



**Mika Ylisirniö**

**Lämpötilan mittaaminen Arduino Uno:lla**  
**Azure IoT harjoitustehtävä**

**CENTRIA-AMMATTIKORKEAKOULU**  
**Huhtikuu 2023**

## SISÄLLYS

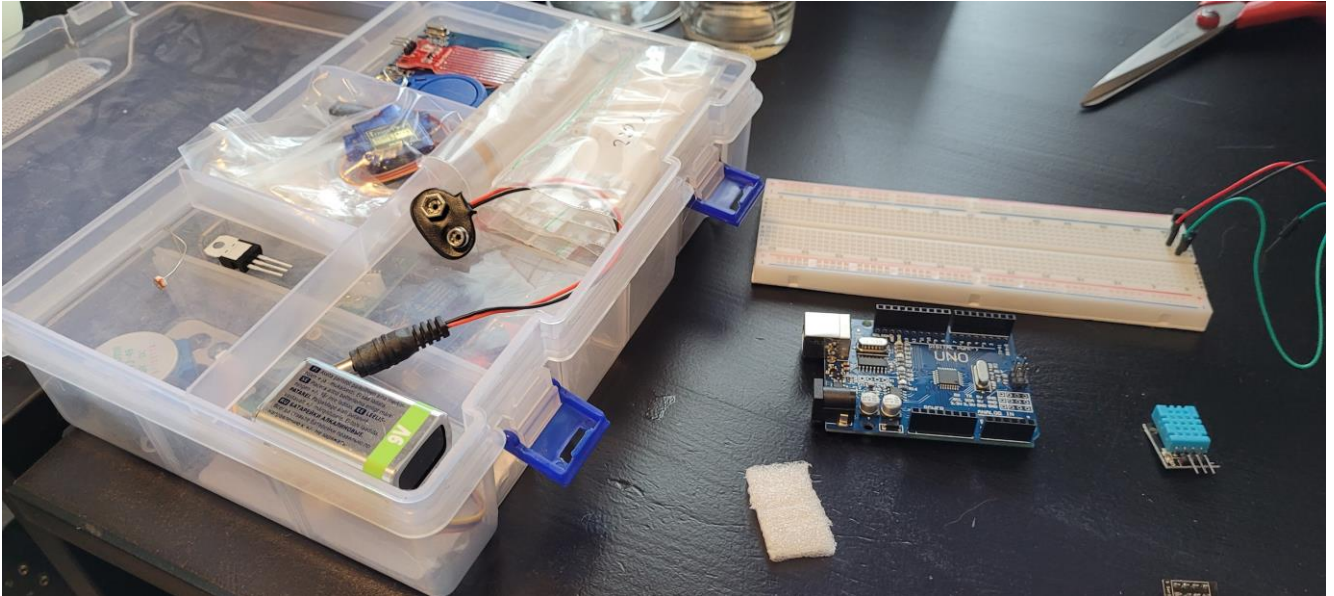
<b>1 JOHDANTO JA AIHEEN ESITTELY.....</b>	<b>1</b>
<b>2 RATKAISUMALLIN ESITTELY.....</b>	<b>4</b>
<b>3 TOTEUTUKSEN ESITTELY.....</b>	<b>5</b>
<b>4 POHDINTA.....</b>	<b>13</b>
<b>LÄHTEET .....</b>	<b>14</b>

## KUVAT

Kuva 1: Arduino UNO-kitti.....	1
Kuva 2: KY-015.....	1
Kuva 3: ESP8266-langaton moduuli .....	2
Kuva 4: TMP36-lämpötila-anturi .....	3
Kuva 5: Datan polku .....	4
Kuva 6: Arduino UNO + TMP36.....	5
Kuva 7: Komentokehotteessa pyörivä tmp36.js .....	5
Kuva 8: IoT Hubiin saapuneet viestit .....	6
Kuva 9: Osa master datasta .....	7

