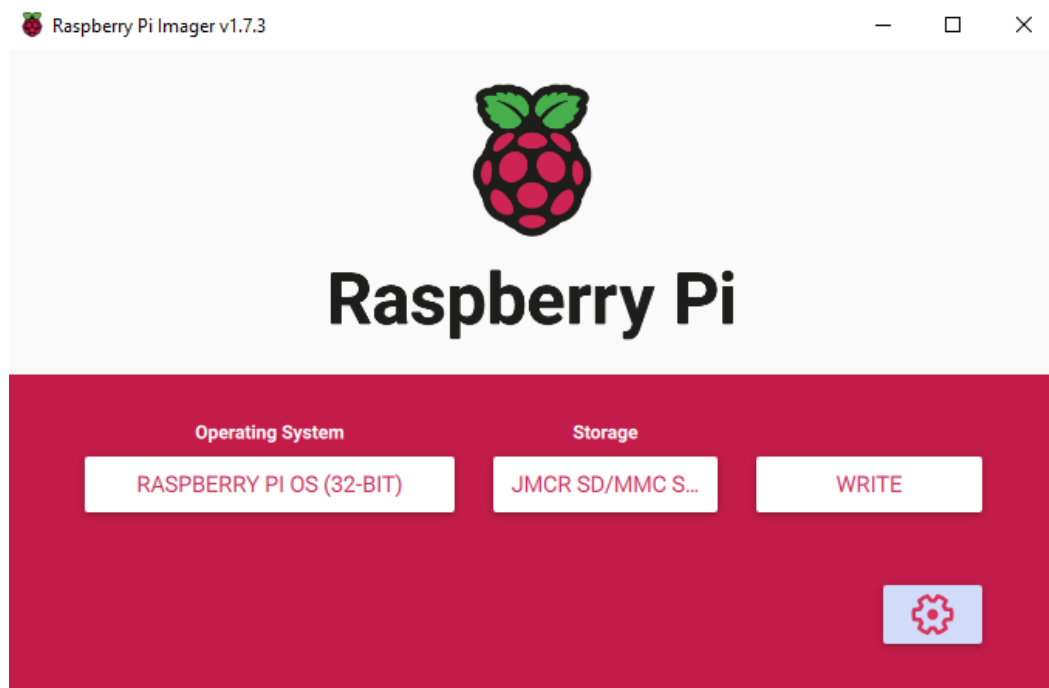


## Raspberry Pi/ESP32-kamera

Projektissa käytetyt tekniikat

- Raspberry Pi 3 Model B 1.2
- ESP32
- DHT11 lämpötila/kosteusanturi
- sarjaliikenne(UART), vaihtui projektin aikana Wifiin
- Raspberry Pi Camera
- ohjelmointikielet: python, c++, html, css, javascript, php
- MQTT (Message Queuing Telemetry Transport)
- pilvipalvelu: Arduino Cloud

Aloitin asentamalla Rasperryn käyttöjärjestelmän uudelleen (kuva yksi) sd-kortille.



Kuva yksi, Raspberry Imager-käyttöjärjestelmän asennusohjelma.

Tämän jälkeen otin yhteyden Raspiin VNC:llä ja ensimmäiseksi suoritin PINOUT-komennon (kuvat kaksi ja kolme) kytkentöjä varten.

```
pi@pi: ~  
File Edit Tabs Help  
pi@pi:~ $ pinout  
-----  
00000000000000000000 J8  
10000000000000000000  
Pi Model 3B V1.2  
+-----+  
[D] [SoC]  
[S] [I]  
[I]  
+-----+  
pwr HDMI [C] [S] [I] [A]  
[V]  
+-----+  
USB  
USB  
Net  
-----  
Revision : a02082  
SoC : BCM2837  
RAM : 1GB  
Storage : MicroSD  
USB ports : 4 (of which 0 USB3)  
Ethernet ports : 1 (100Mbps max. speed)  
Wi-fi : True  
Bluetooth : True  
Camera ports (CSI) : 1
```

Kuva kaksi, PINOUT-komento kytkentöjä varten.

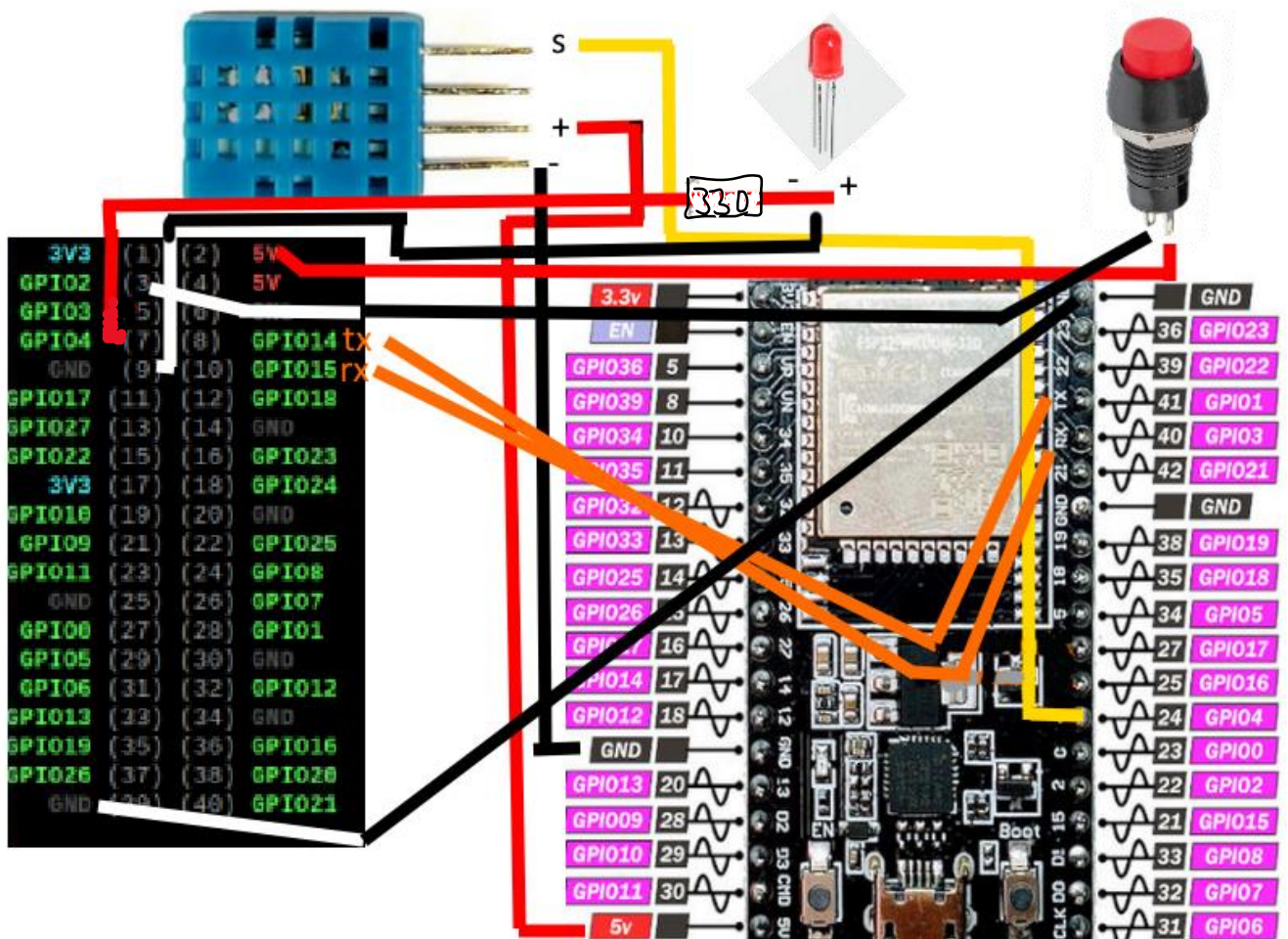
```
pi@pi: ~  
File Edit Tabs Help  
J8:  
3V3 (1) (2) 5V  
GPIO2 (3) (4) 5V  
GPIO3 (5) (6) GND  
GPIO4 (7) (8) GPIO14  
GND (9) (10) GPIO15  
GPIO17 (11) (12) GPIO18  
GPIO27 (13) (14) GND  
GPIO22 (15) (16) GPIO23  
3V3 (17) (18) GPIO24  
GPIO18 (19) (20) GND  
GPIO9 (21) (22) GPIO25  
GPIO11 (23) (24) GPIO8  
GND (25) (26) GPIO7  
GPIO8 (27) (28) GPIO1  
GPIO5 (29) (30) GND  
GPIO6 (31) (32) GPIO12  
GPIO13 (33) (34) GND  
GPIO19 (35) (36) GPIO16  
GPIO26 (37) (38) GPIO20  
GND (39) (40) GPIO21  
For further information, please refer to https://pinout.xyz/  
pi@pi:~ $
```

Kuva kolme, PINOUT-komento kytkentöjä varten.

Kameran ollessa kyseessä olisi hienoa että kuvaajan liikkuminen ei olisi rajoitettu virtajohdon pituuteen mutta en löytänyt tarvikkeistani Raspin tarpeisiin sopivaa virtalähdettä. Raspi saattaa ottaa jopa 2.5 ampeeria virtaa (<https://thepihut.com/blogs/raspberry-pi-tutorials/how-do-i-power-my-raspberry-pi>) joten päädyin käyttämään verkkovirtaa.

Esp32:n virrankulutus on vähemmän ([https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/?utm\\_content=cmp-true](https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/?utm_content=cmp-true)) joten siihen olisi voinut käyttää 1000mA antavaa usb-powerbankia. Päädyin kuitenkin verkkovirtaan.

Jatkoin tekemällä kytkennät, kuva neljä.



Kuva neljä, kytkennät. Huomaa oranssit vedot sarjaporttiliikennettä varten jota tarkoitus käyttää dht-datan siirtämiseen esp:stä Raspiin. Tästä kuitenkin jouduin siirtymään wifin käyttöön sillä sarjaliikennettä käyttäessä data ei jostakin syystä saapunut ehjänä perille. Projektin edetessä käytettävät pinnit muuttuivat hieman mutta periaate pysyi suunnilleen samana.

