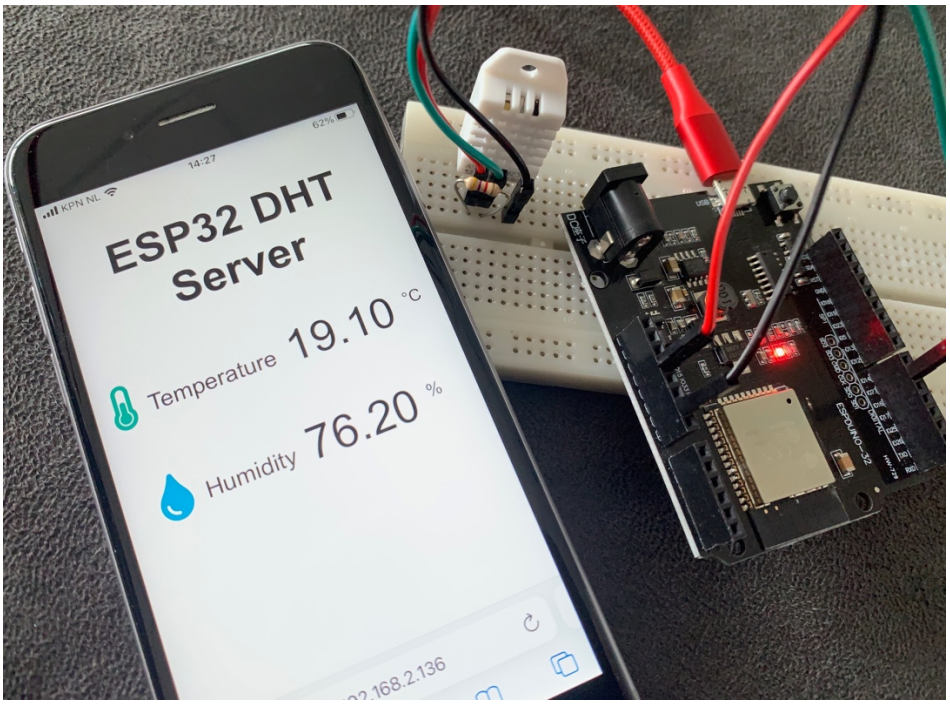


DE JONGE ONDERZOEKERS
Kijk, doe, ontdek!

Boek: ESP32-DHT-Server

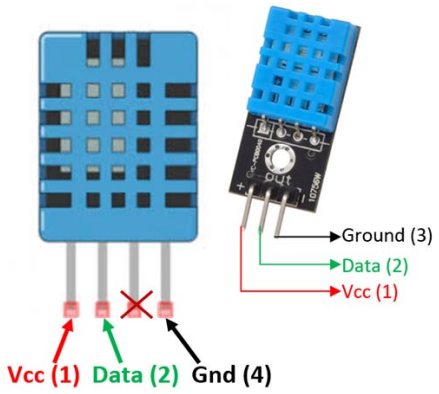
Inhoudsopgave:

1. In deze les:

In deze les bouwen we een Webserver die via HTML code de temperatuur weergeeft van een DHT sensor die zowel temperatuur als luchtvochtigheid kan meten. Tevens maken we gebruik van een andere uitvoering van het Arduino bordje, namelijk de ESP32 versie die onder andere de mogelijkheden heeft om WiFi te gebruiken en zo kan deelnemen aan het netwerk en aan internet !!

2. De DHT sensor:

De DHT sensor is er in verschillende typenummers, kleuren en uitvoeringen. Het belangrijkste verschil is dat de losse uitvoering meestal 4 pennen heeft en een extra pullup_weerstand nodig heeft. De uitvoering op een klein printje heeft 3 pennen en een eigen pullup_weerstand op het printje, dit zien we verderop weer in de bouwtekening terug. (Een pullup_weerstand zorgt er voor dat de datadraad naar 3,3V wordt gezet door de weerstand indien de datalijn “open” is).