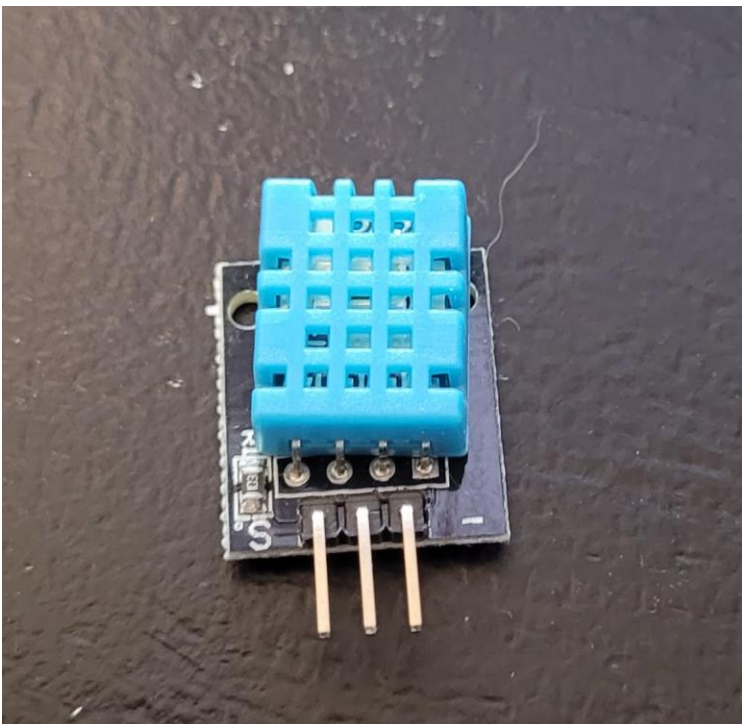
## 1 JOHDANTO JA AIHEEN ESITTELY

Hyllystä sattui löytymään aikoinaan hankkimani Arduino Uno-kitti,



*Kuva 1: Arduino UNO-kitti*

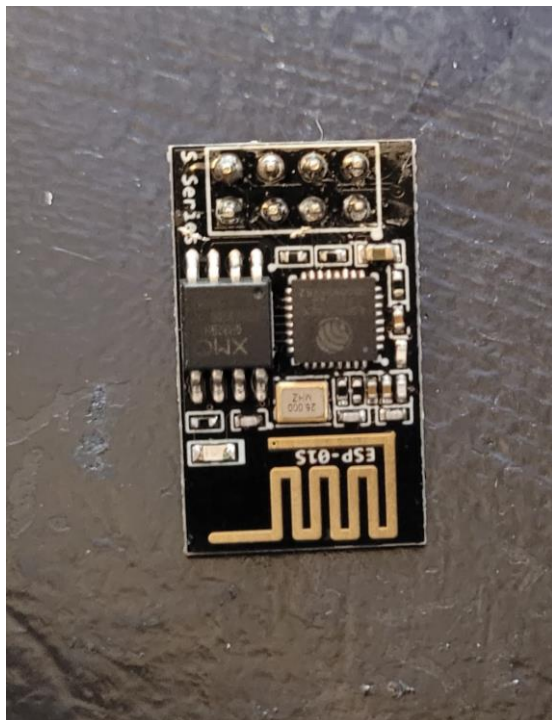
jonka mukana tuli mallia [KY-015](#) oleva digitaalinen lämpötila- ja kosteusanturi (Joy-IT: KY-015 COMBI-SENSOR (TEMPERATURE & HUMIDITY), ei pvm).



*Kuva 2: KY-015*

Alkuperäisenä suunnitelmana oli hankkia Arduinolle mallia [ESP8266](#) oleva Wi-Fi-moduuli (Wikipedia: ESP8266, 2023), mitata lämpötila- ja kosteusarvoja ainakin yhden päivän ajalta, lähettää tiedot Azureen ja käsitellä niitä siellä, mutta asioilla ei kuitenkaan ole tapana mennä kuin Strömsössä...