Seuraavaksi tein Arduino IDE:ssä alla olevan koodin DHT-anturille jonka latasin ESP:en.

```
#include <DHT.h>
#include <HardwareSerial.h>
```

```
#define DHTPIN 4
#define DHTTYPE DHT11
#define TXPIN 17
#define RXPIN 16
```

```
float temperature;
```

```

float humidity;
DHT dht(DHTPIN,DHTTYPE);
HardwareSerial SerialPort(2);

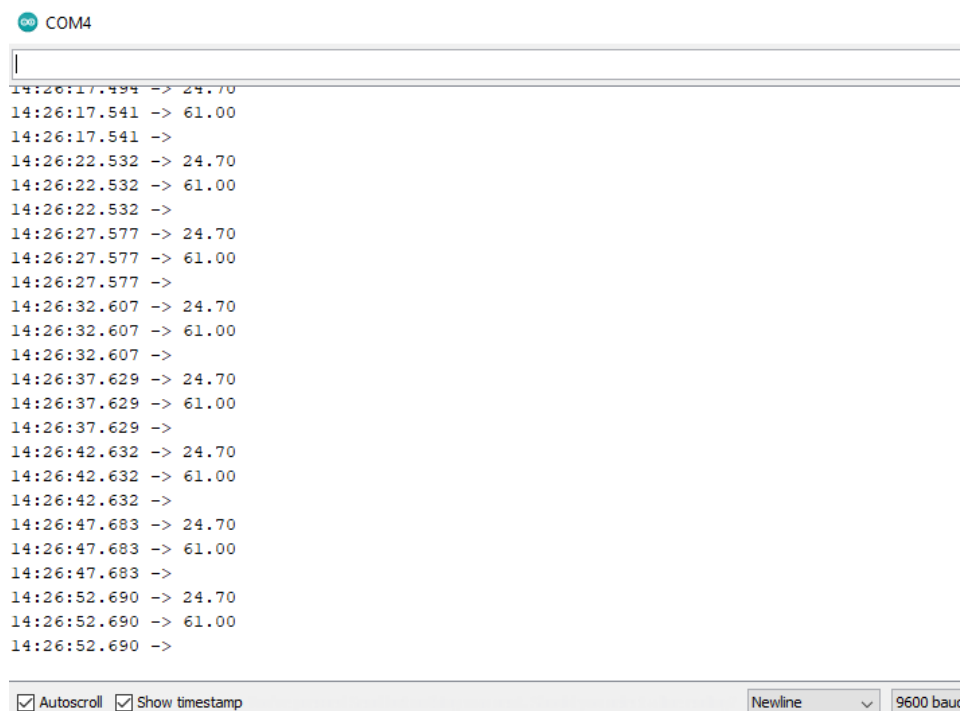
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  SerialPort.begin(9600,SERIAL_8N1,RXPIN,TXPIN);
  dht.begin();
}

void loop() {

  temperature = dht.readTemperature();
  humidity = dht.readHumidity();
  Serial.println(temperature);
  Serial.println(humidity);
  SerialPort.println(temperature);
  SerialPort.println(humidity);
  Serial.println();
  delay(5000);
}

```

Tämän jälkeen tarkistin dht:n toiminnan ja lämpötila ja kosteusprosentti saapuivat sarjamonitoriin(kuva viisi).



The screenshot shows the Arduino IDE serial monitor window. At the top, it says 'COM4'. The main area displays a series of data points, each consisting of a timestamp, a temperature value, and a humidity value, separated by a right-pointing arrow. The data is as follows:

Timestamp	Temperature (°C)	Humidity (%)
14:26:17.494	24.70	61.00
14:26:17.541	24.70	61.00
14:26:22.532	24.70	61.00
14:26:22.532	24.70	61.00
14:26:27.577	24.70	61.00
14:26:27.577	24.70	61.00
14:26:32.607	24.70	61.00
14:26:32.607	24.70	61.00
14:26:37.629	24.70	61.00
14:26:37.629	24.70	61.00
14:26:42.632	24.70	61.00
14:26:42.632	24.70	61.00
14:26:47.683	24.70	61.00
14:26:47.683	24.70	61.00
14:26:52.690	24.70	61.00
14:26:52.690	24.70	61.00

At the bottom of the window, there are checkboxes for 'Autoscroll' and 'Show timestamp', both of which are checked. To the right, there is a dropdown menu for 'Newline' and a text box for '9600 baud'.

Kuva viisi, dht:n mittaamia lukemia Arduino IDE:n sarjamonitorissa.

Sitten ajoin raspissa komennot:

```
sudo apt update
```

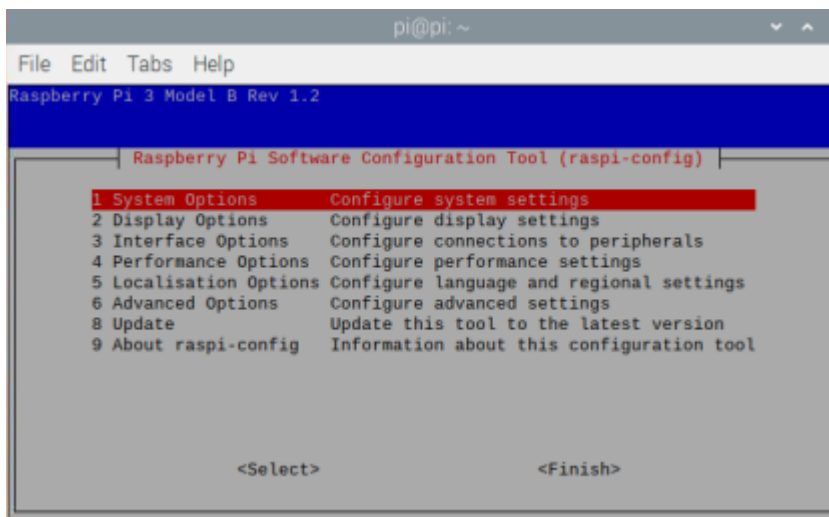
```
sudo apt upgrade
```

jotta sain raspin paketinhallinnan ajantasalle.

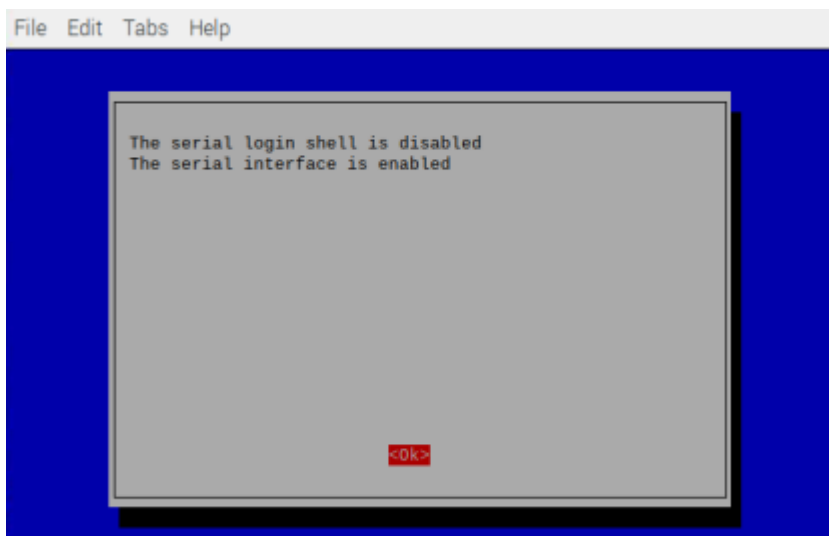
Sitten aktivoidakseni sarjaliikenteen ajoin komennon

```
sudo raspi-config
```

jolla pääsee käsiksi raspin konfiguraatioihin(kuva kuusi). Kohdasta kolme "interface options" pääsee aktivoimaan sarjaporttiliikenteen(kuva seitsemän). Samalla aktivoin mahdollisuuden kameran käyttöön joka myös löytyy kohdan "interface options"-alta.

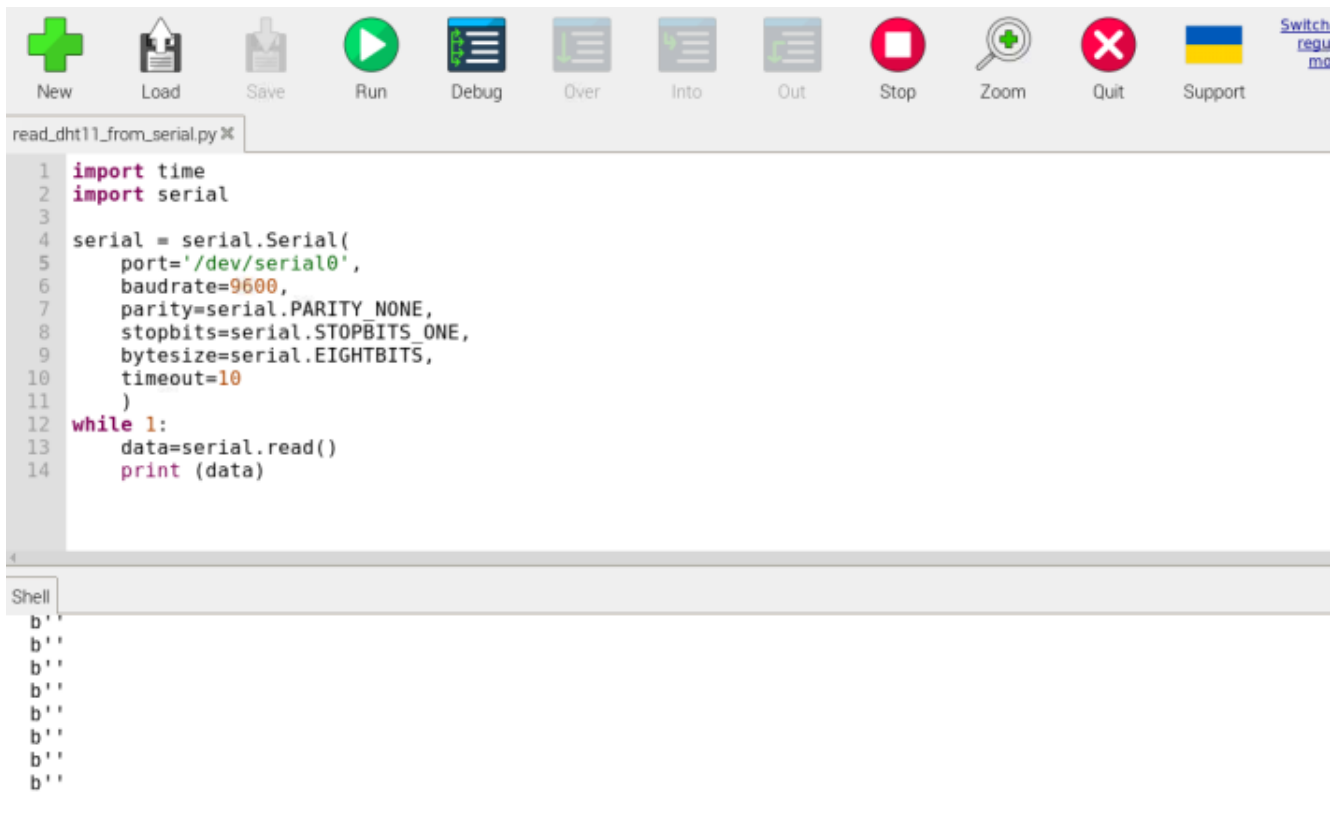


Kuva kuusi, raspin konfiguraatioita.



Kuva seitsemän, raspin sarjaporttiliikenne aktivoitu.

Tein raspiin python-koodin mutta ensiyrityksellä se ei toiminut vaan perille saapui viesti b''(kuva kahdeksan).



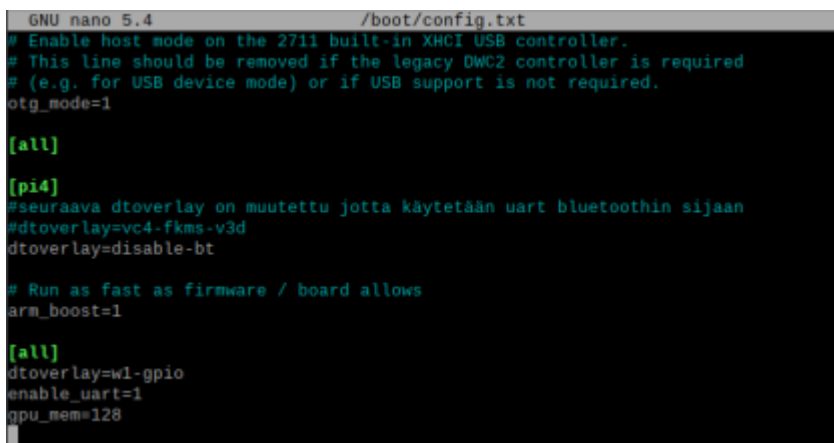
The screenshot shows a Python IDE with a toolbar at the top containing icons for New, Load, Save, Run, Debug, Over, Into, Out, Stop, Zoom, Quit, and Support. The main window displays a Python script named `read_dht11_from_serial.py` with the following code:

```
1 import time
2 import serial
3
4 serial = serial.Serial(
5     port='/dev/serial0',
6     baudrate=9600,
7     parity=serial.PARITY_NONE,
8     stopbits=serial.STOPBITS_ONE,
9     bytesize=serial.EIGHTBITS,
10    timeout=10
11)
12 while 1:
13     data=serial.read()
14     print (data)
```

Below the script, a shell window shows the output of the program, which consists of multiple lines of `b''`.