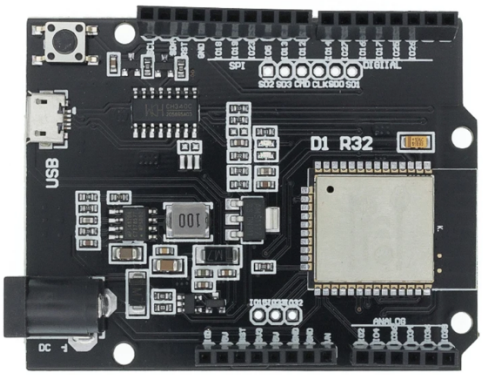


Figuur 1: De DHT sensor.

In de software geven we aan welk type DHT sensor we hebben en op welke pin van de Arduino de datapin van de sensor zit. In deze les gebruiken we pin IO27 van de ESP32.

3. Het ESP32 bord:

Het ESP32 bord is een nieuwe versie van de Arduino UNO waar naast een snellere processor ook leuke extra’s zijn ingebouwd zoals WiFi en Bluetooth. In deze les maken we gebruik van de WiFi mogelijkheden om een HTML server te bouwen die een DHT temperatuur/vochtigheid sensor uitleest en deze via WiFi naar een browser kan sturen zodat je op bijvoorbeeld een smartphone of pc de temperatuur van de sensor kan zien !!



Figuur 2: Het ESP32 bordje.

Het ESP32 bordje lijkt op de Arduino UNO met het belangrijkste verschil de 3,3Volt pinnen in plaats van 5Volt pinnen !! Daarnaast is de USB type B (de grote USB stekker van de UNO) vervangen door een USB micro stekker.

Hmmmm, klinkt goed. Wat kan ik nog meer bedenken met de ESP32 ??

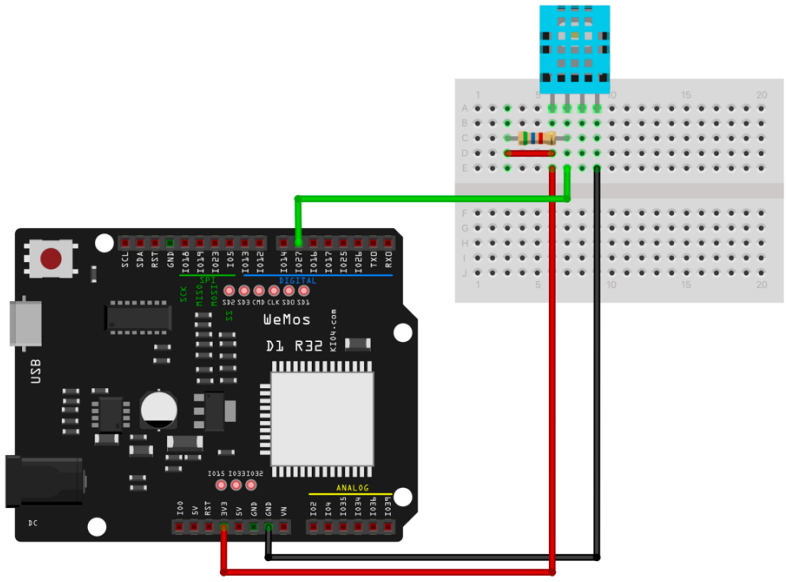
---

4. Aansluiten van de sensor op de ESP32:

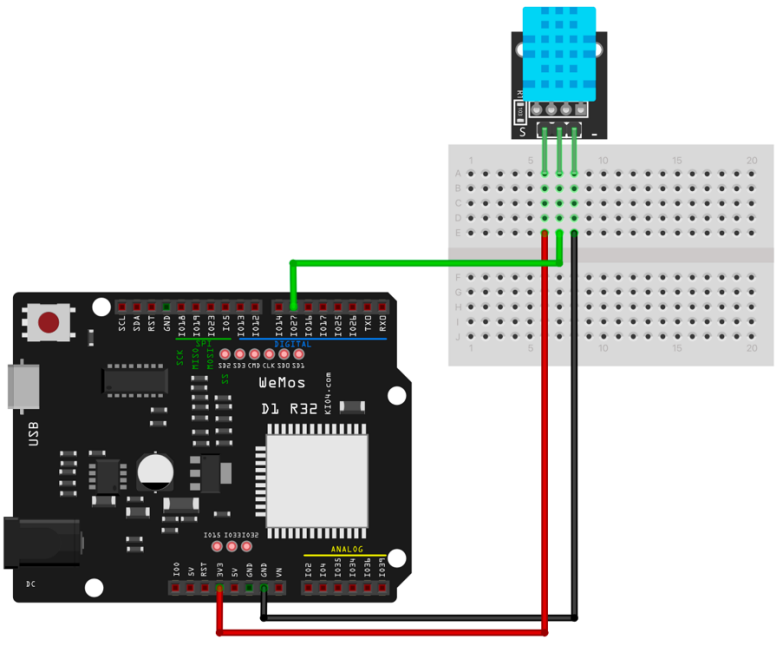
Het aansluiten van de DHT sensor kan met slechts 3 draadjes naar de Arduino ESP32. We gebruiken de GND, de 3,3V en de data naar pin **IO_27** van de ESP32.

