send(200, "text/plain", String(lux));
});

server.begin();
}

long p_millis = 0;
int i=0;
String message;
#define HR_MODE_DELAY 1000 // High Resolution Mode min.120[ms]

void loop() {
if(millis() - p_millis > HR_MODE_DELAY) {
    lux = light_sensor.readLightLevel();
    message = "Light intensity" + String(lux);
    Serial.println(message + "[lx]");
    p_millis = millis();
}
}
```

Uwaga. Fragmentu kodu wyróżniono mniejszą czcionką i kursywą nie należy przepisywać. Wspomniany kod jest grafiką w formacie SVG i rozmiarach 50px x 50px.



Została ona wygenerowana przy pomocy ***svgeditoronline.com/editor/***. Do celów realizacji zadania należy przygotować swoją grafikę.

**Zad.3.** Do układu z zadania 1 podłączyć czujnik BME280. Należy napisać program, który:

- wyświetli na wyświetlaczu OLED aktualną wartość temperatury i ciśnienia atmosferycznego;
- wyświetli wartości wszystkich, możliwych do odczytania z BMP280, parametrów na stronie WWW udostępnianej przez moduł ESP8266;
- każdy parametr powinien mieć własne oznaczenie graficzne, opis oraz jednostkę widoczne na stronie.