Kuva kahdeksan, raspin python-koodi ja perille saapuu vain b''.

Googlettelen jälkeen muutin `/boot/config.txt`-tiedostoa(kuva yhdeksän). Ajoin myös komennon `sudo systemctl disable hciuart` joka alustaa bluetooth-modeemin. Nämä muutokset sen vuoksi että uart on Raspi 3:ssa oletuksena valjastettu bluetoothin käyttöön.



The screenshot shows the `/boot/config.txt` file being edited in the GNU nano 5.4 editor. The file contains the following configuration:

```
GNU nano 5.4 /boot/config.txt
# Enable host mode on the 2711 built-in XHCI USB controller.
# This line should be removed if the legacy DWC2 controller is required
# (e.g. for USB device mode) or if USB support is not required.
otg_mode=1

[all]

[pi4]
#seuraava dtoverlay on muutettu jotta käytetään uart bluetoothin sijaan
#dtoverlay=vc4-fkms-v3d
dtoverlay=disable-bt

# Run as fast as firmware / board allows
arm_boost=1

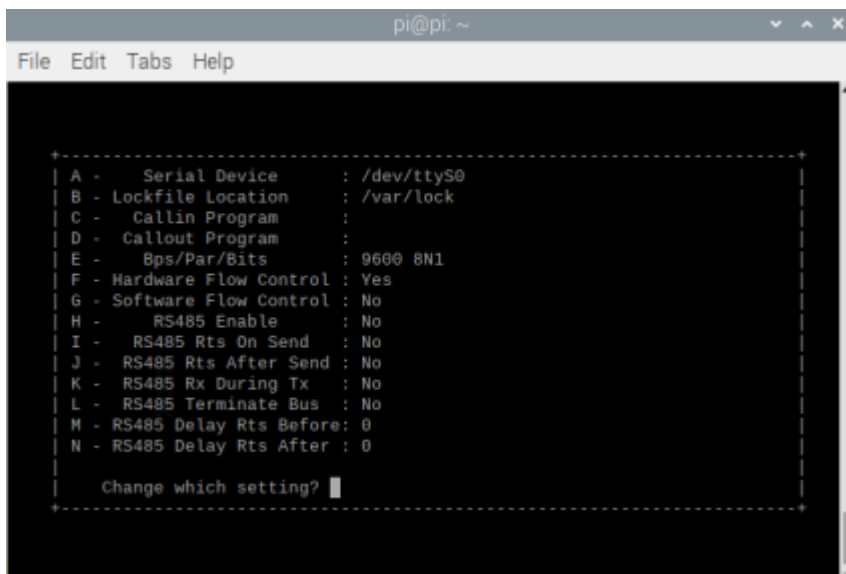
[all]
dtoverlay=w1-gpio
enable_uart=1
gpu_mem=128
```

Kuva yhdeksän, muutettu `/boot/config.txt`.

Kaikesta huolimatta data ei kulkenut perille asti. Tässä vaiheessa komennolla

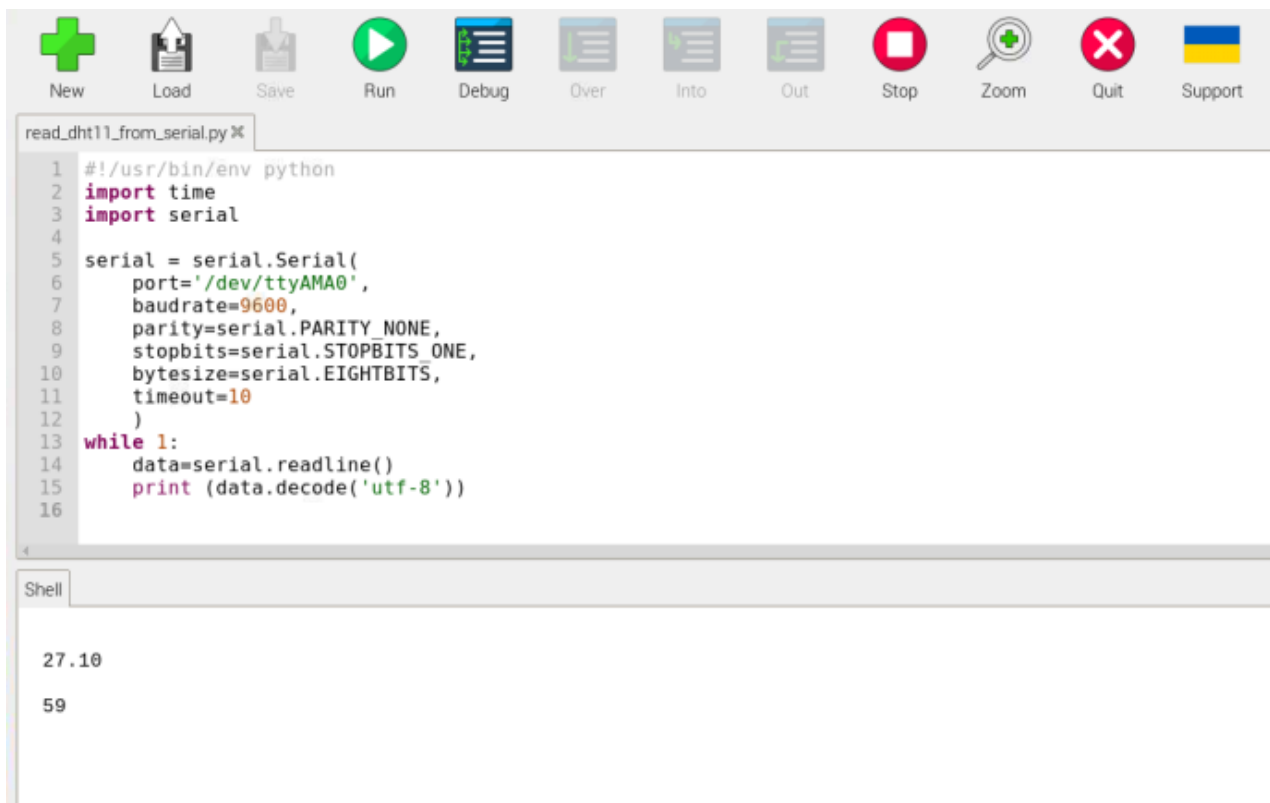
`sudo apt install minicom`

asensin minicomin jolla voi hallita sarjaporttiliikenteen asetuksia (kuva kymmenen).



Kuva kymmenen, minicom-ohjelman näkymää.

Pitkällisen säätämisen jälkeen data alkoi saapua perille(kuva 11).



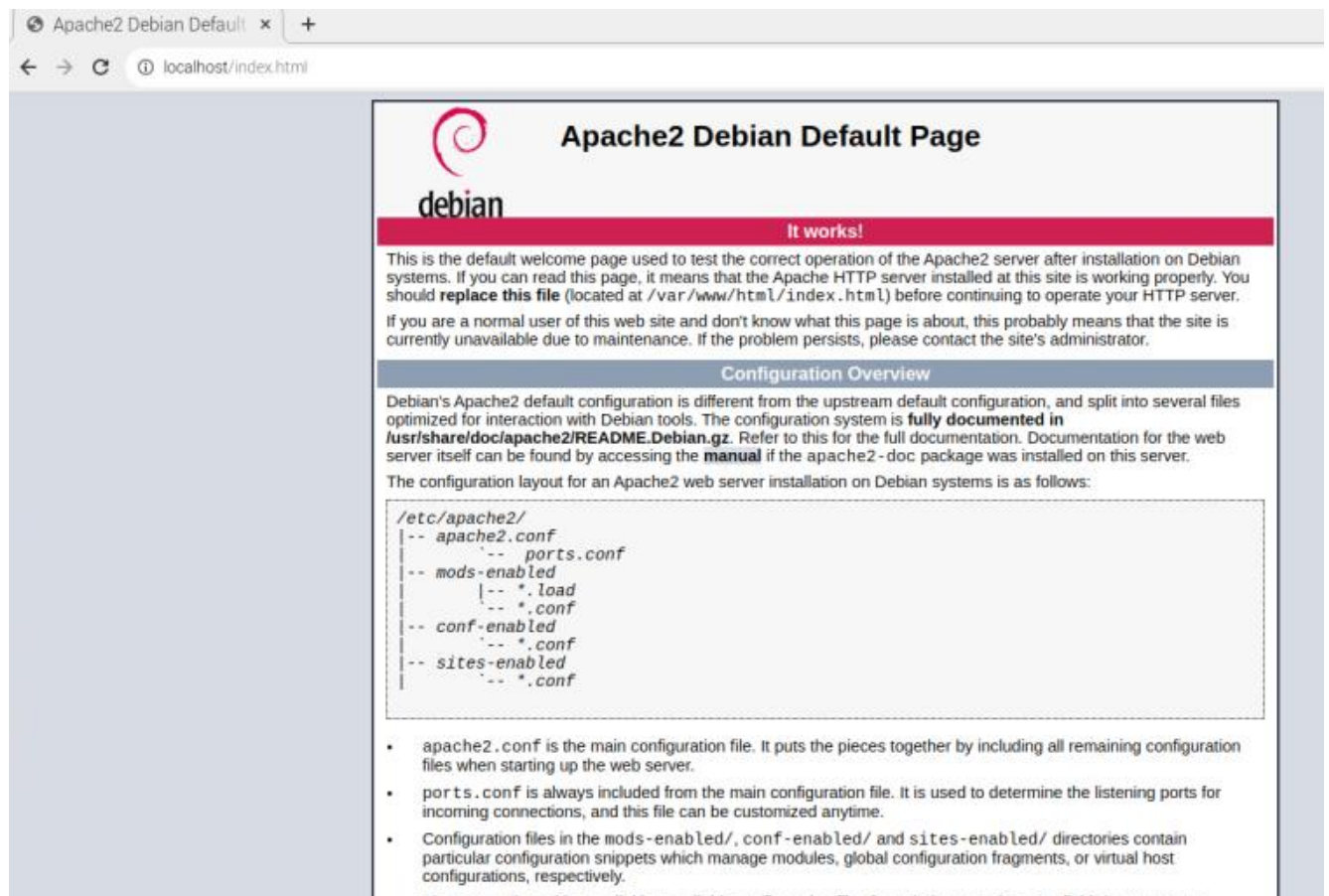
Kuva 11, mitatut arvot tulevat perille raspiin. Kuvassa harvinainen tilanne, data saapuu kokonaisuudessaan. Myöhemmin nimittäin alkoivat ongelmat sillä datasta saapui vain osia.

Ajatuksena on esittää kameran ottama kuva ja anturin mittaama data omalla web-sivulla joten seuraavaksi on serverin asennusvuoro. Ensin itse apache-serveri komennolla



sudo apt install apache2 -y

Kokeillaan josko serveri toimisi (kuva 12).