Omdat de ESP32 op 3,3V werkt en niet op 5V zoals een Arduino UNO, sluiten we de sensor aan op de 3,3V pin van de ESP32. Hierdoor komt de data die uit de DHT sensor komt niet boven de 3,3V en gaat de ESP32 niet defect. De DHT sensor zelf werkt namelijk op zowel 3,3V als op 5Volt.

Kijk even welke sensor je gebruikt, de losse 4 pin uitvoering of de uitvoering die op een klein printje zit. Voor beide soorten is hieronder een tekening hoe je het moet aansluiten.



Figuur 3: losse DHT met 4 pennen en losse 5k6 weerstand.



Figuur 4: DHT met 3 pennen en op een printje ingebouwde weerstand.

5. Een library installeren.

Voor deze les moeten we een aantal “libraries” oftewel bibliotheken met extra commando’s en aansturing installeren.

Om een WiFi verbinding te maken komt er een heleboel code kijken. Hierdoor zou het programma vele pagina’s met commando’s bevatten waardoor het allemaal erg onoverzichtelijk wordt. Daarnaast wordt er in deze les ook nog een speciale sensor gebruikt die de nodige code nodig heeft om te werken. Voor het gebruikt van deze speciale functies zijn er in deze les een aantal libraries gebruikt waar al onze code in verstopt zit en wij een kort en overzichtelijk programma krijgen.

De volgende libraries gebruiken we:

```
#include <WiFi.h>
#include <ESPAsyncWebServer.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
```

Deze zijn te downloaden via de volgende linkjes, maar ze staan ook op onze Google Drive.

https://github.com/me-no-dev/ESPAsyncWebServer/archive/master.zip
https://github.com/adafruit/Adafruit_Sensor/archive/master.zip
https://github.com/adafruit/DHT-sensor-library/archive/master.zip
https://github.com/me-no-dev/AsyncTCP/archive/master.zip

Na het downloaden moet je de .zip files uitpakken en in de C:\Documents\Arduino\Libraries\... zetten en dan Arduino even opnieuw opstarten.

De directory ziet er als volgt uit:

C:\Documents\Arduino\Libraries\ESPAsyncWebServer
C:\Documents\Arduino\Libraries\Adafruit_Sensor
C:\Documents\Arduino\Libraries\DHT-sensor-library
C:\Documents\Arduino\Libraries\AsyncTCP

