**Sääkuiskaaja**

**Pöytälämpömittari**



Näyttötyö: Sulautettujen järjestelmien toteutus ja käyttöönotto

Arthur Ohjelmoija K3TVT21C

Savon ammattiopisto

3.22.2024

Sisällys

[Johdanto 2](#_Toc1502690404)

[Tarvikeluettelo 3](#_Toc1086713)

[Ohjelmointityökalujen käyttöönotto 3](#_Toc1410741887)

[Kytkentä ja johdotuskaavio 4](#_Toc1686530538)

[Laitteen johdotuskaavio 4](#_Toc933624668)

[Laitteen kytkentäkaavio 5](#_Toc661259438)

[Kuva valmiista testikytkennästä 5](#_Toc371498061)

[Ohjelmointi 6](#_Toc2072536657)

[Käytetyt kirjastot 6](#_Toc1595012695)

[Käyttöliittymä: 7](#_Toc2081715410)

[PHP PDO-yhteys: 7](#_Toc1347724708)

[Koodi DHT11 7](#_Toc28540826)

[Koodi BME280 9](#_Toc1949327069)

[Tietokanta: 12](#_Toc1139773486)

[DHT11-taulu: 12](#_Toc1249726629)

[BME280-taulu: 12](#_Toc1015799190)

[KOODIT TAULUSTA: 12](#_Toc1271599864)

[KOODIT TAULUSTA: 13](#_Toc1176054123)

[Johtopäätökset ja itsearviointi 13](#_Toc2057228137)

[Onnistumiset: 13](#_Toc1687706669)

[Ongelmat: 13](#_Toc446676642)

[Toteuttu Käyttöliittymä kuvat: 14](#_Toc53055825)

# Johdanto

Tässä työssä oli tarkoitus rakentaa, ohjelmoida ja ottaa käyttöön Pöytälämpötilamittari, joka toimii etäyhteydellä. Anturin keräämä data lähetetään HTTPS POST -menetelmällä, tallennetaan tietokantaan ja esitettään reaaliajassa lämpötilan, kosteuden ja korkeuden osalta.

Olen ottanut käyttöön erilaisia antureita tämän projektia varten, jotka mittavat eri alueilta ja tiloista lämpöä ja kosteutta.

Ohjelmoinnin osalta, oli tarkoitus kehittää monipuolista ohjelmistoa, joka kykenee vastanottaman, käsittelemään ja tallentamaan saadun data tietokantaan ja tulosta antureiden data yhdelle verkkosivulle. HTTPS yhteyden avulla saadaan turvallinen ja salattu tapa siirtää tietoa etänä.

Tietokannasta voidaan tuottaa erilaisia graafisia näkymiä tallennetuista tiedoista.

Tälle projektille valitsin BME280-anturi, se tarjoa huomattavasti kehittyneemmät ominaisuudet.

Antureiden vaihtoehdot tälle projektille ovat:

BME280-anturi pysty mittamaan lämpöä, kosteutta ja ilmapaine.

DHT11-anturi pysty mittamaan yksikertaisesti perustaso lämpö ja kosteuden.

Valintani oli BME280-anturin koska sen erinomaisen tarkkuuden ja monipuolisten ominaisuuksien vuoksi. Tavoitteenani oli kehittää monipuolista ohjelmistoa, joissa tarvitaan tarkkoja ja luotettavia mittauksia reaaliajassa.

# Tarvikeluettelo

|  |  |  |
| --- | --- | --- |
|  | kpl | Hinta (ei pakollinen) |
| Koekytkentälevy | 1 | 9,99 €,cdon.fi/kodin-elektroniikka/breadboard |
| BME280 Anturi | 1 | 14,99€,cdon.fi/kodit-ja-puutarhat/bme280 |
| ESP32 Adafruit feather | 1 | 36,92€, https://cdon.fi/kodin-elektroniikka/ |
| Micro USB Kaapeli | 1 | 8,95€, www.gigantti.fi |
| DHT11 Anturi | 1 | 5 kpl 35€, cdon.fi/kodit-ja-puutarhat/dht11 |
| Resistori | 1 | 10kΩ cdon.fi |

# Ohjelmointityökalujen käyttöönotto

Asenna Arduino IDE

* https://www.arduino.cc/en/software

Asenna CP2014 USB ajuri

* https://learn.adafruit.com/adafruit-huzzah32-esp32-feather/using-with-arduino-ide

Asenna Adafruit Huzzah with ESP32 kirjastot Arduino Ideen

* Lisää tämä rivi Arduino Iden asetuksiin: https://dl.espressif.com/dl/package\_esp32\_index.json
* (<https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json>)
* Mene sitten “Tools” > “Board” > “Boards manager” ja esti ESP32-Korttipaketti. Asenna ESP32-Korttipaketti.