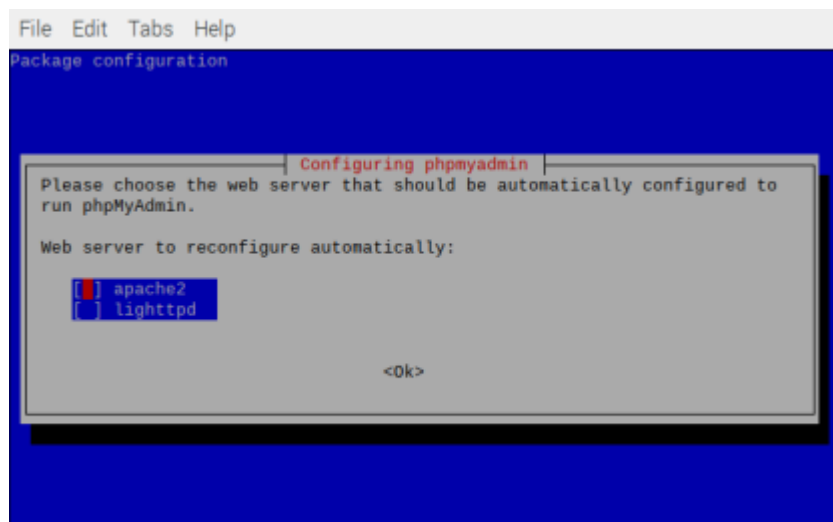
Kuva 12, Apache-serverin oletussivu.

Komennoilla

sudo apt install mariadb-server php-mysql -y

sudo apt install phpmyadmin -y

asensin Mariadb-tietokannan ja phpmyadmin-tietokannan hallintatyökalun(kuva 13).



Kuva 13, phpmyadmin-asennus.



Näiden jälkeen serverin uudelleen käynnistys komennolla

`sudo service apache2 restart`

Seuraavaksi kokeilin pääsenkö phpmyadminiin, ei onnistunut sillä phpmyadmin on väärässä kansiossa. Ongelman korjaa linkki komennolla

`sudo ln -s /usr/share/phpmyadmin /var/www/html/phpmyadmin`

Phpmyadmin ei näyttänyt aivan sille kuin pitäisi. Asennuksesta jäi pois php-kieli ja jos sivu on tehty php:llä niin ulkoasu on karuhko (kuva 14).



```
<?php
/* vim: set expandtab sw=4 ts=4 sts=4: */
/**
 * Main loader script
 *
 * @package PhpMyAdmin
 */
declare(strict_types=1);

use PhpMyAdmin\Controllers\HomeController;
use PhpMyAdmin\Core;
use PhpMyAdmin\DatabaseInterface;
use PhpMyAdmin\Response;
use PhpMyAdmin\Url;
use PhpMyAdmin\Util;

if (! defined('ROOT_PATH')) {
    define('ROOT_PATH', __DIR__ . DIRECTORY_SEPARATOR);
}

global $server;

require_once ROOT_PATH . 'libraries/common.inc.php';

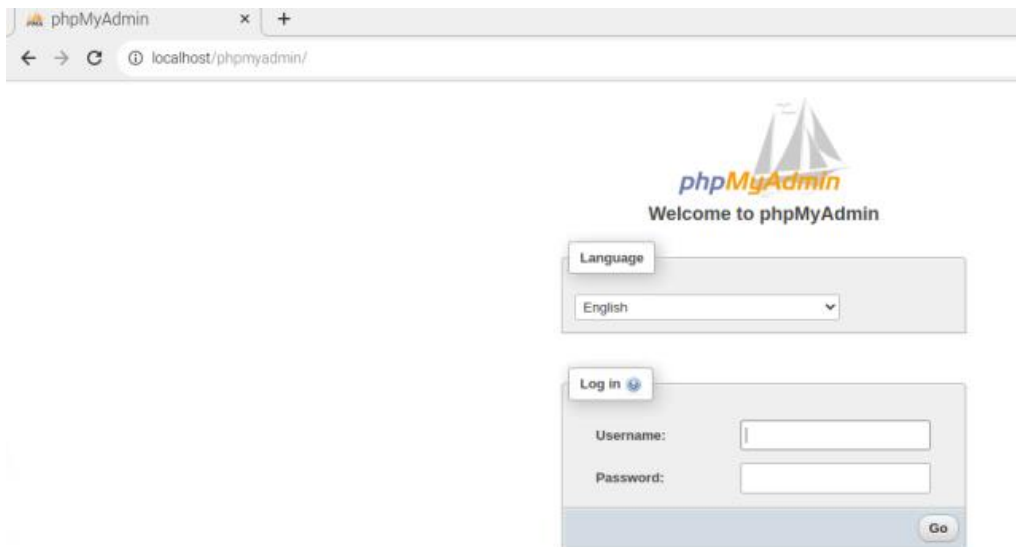
/**
 * pass variables to child pages
 */
$drops = [
    'lang',
    'server',
    'collation_connection',
    'db',
    'table',
];
foreach ($drops as $each_drop) {
    if (array_key_exists($each_drop, $_GET)) {
        unset($_GET[$each_drop]);
    }
}
unset($drops, $each_drop);
```

Kuva 14, phpmyadmin-sivu näytettynä ilman asennettua php-kieltä.

Asensin php-kielen komennolla

`sudo apt install php -y`

jonka jälkeen phpMyAdmin näyttää sille kuin pitäisikin (kuva 15).



Kuva 15, phpmyadmin näytettynä oikein.

Komennoilla

```
sudo mysql --user=root --password
```

```
> create user admin@localhost identified by 'your_password';
```

```
> grant all privileges on *.* to admin@localhost;
```

```
> FLUSH PRIVILEGES;
```

tein uuden käyttäjän jolla on pääsy phpmyadminiin, Serverin asennuksen lopuksi ajoin vielä komennon

```
sudo systemctl enable apache2
```

jotta serveri käynnistyy samalla kun raspi käynnistyy.

Seuraavaksi mqtt-asennus, ajatuksena että esp:n data tulee raspiin, raspin python-koodissa mqtt-client joka julkaisee datan ja serverin mqtt-client vastaanottaa datan.

Mqtt-asennus komennolla

```
sudo apt install mosquitto mosquitto-clients
```

Python-koodia varten myös mqtt komennolla

```
sudo pip3 install paho-mqtt
```

Lisätään tarvittavat muutokset mqtt:n konffauksiin (kuva 16). Huomaa websockets-osio jota tarvitaan jotta mqtt-dataa voidaan esittää web-sivulla.

```

GNU nano 5.4 /etc/mosquitto/mosquitto.conf
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example

pid_file /run/mosquitto/mosquitto.pid

persistence true
persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

#include_dir /etc/mosquitto/conf.d #kommentoitu koska haluamme oletusasetukset

#seuraavat rivit lisätty jälkikäteen
allow_anonymous true
password_file /etc/mosquitto/pwfile
listener 1883

# websockets
listener 9001
protocol websockets

```

Kuva 16, muutettu mosquitto.conf.

Lisätään mqtt-käyttäjä pi

```
sudo -h sudo mosquitto_passwd -c /etc/mosquitto/pwfile pi
```

Mosquiton käynnistys raspin käynnistyessä

```
sudo systemctl enable mosquitto
```

Tässä vaiheessa aloin korjaamaan sarjaliikenteen ongelmaa, raspiin saapuva data nimittäin ei ollut sitä mitä pitäisi (kuva 17).

```

File Edit Tabs Help

Wed Aug 9 15:44:59 2023
26.00
Wed Aug 9 15:45:02 2023
0
Wed Aug 9 15:45:02 2023
60
Wed Aug 9 15:45:05 2023
0
Wed Aug 9 15:45:05 2023
6
Wed Aug 9 15:45:08 2023
6.0
Wed Aug 9 15:45:11 2023

```

Kuva 17, sarjaliikennettä käyttävä mqtt-data saapuu perille raspin konsoliin esp:stä puutteellisena.

Pitkällisen säätämisen jälkeen päätin luovuttaa uart-liikenteen käytöstä ja siirtyä jo kokeiltuun wifiin. Esp:en latasin koodin joka julkaisee mqtt-viestejä ja raspiin vastaanottavan koodin. Epäselväksi jäi miksi sarjaliikenne ei toiminut luotettavasti, tosin vastaavia esimerkkejä löytyi netistä ja korjausohjeitakin mutta mikään ei tuntunut toimivan.

Lisäsin crontabiin komennon

```
@reboot sudo python3 /home/pi/subscribe_data_from_esp32mqtt.py
```

joka käynnistää vastaanottavan koodin kun raspi käynnistetään.

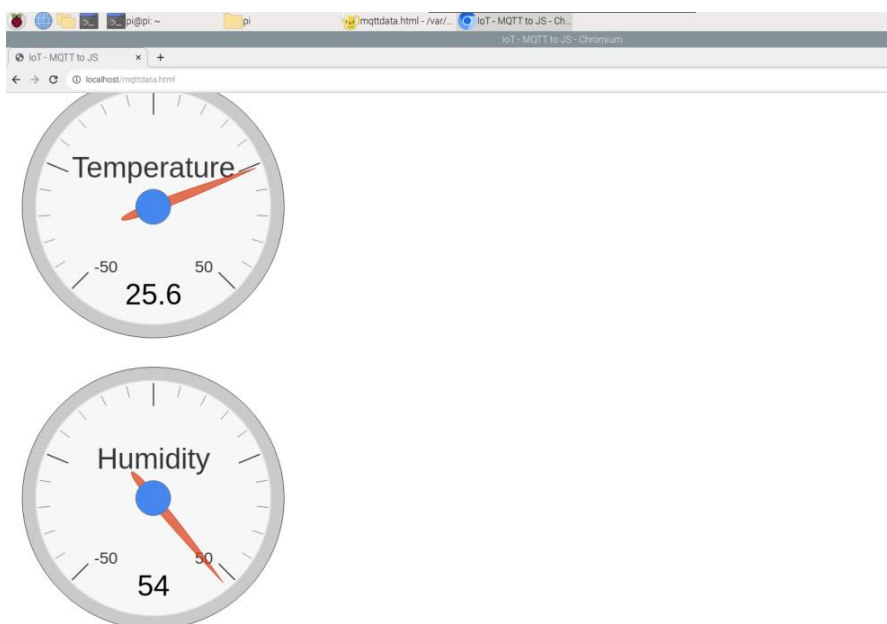
Wifi/mqtt tuntuu toimivan luotettavasti verrattuna edelliseen sarjaliikenne-yritykseen (kuva 18).

```
File Edit Tabs Help
^Cpi@pi:~$ sudo mosquitto_sub -h localhost -p 1883 -t Humidity -u pi -P pi
53.00
53.00
53.00
53.00
53.00
52.00
53.00
53.00
53.00
^Cpi@pi:~$ sudo mosquitto_sub -h localhost -p 1883 -t Temperature -u pi -P pi
29.00
29.00
28.90
29.00
28.90
29.00
29.00
29.00
28.90
29.00
28.90
28.90
^Cpi@pi:~$
```

Kuva 18, data saapuu wifin avustuksella kokonaisena.

Sitten alkoi miettiminen kuinka data saadaan web-sivulle. Löysin toimivan ohjeen osoitteesta <https://funprojects.blog/2018/11/28/mqtt-and-javascript/> ja jonkin ajan säätämisen jälkeen data oli visualisoitu. Javascript-osio käyttää Google Charts-nimistä palvelua jossa on valmiita graafisia esitystapoja datalle.