6. Arduino code voor de ESP32 DHT Server:

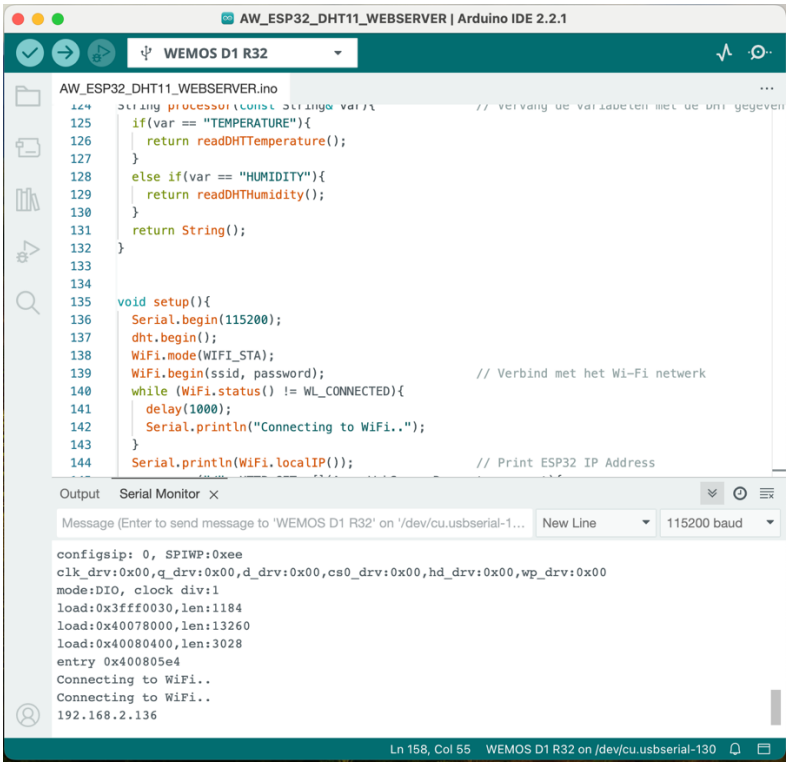
```
/*********************************************************************************
 *   ESP32 DHT sensor Webserver     (versie 16-10-2023)                           *
 *                                                                                *
 *   Add ESP32 boards:   https://dl.espressif.com/dl/package_esp32_index.json     *
 *                                                                                *
 *   Select esp32 board: WEMOS D1 R32                                             *
 *                                                                                *
 *   !!! De DHT sensor en 5k6 weerstand aansluiten op de 3,3 V en niet 5V !!!     *
 *       omdat de ESP32 op 3,3V werkt (de UNO werkt op 5V)                        *
 *                                                                                *
 *   Library downloads:                                                           *
 *   https://github.com/me-no-dev/ESPAsyncWebServer/archive/master.zip            *
 *   https://github.com/adafruit/Adafruit_Sensor/archive/master.zip               *
 *   https://github.com/adafruit/DHT-sensor-library/archive/master.zip            *
 *   https://github.com/me-no-dev/AsyncTCP/archive/master.zip                     *
 *********************************************************************************/
#include <WiFi.h>
#include <ESPAsyncWebServer.h>                    //
#include <Adafruit_Sensor.h>                      //
#include <DHT.h>                                  //

const char* ssid = "DJOG-cursisten";              // Vervang met jouw WiFi userid/password gegevens
const char* password = "DJOG1402";

//#define DHTTYPE    DHT11                        // Selecteer het model sensor: DHT 11
//#define DHTTYPE    DHT21                        // DHT 21 (AM2301)
#define DHTTYPE    DHT22                          // DHT 22 (AM2302)
#define DHTPIN 27                                 // Pin waar de DHT sensor op aangesloten is

DHT dht(DHTPIN, DHTTYPE);

AsyncWebServer server(80);                        // Maak een AsyncWebServer op poort 80 voor HTML verkeer

String readDHTTemperature(){                      // Sensor temperatuur uitlezen
  float t = dht.readTemperature();
  if (isnan(t)){                                  // Als het uitlezen mislukt exit deze loop
    Serial.println("Failed to read from DHT sensor!");
    return "--";
  }
  else{
    Serial.println(t);
    return String(t);
  }
}

String readDHTHumidity(){                         // Sensor vochtigheid uitlezen
  float h = dht.readHumidity();
  if (isnan(h)){
    Serial.println("Failed to read from DHT sensor!");
    return "--";
  }
  else{
    Serial.println(h);
    return String(h);
  }
}


const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML><html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css" integrity="sha384-
fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr" crossorigin="anonymous">
  <style>
    html {
     font-family: Arial;
     display: inline-block;
     margin: 0px auto;
     text-align: center;
    }
    h2 { font-size: 3.0rem; }
    p { font-size: 3.0rem; }
    .units { font-size: 1.2rem; }
    .dht-labels{
      font-size: 1.5rem;
      vertical-align:middle;
      padding-bottom: 15px;
    }
  </style>
</head>
```

```
<body>
  <h2>ESP32 DHT Server</h2>
  <p>
    <i class="fas fa-thermometer-half" style="color:#059e8a;"></i>
    <span class="dht-labels">Temperature</span>
    <span id="temperature">%TEMPERATURE%</span>
    <sup class="units">&deg;C</sup>
  </p>
  <p>
    <i class="fas fa-tint" style="color:#00add6;"></i>
    <span class="dht-labels">Humidity</span>
    <span id="humidity">%HUMIDITY%</span>
    <sup class="units">&percnt;</sup>
  </p>
</body>
<script>
setInterval(function ( ) {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("temperature").innerHTML = this.responseText;
    }
  };
  xhttp.open("GET", "/temperature", true);
  xhttp.send();
}, 10000 ) ;

setInterval(function ( ) {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("humidity").innerHTML = this.responseText;
    }
  };
  xhttp.open("GET", "/humidity", true);
  xhttp.send();
}, 10000 ) ;
</script>
</html>)rawliteral";

String processor(const String& var){              // Vervang de variabelen met de DHT gegevens
  if(var == "TEMPERATURE"){
    return readDHTTemperature();
  }
  else if(var == "HUMIDITY"){
    return readDHTHumidity();
  }
  return String();
}

void setup(){
  Serial.begin(115200);
  dht.begin();
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);                     // Verbind met het Wi-Fi netwerk
  while (WiFi.status() != WL_CONNECTED){
    delay(1000);
    Serial.println("Connecting to WiFi..");
  }
  Serial.println(WiFi.localIP());                 // Print ESP32 IP Address
  server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/html", index_html, processor);
  });
  server.on("/temperature", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", readDHTTemperature().c_str());
  });
  server.on("/humidity", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", readDHTHumidity().c_str());
  });
  server.begin();                                 // Start HTTP server
}


void loop(){
}
```