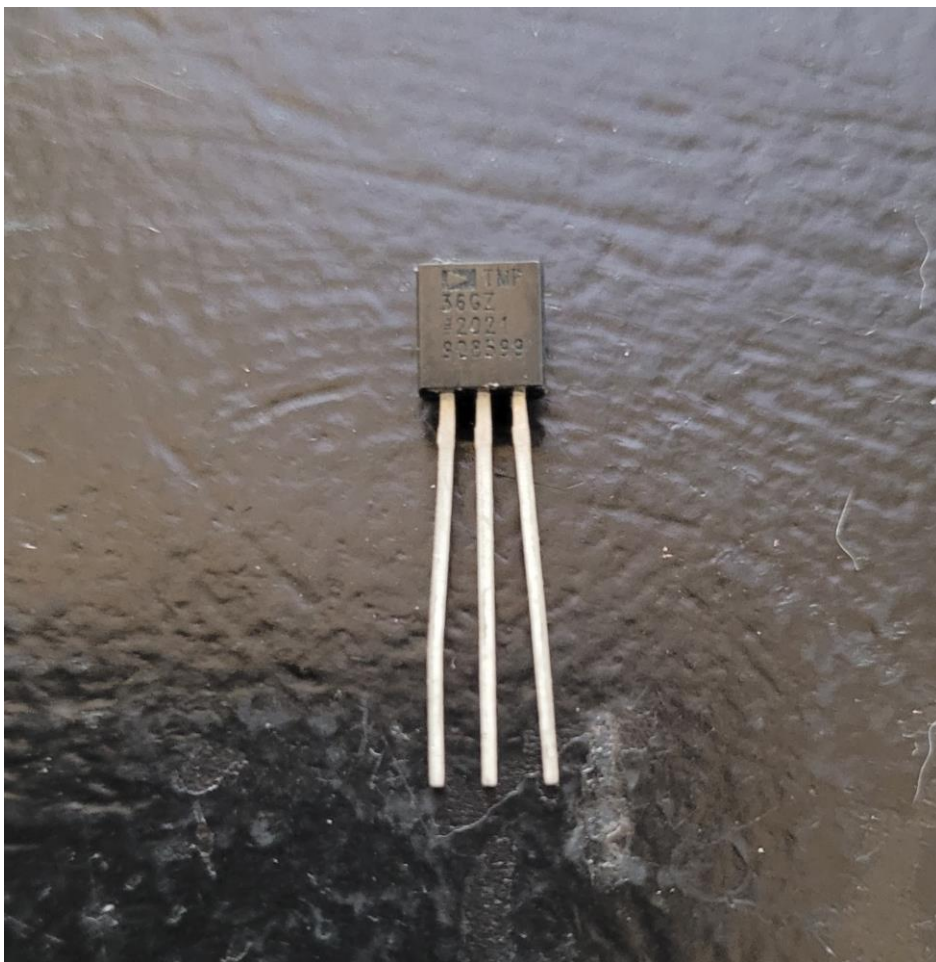
Arduino Uno ja hankkimani ESP8266-moduuli eivät oikein suostuneet kommunikoimaan keskenään.



*Kuva 3: ESP8266-langaton moduuli*

Parin tunnin tappelun jälkeen sain moduuliin lopultakin ajettua testiskriptin, joka sai sen ledin vilkkumaan, mutta tässä vaiheessa totesin, että raportin deadline lähestyy jo sellaista vauhtia, että on vaihdettava suunnitelmaa. Pienen Googlettelun jälkeen alkoi myös vaikuttamaan siltä, että Arduino Uno ja ESP8266 moduuli eivät ainakaan ihan suoraltaan ja helposti pystyisi lähettämään dataa suoraan Azure IoT Hubiin (Microsoft: Streaming IoT Telemetry to Azure Event Hub : Part 1, 2016).

Varasuunnitelmana oli seurata näitä [ohjeita](#) (Thinglabs.io: Sending Telemetry to the Cloud, ei pvm) ja lähettää data tietokoneen välityksellä Azureen. Paljastui kuitenkin, että ohjeessa hyödynnetyt Johnny-Five kirjastot eivät suoraltaan tue KY-015-moduulia, joten se oli vaihdettava yksinkertaisempaan TMP36-lämpötila-anturiin (GitHub: Adding DHT11 or DHT22 sensor support, ei pvm).