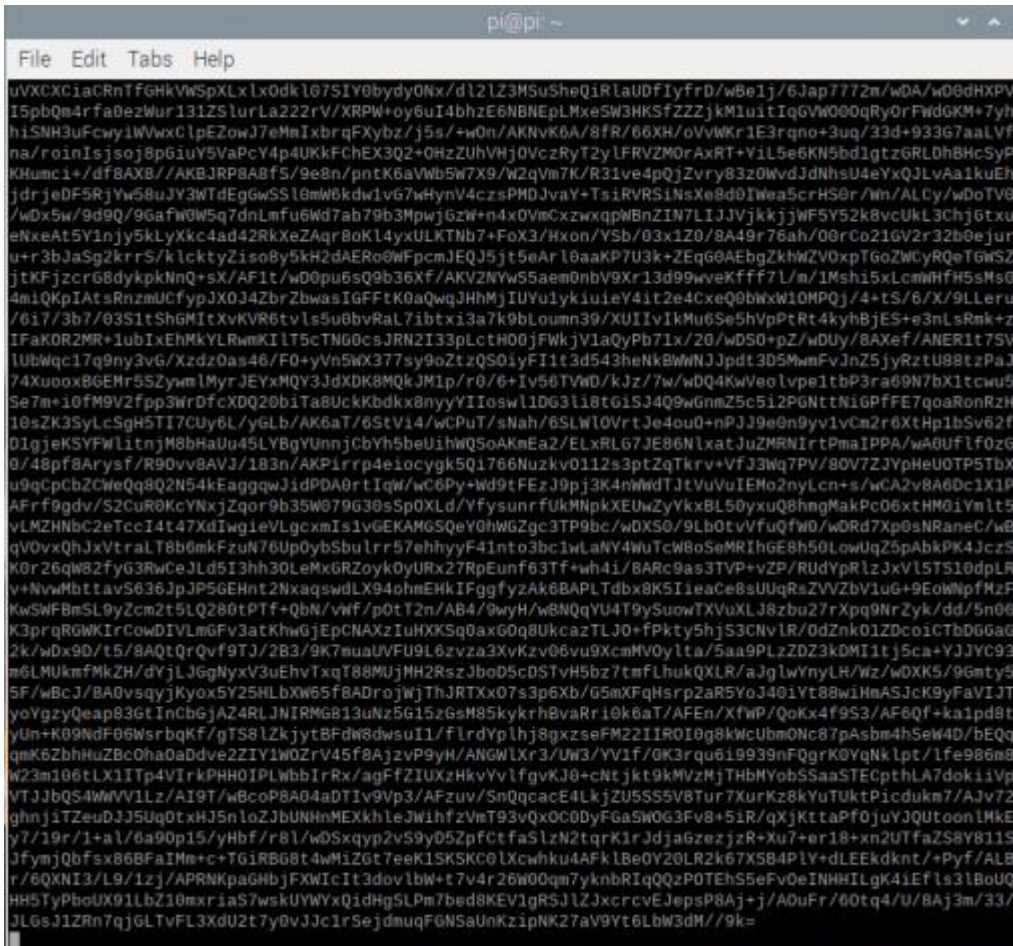
Selaimessa osoitteessa localhost/mqttdata.html voi todeta että data tulee toivotusti perille (kuva 19).



Kuva 19, mqtt-data on luettavissa selaimessa.

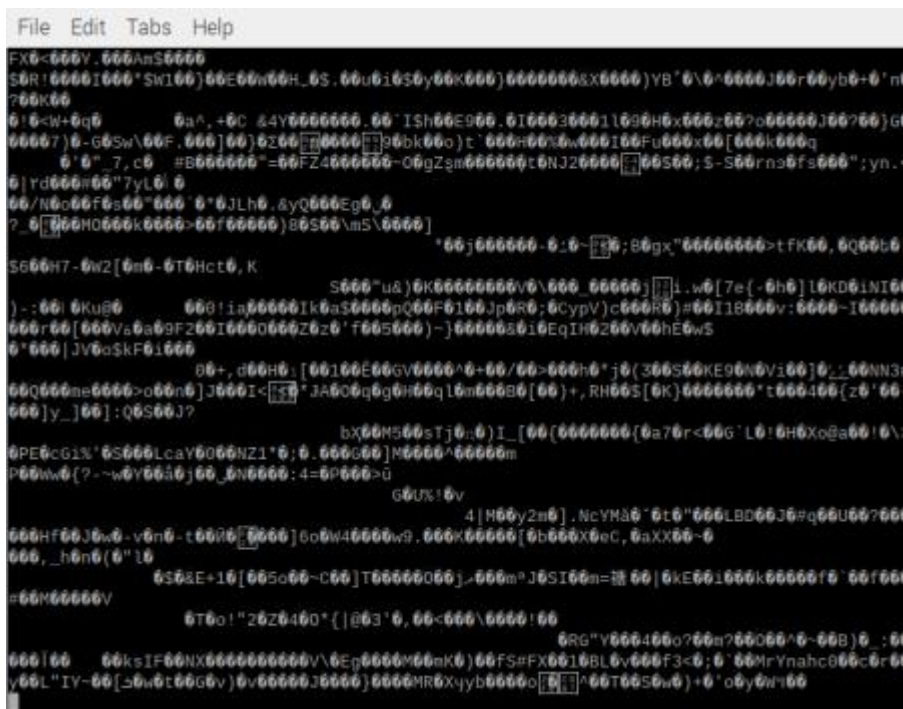
Sitten oli aika perehtyä kuinka kameran kuva saadaan web-sivulle. Aluksi tarkistin että kameran ja kameran laukaisevan painonapin kytkennät ovat oikein ja latsin raspiin

yksinkertaisen python-koodin joka tallentaa kuvan tiedostoon. Todettuani että kaikki toimii odotetusti oli aika tutkia missä muodossa kuva tulisi julkaista mqtt:llä. Löysin monia ohjeita joissa kehoitettiin muuttamaan kuva joko bytearrayksi tai käyttämään base64:ää joka muuttaa kuvan ASCII-muotoon. Base64-menetelmällä sainkin kuvan julkaistua terminaaliin jolloin kuva oli muuttunut ASCII-muotoon ja terminaali täyttyi erilaisista merkeistä (kuva 20).



Kuva 20, mqtt-subscribe terminaaliissa ja kameran ottama kuva muutettu ASCII-muotoon. Terminaalia ylöspäin selaamalla selaamalla merkkejä löytyy vielä lisää.

En kuitenkaan onnistunut muuttamaan ASCII-muotoista kuvaa jpg-muotoon vaan kuvan tilalla selaimessa näkyi samanlaiset merkit kuin terminaaliin julkaistussa kuvassa. Lopulta selvisi että kuvan siirtämiseen mqtt:llä toimii aivan hyvin binäärimuotoinen data joka saadaan aikaiseksi pythonin yksinkertaisella read()-funktioilla. Kuvassa 21 mqtt-subscribe terminaaliissa jossa kuva binäärimuodossa read()-funktioilla aukaistuna.



Kuva 21, kameran ottama kuva binäärimuodossa.

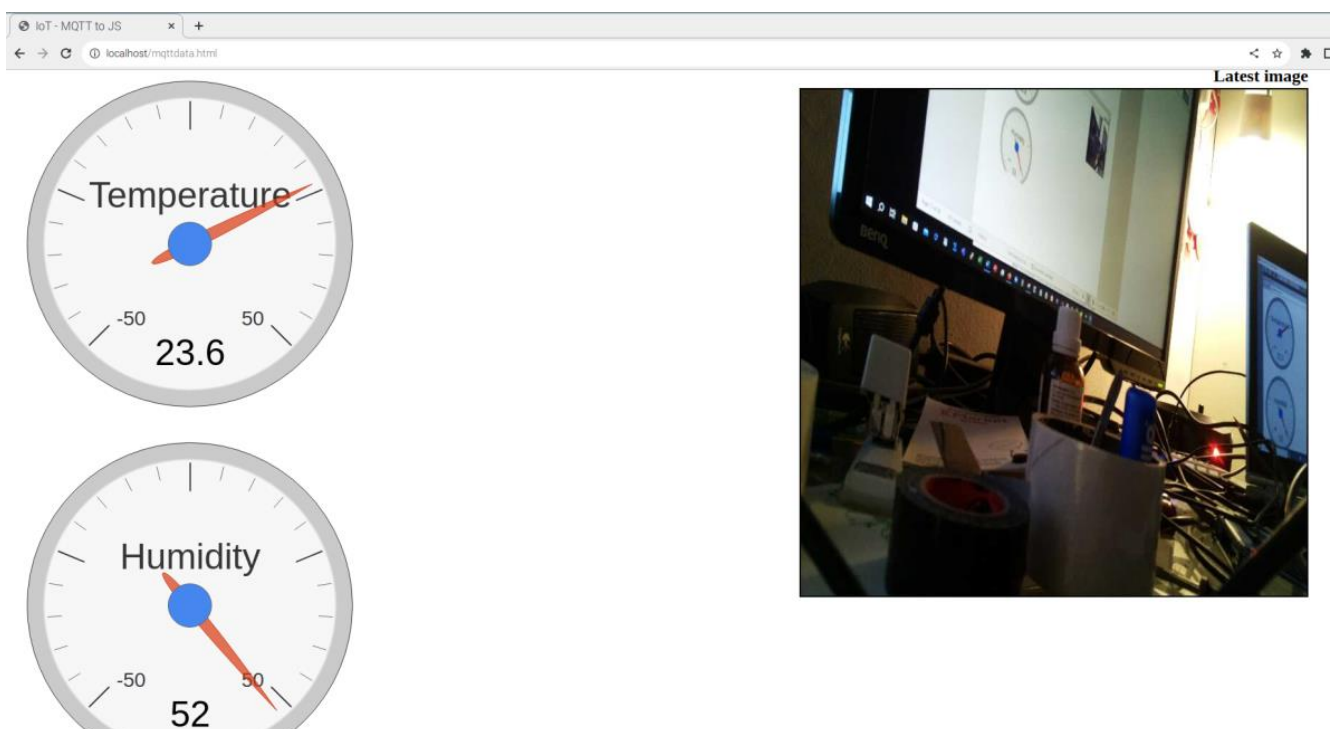
Seuraavana oli jäljellä binäärimuotoisen kuvan muuttaminen takaisin kuvaksi serverin puolella.