Asenna BME280 kirjasto Arduino Ideen

* Valitset “install all preferences” asennuksen yhteydessä
* Valitsemalla “install all preferences” annat Arduino IDELLE luvan asentaa kaikki nämä valimiksi määritetty asetukset automaattisesti.

Asenna DHT11 kirjasto Arduino Ideen

* Valitset “install all preferences” asennuksen yhteydessä (jos sitä kysytään.)

Asenna Boards Managerissa esp32 korttipaketti Arduino Ideen

Valitse Arduino Idessä kortti Adafruit ESP32 Feather

Asenna Wamp palvelimen (WAMP (Windows, Apache, MySQL, PHP/Perl/Python) Tietokoneellesi

Tässä projektissa käytin wampserver palvelimen.

* Menee osoitteeseen <https://wampserver.aviatechno.net/> ja lataa Wampserver
* Muista noudata asennusohjeita ja asenna se laitteellesi.

Tarkista kirjastojen saatavuus

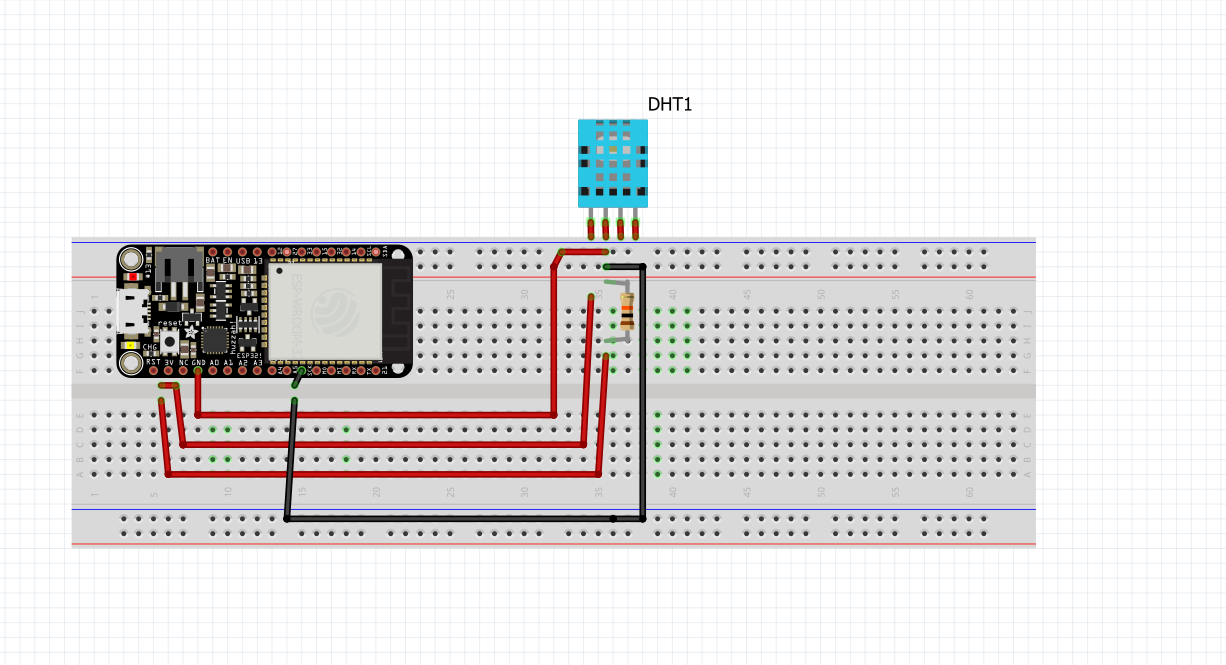
* Kun Arduino IDE on avattu, varmistaa, että tarvittavat kirjastot ovat käytettävissä. voidaan kirjastojen asennuksen varmistella tästä Siirry valikkoon "Sketch" > "Include Library" > "Manage Libraries..."