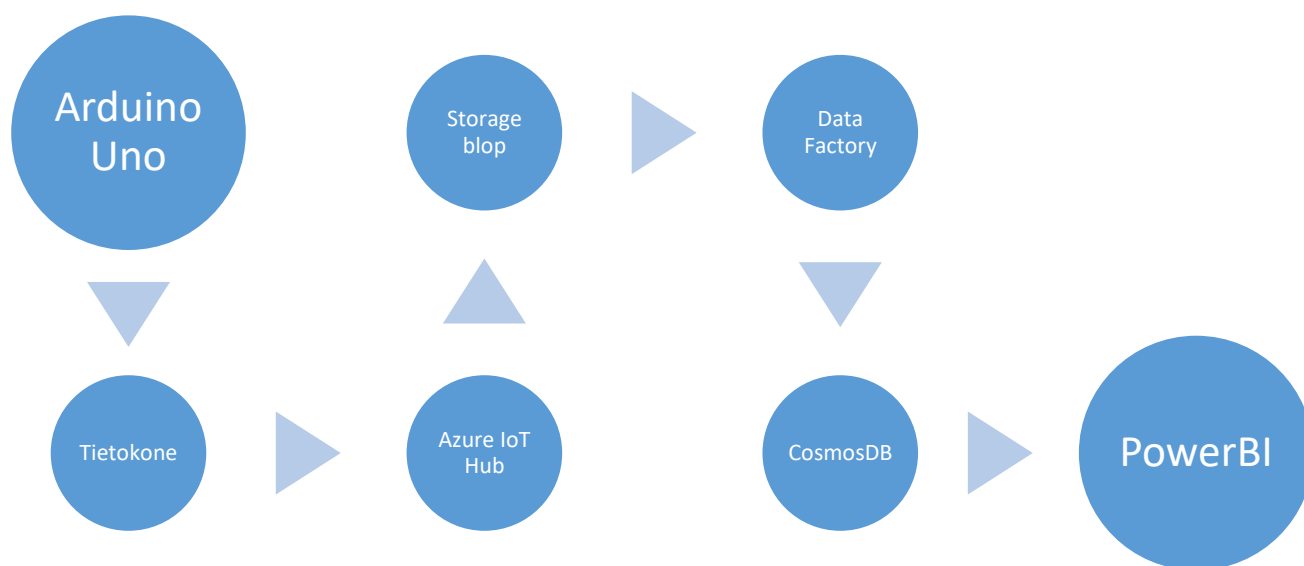


*Kuva 4: TMP36-lämpötila-anturi*

Näiden vaikeuksien ja useampien tuntien aiheeseen perehtymisen jälkeen olin kuitenkin lopultakin valmis toteuttamaan projektin.

## 2 RATKAISUMALLIN ESITTELY

Projektissa suurimman osan ajasta söi oikean mittausdatan saaminen Azureen, joten sen käsittely Azuressa jätettiin varsin yksinkertaiseksi ja enimmäkseen seurattiin vain tunneilla olutta mallia:



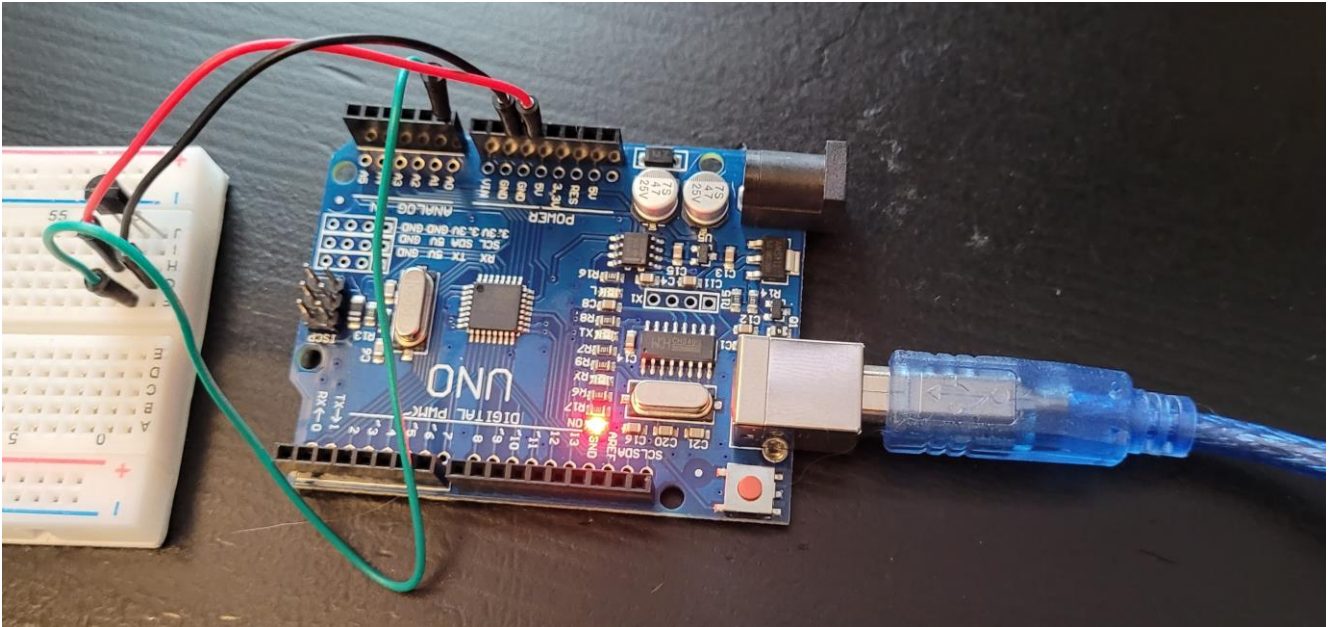
*Kuva 5: Datan polku*

Mittausdata tallennetaan Master datana Storage Blob-säilöön, sille tehdään Cold Path-mallin mukainen eräajokäsittely Data Factorya käyttäen, eräajon tulokset viedään CosmosDB:hen, josta niitä lopuksi visualisoidaan käyttäen PowerBI:tä.



### 3 TOTEUTUKSEN ESITTELY

Mitattava data kerättiin yhden vuorokauden ajalta Arduino Unoan kytketyllä TMP36-lämpötila-anturilla.