Tutkimisen jälkeen kokeilin serverin javascript-koodissa blob:ia (binary large object).

```
var blob = new Blob([message.payloadBytes], {type: 'image/png'});
```

```
document.getElementById("latest_image").src = URL.createObjectURL(blob);
```

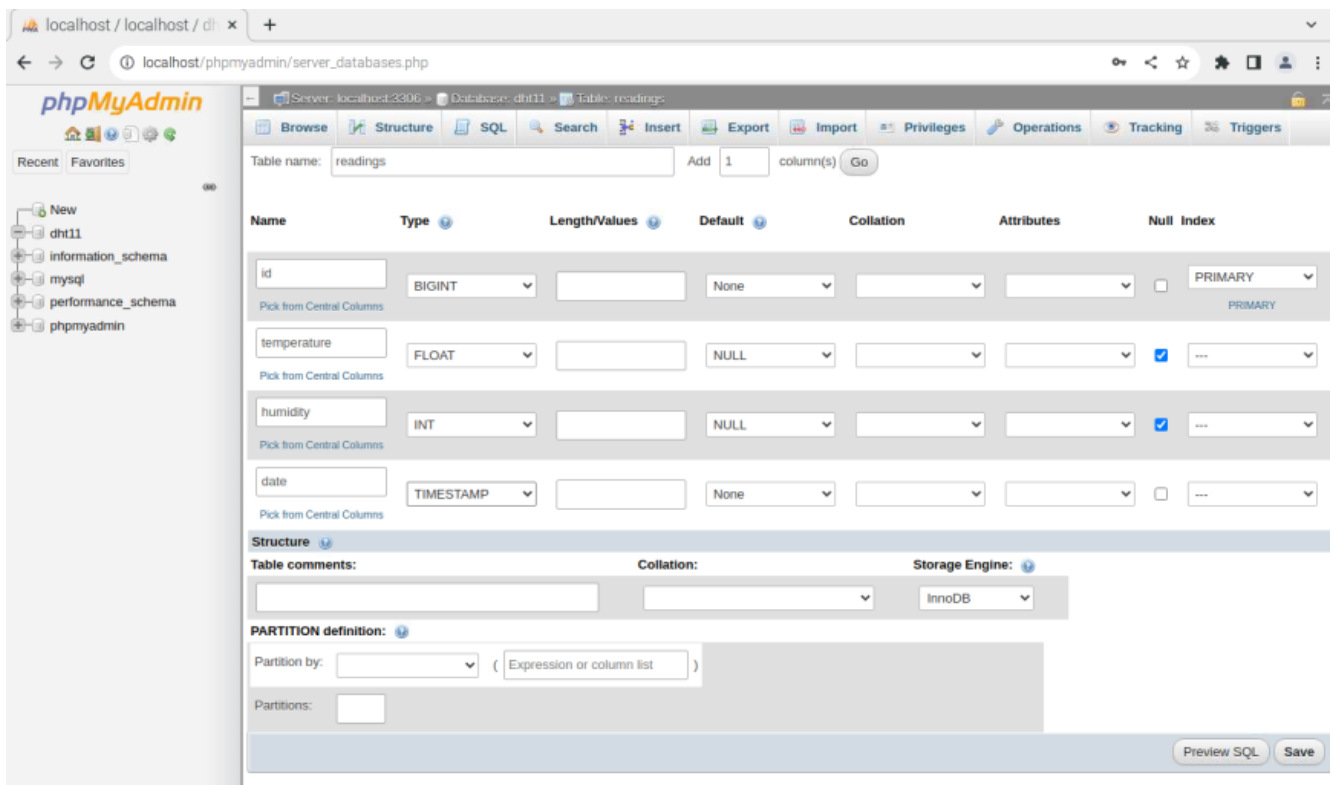
Kahden edellisen rivin koodilla kuva ilmestyi selaimen kuin taikauskusta (kuva 22).



Kuva 22, kameran kuva lisättyä web-sivulle



Sitten oli tietokannan vuoro, aloitin kirjautumalla phpmyadminiin ja tekemällä tietokannan "dht11"(kuva 23).



Kuva 23, tietokannan luomista phpmyadminissa.

Seuraavaksi komennolla

```
sudo pip3 install mysql-connector-python
```

python-liitännäinen jotta voidaan tallentaa tietokantaan pythonilla. Tämän jälkeen muokkasin raspissa koodia joka vastaanottaa dht:n lähettämää mqtt-dataa.

Mysql-terminaalissa komennolla

```
use dht11;
```

```
create user 'pi'@'localhost' identified by 'pi';
```

ja

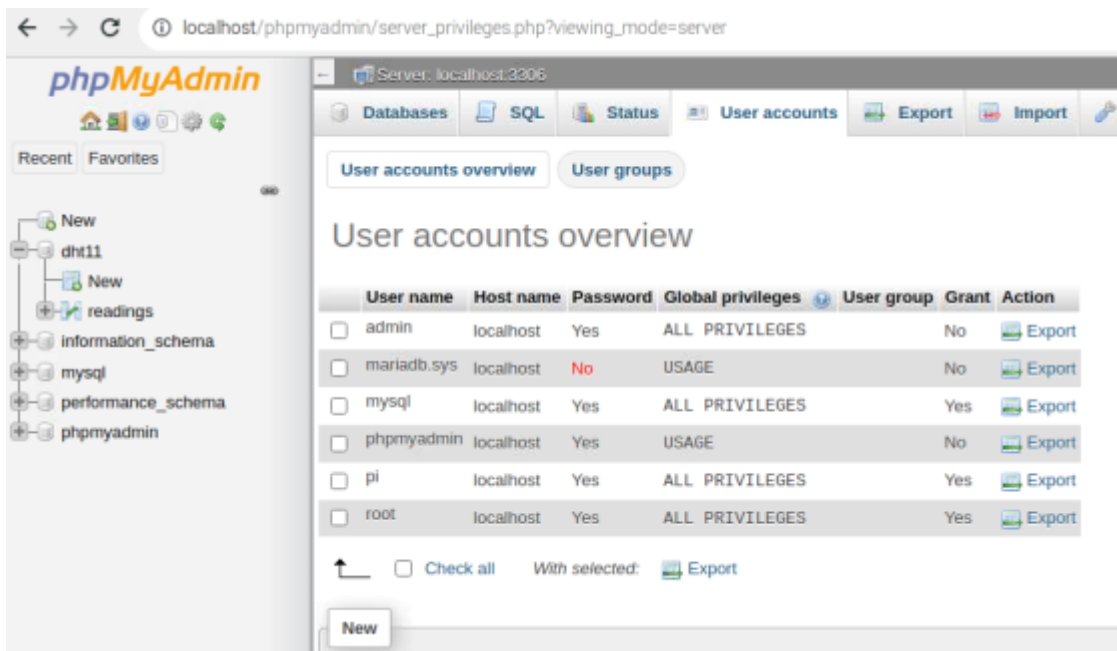
```
flush privileges;
```

loin käyttäjän tietokannalle. Vielä tässä vaiheessa python-koodi valitti että yhdistäminen ei onnistu. Lopulta aloin epäilemään että käyttöoikeuksissa on sanottavaa. Komennoilla

```
mysql -u root -p
```

```
GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' IDENTIFIED BY 'password';
```

sain käyttäjälle "pi" kaikki oikeudet (kuva 24). Tämäkään ei vielä riittänyt sillä raspin geany-koodieditori valitti koodin sisennyksistä sillä koodissa oli sekaisin sisennyksiä jotka oli tehty välilyönneillä että tabulaattorilla. Korjausten jälkeen testidataa onnistui tallentaa tietokantaan.



Kuva 24, käyttöoikeuksien tarkastelua phpmyadminissa.

Kun testidataa oli saatu tallennettua tietokantaan oli varsinaisen anturidatan vuoro. Mietintää aiheutti raspin python/mqtt/subscribe-koodi johon oli tarkoitus lisätä tietokantaan tallentava osuus. Lopulta sijoitin tietokantaoperaatiot mqtt:n "on message"-funktioon joka ajetaan aina kun mqtt ottaa vastaan dataa.

Lopulta sain myös varsinaisen anturidatan tallentumaan tietokantaan. Kuvassa 25 näkymä phpmyadminista jossa anturin antamia lukemia.

Funktio "on message" ajetaan aina kun jokin topic saa dataa joten muiden topicien arvot ovat tällöin nolla. Yritin tehdä koodista sellaisen että tallennus tietokantaan tapahtuu vasta kun kummassakin topicissa on mitatut arvot mutta vielä tätä kirjoittaessa en siinä ollut onnistunut.

localhost/phpmyadmin/sql.php?server=1&db=dht11&table=readings&pos=0

Server: localhost:3306 » Database: dht11 » Table: readings

Options: ☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	id	temperature	humidity	date
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	54	0	54	2023-08-24 17:04:05
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	52	0	54	2023-08-24 17:04:02
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	38	0	54	2023-08-24 17:03:41
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	50	0	54	2023-08-24 17:03:59
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	40	0	54	2023-08-24 17:03:43
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	48	0	54	2023-08-24 17:03:57
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	42	0	54	2023-08-24 17:03:46
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	46	0	54	2023-08-24 17:03:52
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	44	0	54	2023-08-24 17:03:49
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	22.2	66	2023-08-19 17:25:57
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	41	23.1	0	2023-08-24 17:03:46
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	51	23.1	0	2023-08-24 17:04:02
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	37	23.1	0	2023-08-24 17:03:41
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	43	23.2	0	2023-08-24 17:03:49
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	47	23.2	0	2023-08-24 17:03:55
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	49	23.2	0	2023-08-24 17:03:58
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	39	23.2	0	2023-08-24 17:03:43
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	53	23.2	0	2023-08-24 17:04:05
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	45	23.2	0	2023-08-24 17:03:52

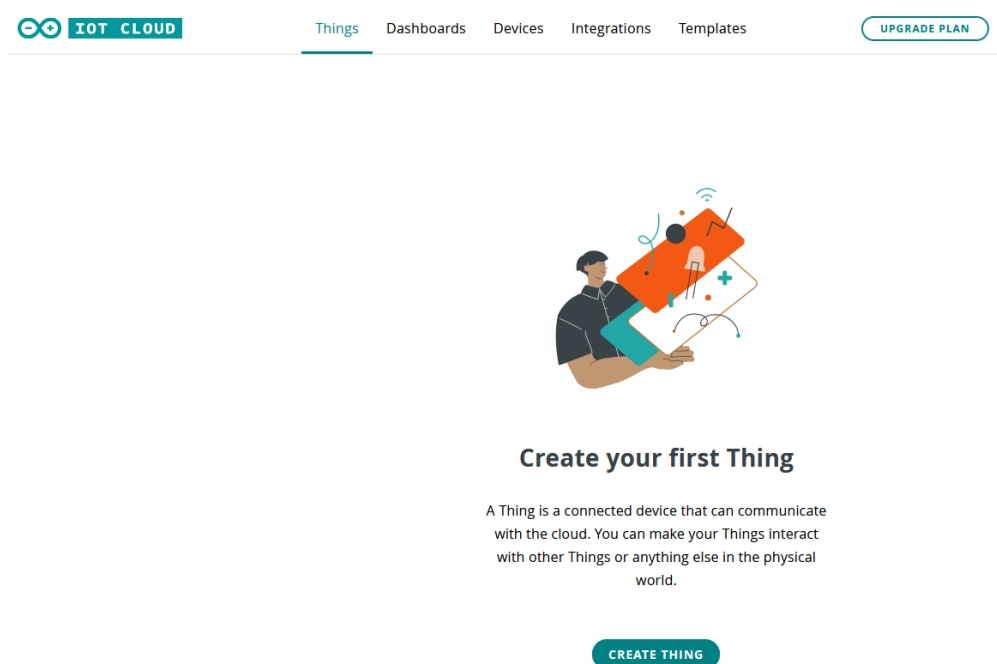
Options: ☐ Check all | With selected: ☐ Edit ☐ Copy ☐ Delete ☐ Export