# Kytkentä ja johdotuskaavio

Johdotuskaavio tehty Frizing ohjelmalla. Lisäkomponentteja joutui lataamaan valmistajan sivuilta ja tuoda ne manuallisesti Frizingiin. Esimerkiksi ESP32 –kuvaa ei löydy suoraan Fritzingistä. Samoin DHT11-sensorin kuvaa ei löytynyt suora Fritzingistä mutta bme280 –anturin kuva oli saatavilla Fritzing ohjelmassa.

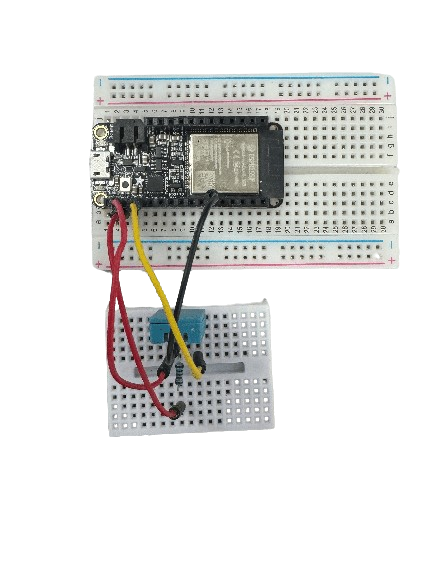
Kytkentäkaavio on tehty Frizing ohjelmalla myös samalla periaatteella.

# Laitteen johdotuskaavio



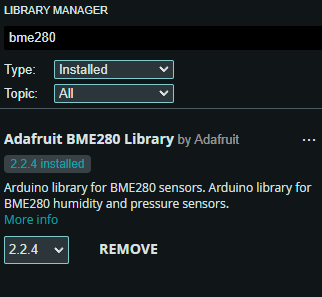
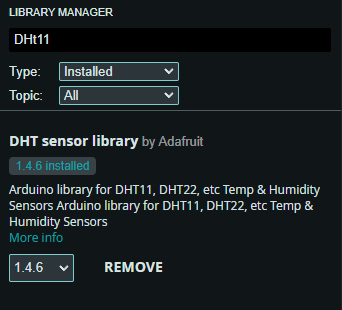
# Laitteen kytkentäkaavio

# Kuva valmiista testikytkennästä



# Ohjelmointi

### Käytetyt kirjastot



### Käyttöliittymä:

Käyttöliittymä luomisessa käytin MDB Bootstrap että sen avulla saadaaan mobiilistävällinen verkkosivu. Sitä voidaan muokata tarvittaessa helposti.

* MDB Bootstrap on Bootstrap-kehykseen perustuva, responsiivinen ja moderni käyttöliittymäkehys.
* Käytämme sitä verkkosivun luomisessa, jotta saamme mobiiliystävällisen käyttöliittymän.
* MDB Bootstrap tarjoaa valmiita tyylejä ja komponentteja, joiden avulla voimme helposti suunnitella ja toteuttaa käyttöliittymän.
* Käyttöliittymää voidaan helposti laajentaa lisäämällä uusia antureita tarvittaessa, koska MDB Bootstrap tarjoaa joustavan ja skaalautuvan rakenteen.

### PHP PDO-yhteys:

* Käytämme PHP:n PDO (PHP Data Objects) -luokkaa tietokantayhteyden luomiseen ja hallintaan. PDO auttaa suojaamaan sovellustamme tietoturvaongelmilta, kuten SQL-injektioilta.
* PDO mahdollistaa myös helpomman siirtymisen erilaisten tietokantojen välillä

### Koodi DHT11

#include <WiFi.h>

#include <HTTPClient.h>

#include "DHT.h"

const char\* ssid = "TVT-WLAN"; // Enter your WiFi SSID here

const char\* password = "salasana"; // Enter your WiFi password here

// DHT Sensor

#define DHTPIN 5

#define DHTTYPE DHT11 // Change this if you're using a different DHT sensor

DHT dht(DHTPIN, DHTTYPE);

const char\* serverIP = "10.124.12.95"; // Use the IP of your WAMP server

const char\* endpoint = "/Weather-app-IOT/insert\_data\_dht11.php"; // Endpoint for the PHP script

#define LED\_PIN 13 // LED pin

void setup() {

Serial.begin(115200);

pinMode(LED\_PIN, OUTPUT); // Initialize LED pin

pinMode(DHTPIN, INPUT);

dht.begin();

Serial.println("Connecting to ");

Serial.println(ssid);

// Connect to Wi-Fi

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected..!");

Serial.print("Got IP: ");

Serial.println(WiFi.localIP());