Probeer nu zelf eens de andere mogelijkheden van de HTML code !!

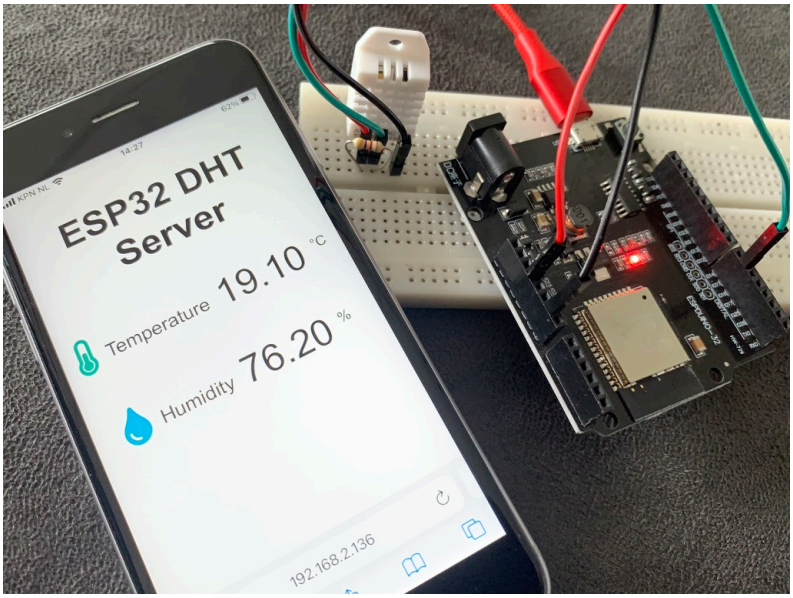
7. Verbinding maken met de DHT WebServer.

Als alles opgebouwd is en de code ge-upload is naar de ESP32 kan via de seriële monitor gekeken worden welk IP nummer je ESP32 heeft in het WiFi netwerk. (Een IP nummer is een adres waarmee een apparaat onderling kan communiceren in het netwerk. Elk apparaat heeft een eigen uniek adres)



Figuur 5: Via de seriële monitor zien we het adres 192.168.2.136.

We zien nu het IP adres 192.168.2.136 en hiermee kunnen we in een browser door dit in de zoekbalk in te typen verbinding maken met de ESP32.



Figuur 6: Via een browser en adres: 192.168.2.136 hebben we verbinding !!