Options: ☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

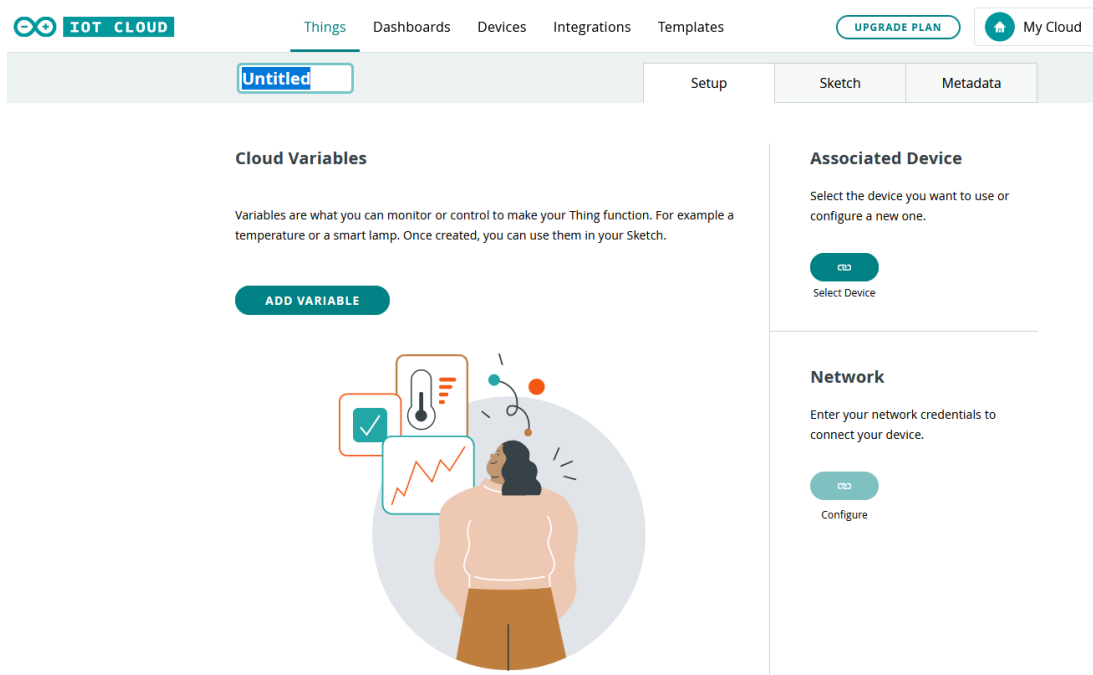
Kuva 25, anturin dataa tallennettuna tietokantaan. Tallennus tapahtuu aina kun jokin topic saa dataa joten muiden topicien arvot ovat silloin nollia.

Tietokantaan tallennus tuntuu toimivan. Mutta vain mikäli "subscribe\_data\_from\_esp32mqtt.py" suoritetaan koodieditorista käsin tai terminaalista. Outoa sikäli että koodissa oleva mqtt-osuus toimii mutta tietokantaosuus ei. Tämä tiedosto on siis crontabissa varustettu @reboot-argumentilla jotta suoritus käynnistyy raspin käynnistyessä. Epäilin tässä vaiheessa tiedoston käyttöoikeuksia. Lopulta syyksi osoittautui se että python-koodin suoritus käynnistyi ennen kuin Mariadb. Korjaus onnistui yksinkertaisesti lisäämällä subscribe/tietokantakoodin alkuun python-funktio "time.sleep(5)" joka odottaa 5 sekuntia jotta Mariadb ehtii käynnistyä.

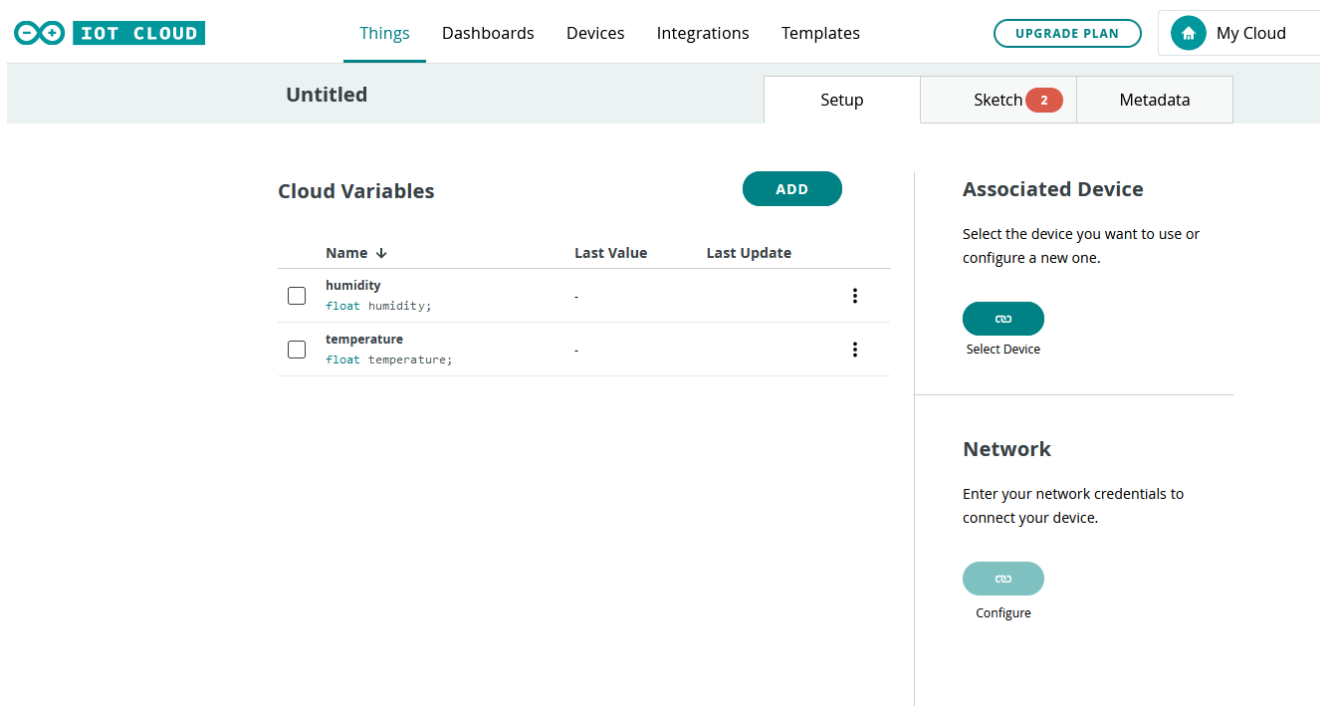
Jotta projektia voisi sanoa valmiiksi oli jäljellä vielä pilvipalvelun yhdistäminen jäljellä. Periaatteena ajattelin käyttää pilvipalvelua siihen että anturidatan saisi näkymään myös puhelimessa, ilman nettiselaimen käyttöä. Olin aiemmin käyttänyt Blynk Iot-palvelua mutta sen muututtua maksulliseksi käännysin Arduino Cloud:n puoleen. Aluksi ajattelin että yhdistän pilveen raspilla mutta luultavasti valmista tulee helpommin jos tekee yhdistämisen esp:llä. Arduino Cloudista saatavat koodin pätkät voi myös ladata suoraan esp:n kun taas raspilla c++-koodin suoritus on oma taiteenlajinsa. Aloitin kirjautumalla Arduino Cloudiin (kuva 26).



Kuva 26, Arduino IOT-cloud-näkymää.





Kuva 27, aloitetaan tekemällä muuttujat joiden arvot halutaan pilveen.




Kuva 28, muuttujina dht:n mittaamat "temperature" ja "humidity".

**RECOMMENDED** **AUTOMATIC** Auto-sketch generation, online editing, easy upload

  
**Arduino board** ⓘ  
⌘ Arduino language (C++)

  
**Third party device** ⓘ  
⌘ Arduino language (C++)

**MANUAL** Manual programming and upload, offline editing

  
**Any Device** ⓘ  
⌘ Python, MicroPython, JavaScript (NodeJS)

Kuva 29, määritellään käytettävä laite, tässä tapauksessa "third party device" kun kyseessä on esp32.

### Select device type

Please select the device type and model you want to configure

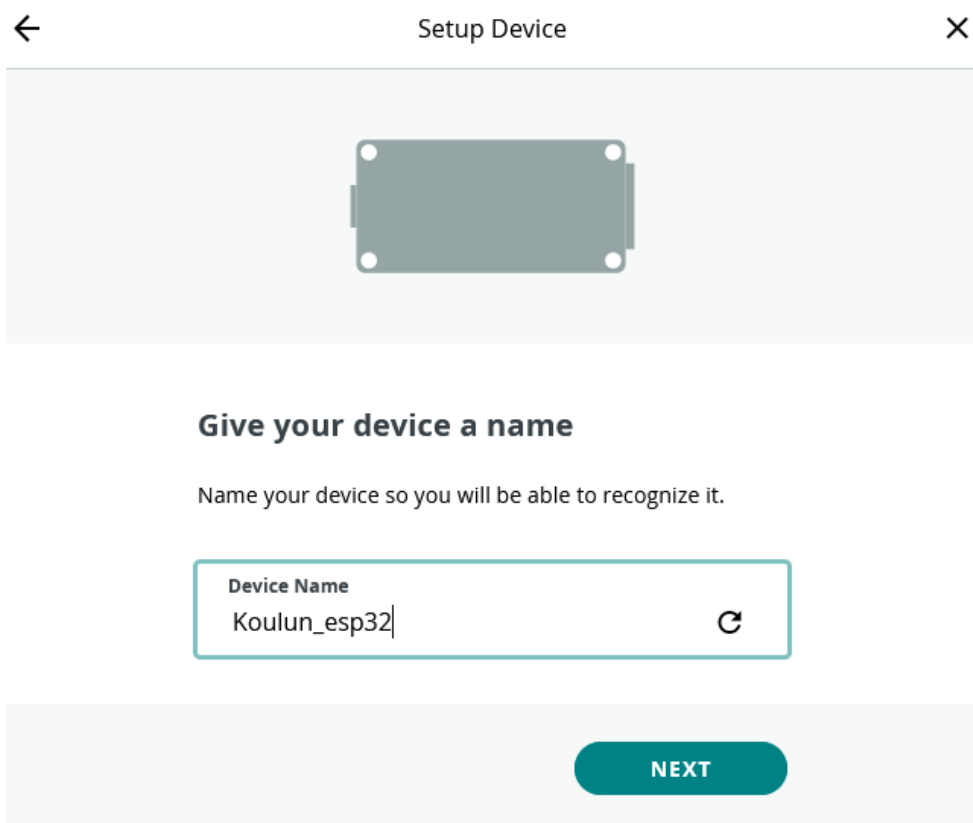
☐ ESP8266 ☒ ESP32 ☐ LoRaWAN

DOIT ESP32 DEVKIT V1 ▼

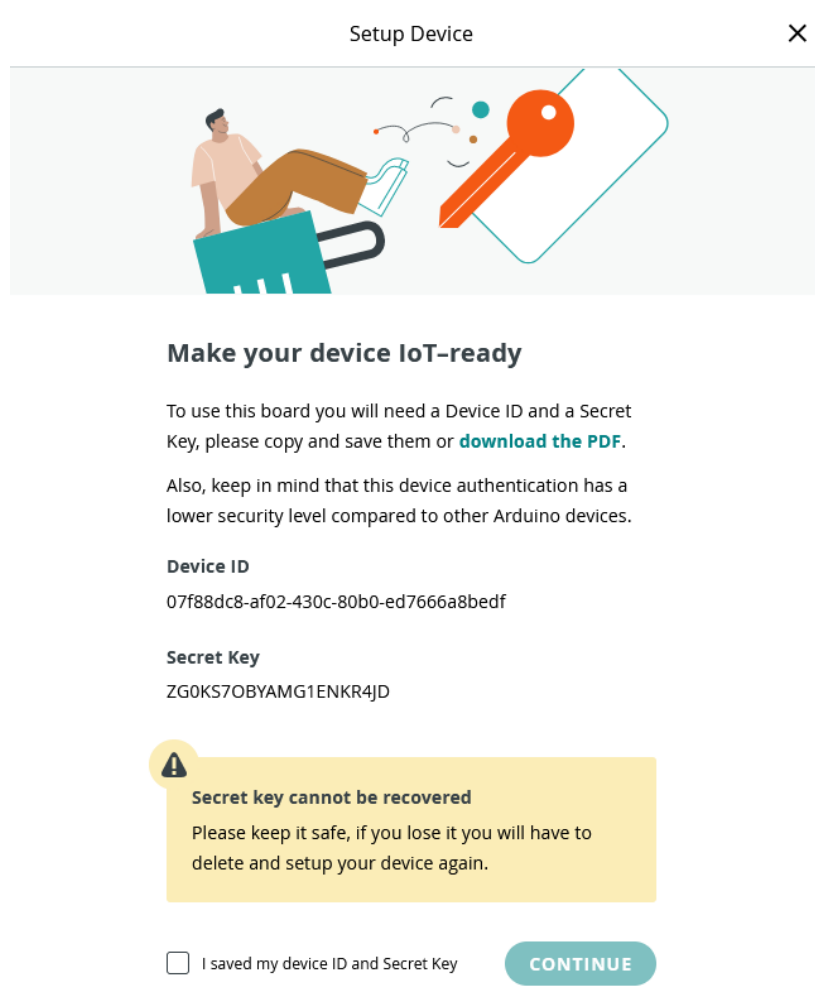
CONTINUE

Kuva 30, laitteen määrittäminen jatkuu.