}

void loop() {

float temperature = dht.readTemperature();

float humidity = dht.readHumidity();

if (isnan(temperature) || isnan(humidity)) {

Serial.println("Failed to read from DHT sensor!");

return;

}

// Construct the data string

String data = "temperature=" + String(temperature) + "&humidity=" + String(humidity);

Serial.print("Sending data: ");

Serial.println(data);

// Send HTTP POST request to WAMP server

HTTPClient http;

http.begin("http://" + String(serverIP) + String(endpoint));

http.addHeader("Content-Type", "application/x-www-form-urlencoded");

int httpResponseCode = http.POST(data);

if (httpResponseCode > 0) {

Serial.print("HTTP Response code: ");

Serial.println(httpResponseCode);

} else {

Serial.print("Error in HTTP request: ");

Serial.println(httpResponseCode);

}

http.end();

delay(10000); // Wait for 1 second before sending the next request

}

### Koodi BME280

#include <WiFi.h>

#include <HTTPClient.h>

#include <Adafruit\_Sensor.h>

#include <Adafruit\_BME280.h>

#include <Wire.h>

const char\* ssid = "TVT-WLAN"; // Enter your WiFi SSID here

const char\* password = "salasana"; // Enter your WiFi password here

// BME280 Sensor

#define BME\_SDA 23

#define BME\_SCL 22

#define SEALEVELPRESSURE\_HPA (1013.25) // Adjust this to your local sea level pressure

Adafruit\_BME280 bme;

const char\* serverIP = "10.124.12.95"; // Use the IP of your WAMP server

const char\* endpoint = "/Weather-app-IOT/insert\_data\_bme280.php"; // Endpoint for the PHP script

#define LED\_PIN 13 // LED pin

void setup() {

Serial.begin(115200);

pinMode(LED\_PIN, OUTPUT); // Initialize LED pin

Wire.begin(BME\_SDA, BME\_SCL);

if (!bme.begin()) {

Serial.println("Could not find BME280 sensor, check wiring!");

while (1);

}

Serial.println("Connecting to ");

Serial.println(ssid);

// Connect to Wi-Fi

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

digitalWrite(LED\_PIN, LOW); // Turn off LED if not connected

delay(500);

digitalWrite(LED\_PIN, HIGH); // Turn on LED if connected

delay(500);

Serial.print(".");

}

digitalWrite(LED\_PIN, HIGH); // Turn off LED if connected

Serial.println("");

Serial.println("WiFi connected..!");

Serial.print("Got IP: ");

Serial.println(WiFi.localIP());

}

void loop() {

float temperature = bme.readTemperature();

float humidity = bme.readHumidity();

float pressure = bme.readPressure() / 100.0F; // Pressure in hPa

float altitude = bme.readAltitude(SEALEVELPRESSURE\_HPA);

Serial.print("Temperature = ");

Serial.print(temperature);

Serial.println(" \*C");

Serial.print("Humidity = ");

Serial.print(humidity);

Serial.println(" %");

Serial.print("Pressure = ");

Serial.print(pressure);

Serial.println(" hPa");

Serial.print("Altitude = ");

Serial.print(altitude);

Serial.println(" meters");

// Construct the data string

String data = "temperature=" + String(temperature) + "&humidity=" + String(humidity) + "&pressure=" + String(pressure) + "&altitude=" + String(altitude);

Serial.print("Sending data: ");

Serial.println(data);

// Send HTTP POST request to WAMP server

HTTPClient http;

http.begin("http://" + String(serverIP) + String(endpoint));

http.addHeader("Content-Type", "application/x-www-form-urlencoded");

int httpResponseCode = http.POST(data);

if (httpResponseCode > 0) {

Serial.print("HTTP Response code: ");

Serial.println(httpResponseCode);

digitalWrite(LED\_PIN, HIGH); // Turn on LED

delay(500); // Delay for LED blinking

digitalWrite(LED\_PIN, LOW); // Turn off LED

} else {

Serial.print("Error in HTTP request: ");

Serial.println(httpResponseCode);

}

if (httpResponseCode == 200) {

String responseBody = http.getString();

Serial.println("Received response:");

Serial.println(responseBody);

}

http.end();

delay(10000); // Wait for 10 seconds before sending the next request

}

**\*\*\*\*\*Muista käydä muokkaamassa Wampserverin asetuksista, “httpd-vhosts “tiedostosta, jotta data siirto verkkosivulle onnistuu. Tämä on tärkeää, ilman muokkauksia dataa ei voida siirtää verkkosivulle.\*\*\*\***