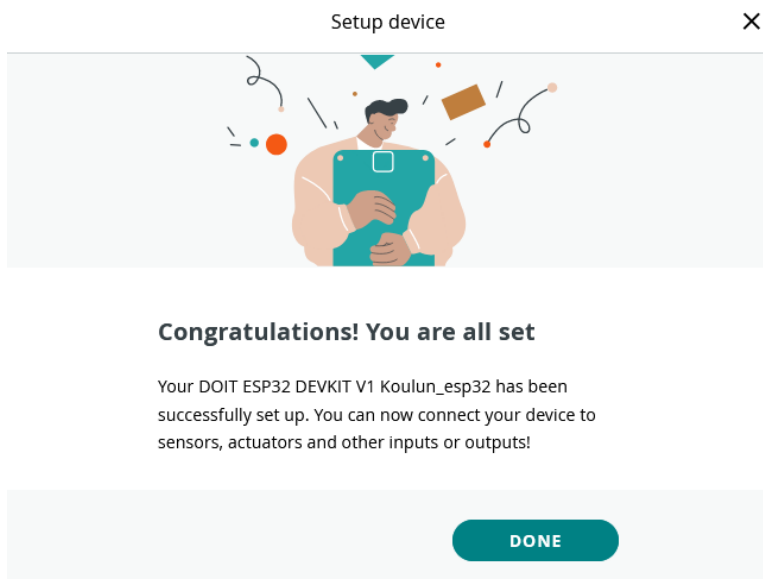




Kuva 31, laitteen määrittys jatkuu edelleen.



Kuva 32, käytettävälle laitteelle annetaan pilvestä yksilöivät tunnukset joita tarvitaan esp:n koodissa.



Kuva 33, laitteen määrittäminen on valmis.

IOT CLOUD

Things

Dashboards

Devices

Integrations

Templates

UPGRADE PLAN

My Cloud

Untitled

Setup

Sketch 2

Metadata

Cloud Variables

ADD

Name ↓	Last Value	Last Update
<div><input type="checkbox"/></div> <div>humidity</div> <div>float humidity;</div>	-	
<div><input type="checkbox"/></div> <div>temperature</div> <div>float temperature;</div>	-	

Associated Device

Koulun\_esp32

ID: 07f88dc8-af02-430c-80b0-e...

Type: DOIT ESP32 DEVKIT V1

Status: Offline

Change

Detach

Network

Enter your network credentials to connect your device.

Configure

Kuva 34, kaikki on valmista paitsi "network"-osio.

You will find these network parameters in the secret tab in your sketch, and your device will be able to connect to the network once the sketch will be uploaded.

[SAVE](#)

Kuva 35, käytettävän verkon tietoja, näitä tarvitaan myös esp:n koodissa.

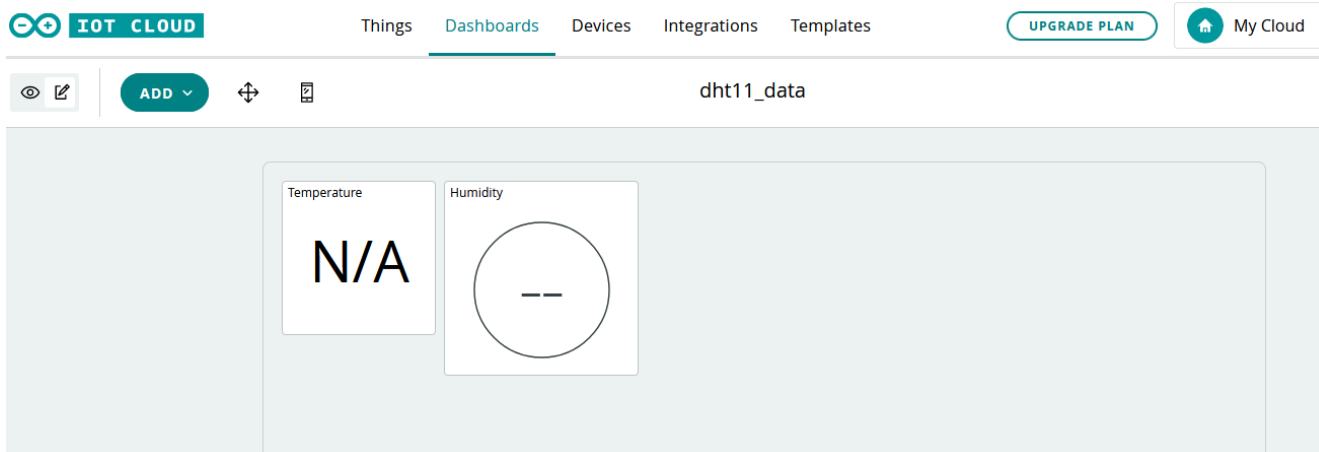


## Monitor your Things

Build a Dashboard to easily monitor the status of your Things and control them.

[BUILD DASHBOARD](#)

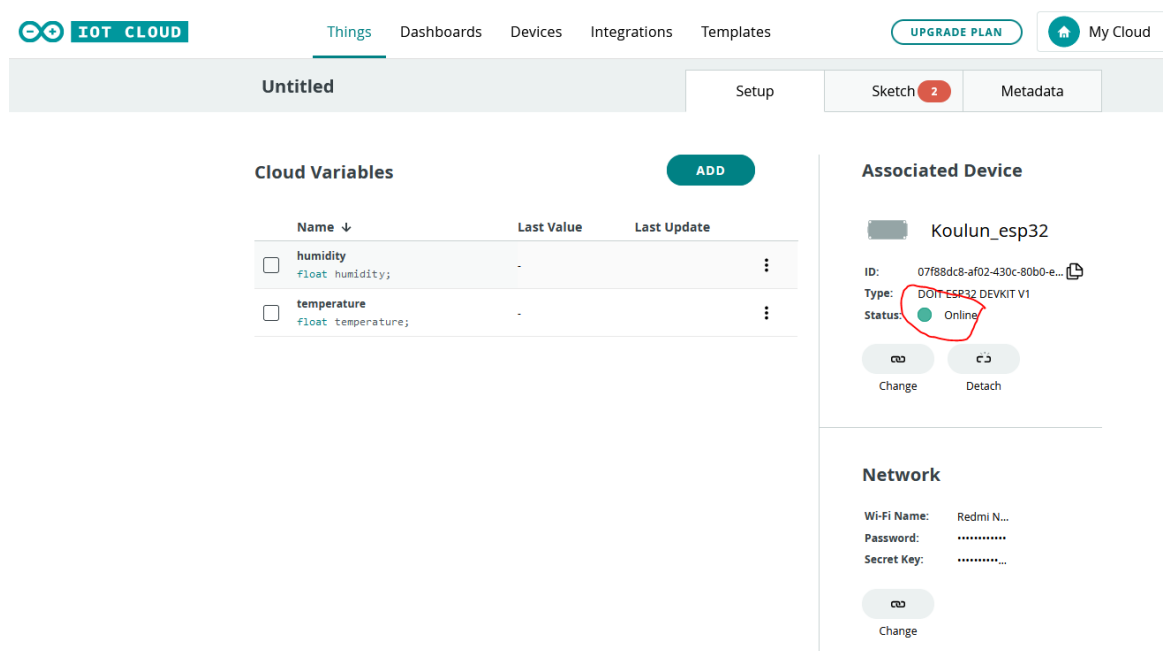
Kuva 36, vielä "dashboard" niin olisimme valmiita.



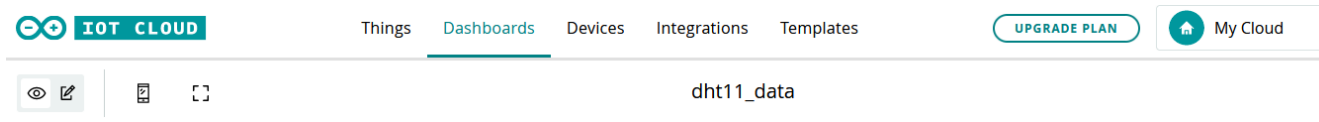
Kuva 37, valmis dashboard jossa näytöt lämpötilalle ja kosteudelle.

Tässä vaiheessa kaiken piti olla valmista mutta pilvestä saadussa koodissa oli lause `#include "thingProperties.h"`

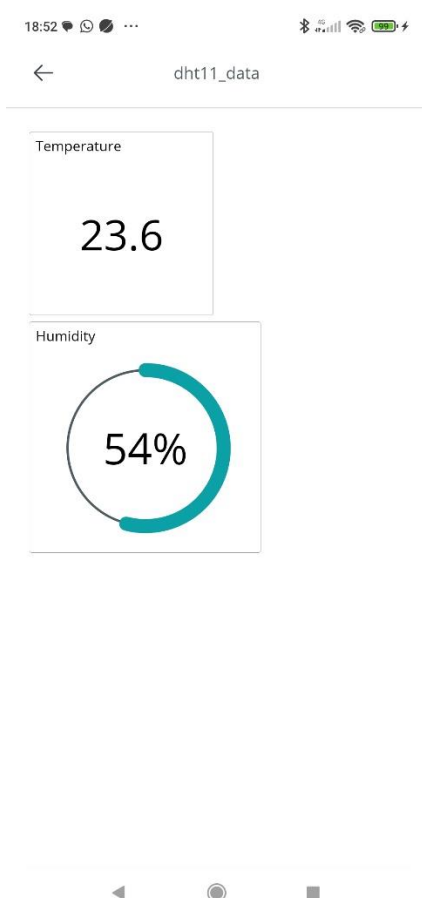
jota Arduino IDE ei ymmärtänyt. Etsin netistä sen nimisen kirjaston ja siirsin arduinon "libraries"-kansioon mutta tämäkään ei toiminut. Lopulta selvisi että koodi täytyy ladata esp:een käyttäen Arduino Cloudin omaa koodieditoria.



Kuva 38, kaikki valmiina ja laite yhdistetty.



Kuva 39, dht:n mitaamat lukemat dashboardissa.



Kuva 40, kuvakaappaus puhelimeen asennetusta Arduino Cloud-sovelluksesta. Dashboard on sama selaimessa tehty versio, kätevää verrattuna esimerkiksi Blynk IoT-pilveen jossa täytyy tehdä erikseen selaimen ja puhelimen dashboardit.

Raportin liitteistä löytyy käyttämäni koodit

**Velipekka Härkönen**  
Velipekka Härkönen