**Polkku : C:\wamp64\bin\apache\apache2.4.54.2\conf\extra\ httpd-vhosts”**

# Tietokanta:

### DHT11-taulu:

-------------------------------------------------

| ID (PK) | lampo | kosteus | pvm (timestamp) |

-------------------------------------------------

BME280-taulu: (taulu on “sensori” nimellä nimitetty)

---------------------------------------------------------------------------------

| ID (PK) | nimi | lampo (decimal) | kosteus (decimal) | paine (decimal) | korkeus (decimal) | pvm (timestamp) |

---------------------------------------------------------------------------------

**DHT11-taulu:**

**ID:** Ensimmäisen taulun pääavain, joka on automaattisesti kasvava kokonaisluku.

**lampo:** Kokonaisluku, joka tallentaa lämpötilan lukeman.

**kosteus:** Kokonaisluku, joka tallentaa kosteuden lukeman.

**pvm:** Aikaleima, joka tallentaa tietueen lisäysajan.

**BME280-taulu:**

**ID:** Toinen taulun pääavain, joka on automaattisesti kasvava kokonaisluku.

**nimi:** Merkkijono, joka tallentaa anturin nimen tai tunnisteen.

**lampo:** Desimaaliluku, joka tallentaa lämpötilan lukeman.

**kosteus:** Desimaaliluku, joka tallentaa kosteuden lukeman.

**paine:** Desimaaliluku, joka tallentaa ilmanpaineen lukeman.

**korkeus:** Desimaaliluku, joka tallentaa korkeuden lukeman.

**pvm:** Aikaleima, joka tallentaa tietueen lisäysajan.

### KOODIT TAULUSTA:

**(bme280)**

Tietokannan muuttujat voi olla eri, kun koodissa, tarkista huolellisesti ennen koodin suorittaminen. Voit myös myöhemmin muokata tietokanta,

DROP TABLE IF EXISTS `sensordata`;

CREATE TABLE IF NOT EXISTS `sensordata` (

`ID` int NOT NULL AUTO\_INCREMENT,

`nimi` varchar(50) NOT NULL,

`lampo` decimal(5,2) NOT NULL,

`kosteus` decimal(5,2) NOT NULL,

`paine` decimal(7,2) NOT NULL,

`korkeus` decimal(7,2) NOT NULL,

`pvm` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (`ID`)

) ENGINE=MyISAM AUTO\_INCREMENT=31024 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

### KOODIT TAULUSTA:

**(DHT11)**

DROP TABLE IF EXISTS `sensor`; CREATE TABLE IF NOT EXISTS `sensor` (

`ID` int NOT NULL AUTO\_INCREMENT,

`lampo` int NOT NULL, `kosteus` int NOT NULL,

`pvm` timestamp NOT NULL,

PRIMARY KEY (`ID`) )

ENGINE=MyISAM AUTO\_INCREMENT=2020 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

# Johtopäätökset ja itsearviointi

### Onnistumiset:

Projekti alkuperäiset tavoitteet savutettiin hyvin ja aikataulu pysyi hyvänä.

Helpoin oli käyttöliittymän luominen, minulla oli etukäteen idea, että miltä se lopputulos näyttää.

Käyttöliittymä toteuttaminen oli helppo koska minulla on MDB Bootstrap aikaisempia kokemuksia. Projektin aikana tajusin hyvin käytäntöjä, että miten voin lisätä monta antureita yhdelle verkkosivulle näkyväksi.

### Ongelmat:

Kasaus oli varsin hankalaa, koska irtonaisia epäonnistuneita 3d tulostuksia.

Kasauksen vaikeudet johtuivat epäonnistuneista 3d-tulosteista.

Epäonnistuneita paloista kootun kotelon kokoonpanoa helpottivat kuumaliiman käyttö.

Http yhteyden kanssa oli haasteita, muut teknologiat olivat tuttuja jollain tavalla, joten oli helppo, lähtee tekemään.

Projekti kokonaisuudessaan oli opettava kokemus siitä, miten Arduino IDE:stä kerättyä dataa voidaan siirtää WAMP-palvelimelle ja miten tämän datan voi etäyhteyden kautta olla näkyvillä missä vaan.

# Toteuttu Käyttöliittymä kuvat:

|  |
| --- |
| LG    MD    MD    SM |
| XS |