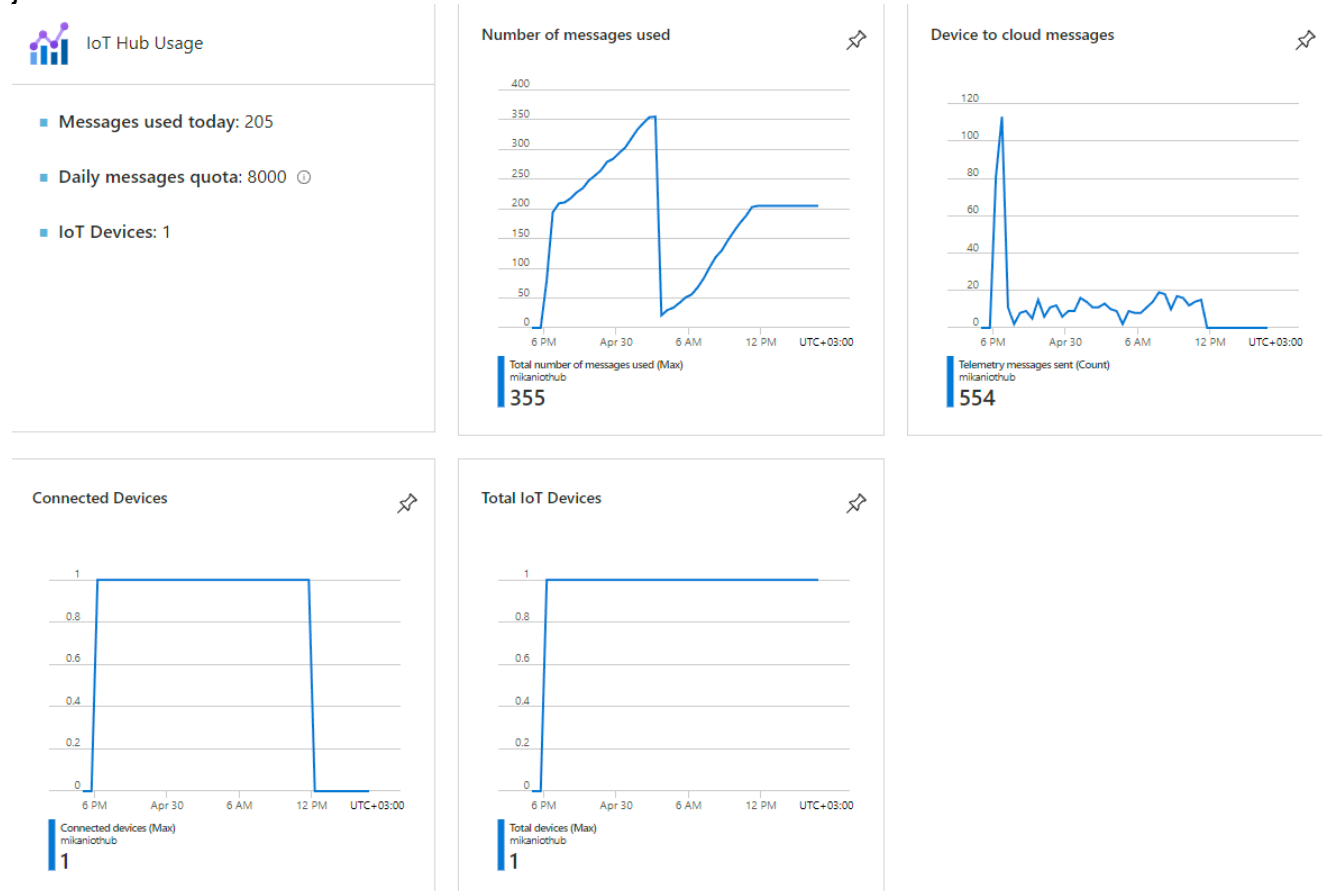
Kuva 6: Arduino UNO + TMP36

Arduino Unoan kytketyllä tietokoneella ajettiin [ohjeiden](#) pohjalta laadittua Node.js skriptiä (liitteenä tiedostossa **tmp36.js**), joka mittasi huoneen lämpötilaa kerran minuutissa:

```
G:\Google Drive\Opiskelu\Centria\IoT ja Big Data\Harjoitustyö>node tmp36.js
1682781236926 Connected COM7
1682781240774 Repl Initialized
>> Board connected...
Sending message: {"deviceId":"MikanArduinoUno","location":"Tyohuone","celsius":22,"fahrenheit":71.6,"kelvin":295.15}
send status: undefined undefined
Sending message: {"deviceId":"MikanArduinoUno","location":"Tyohuone","celsius":23,"fahrenheit":73.4,"kelvin":296.15}
send status: undefined undefined
Sending message: {"deviceId":"MikanArduinoUno","location":"Tyohuone","celsius":22,"fahrenheit":71.6,"kelvin":295.15}
send status: undefined undefined
Sending message: {"deviceId":"MikanArduinoUno","location":"Tyohuone","celsius":21,"fahrenheit":69.8,"kelvin":294.15}
send status: undefined undefined
```

Kuva 7: Komentokehotteessa pyörivä tmp36.js

ja lähetti mittausdatan Azure IoT Hubiin



Kuva 8: IoT Hubiin saapuneet viestit

Tarkemmin tmp36.js skriptin pohjana käytettiin [täältä](#) löytyviä skriptejä **weather.js** ja **simulator.js**. Näiden pohjalta syntyneeseen koodin lisättiin Johnny-Five [sivuston](#) mallikoodit TMP36-anturiin liittyen.



IoT Hub kytkettiin tunneilla olleen mallin pohjalta Storage Accountiin, jonne luotiin Master dataa varten Blob container


Name	Last modified	Access tier	Blob type	Size	Lease state
[-]					
04.json	30.4.2023 2.05.55	Hot (Inferred)	Block blob	1.09 KiB	Available
06.json	30.4.2023 2.07.55	Hot (Inferred)	Block blob	1.09 KiB	Available
08.json	30.4.2023 2.09.55	Hot (Inferred)	Block blob	556 B	Available
16.json	30.4.2023 2.17.55	Hot (Inferred)	Block blob	1.09 KiB	Available
18.json	30.4.2023 2.19.55	Hot (Inferred)	Block blob	556 B	Available
20.json	30.4.2023 2.21.56	Hot (Inferred)	Block blob	1.09 KiB	Available
24.json	30.4.2023 2.25.56	Hot (Inferred)	Block blob	556 B	Available
30.json	30.4.2023 2.31.56	Hot (Inferred)	Block blob	556 B	Available
32.json	30.4.2023 2.33.56	Hot (Inferred)	Block blob	556 B	Available
43.json	30.4.2023 2.44.56	Hot (Inferred)	Block blob	1.09 KiB	Available
45.json	30.4.2023 2.46.56	Hot (Inferred)	Block blob	556 B	Available
47.json	30.4.2023 2.48.56	Hot (Inferred)	Block blob	556 B	Available
53.json	30.4.2023 2.54.56	Hot (Inferred)	Block blob	1.09 KiB	Available
56.json	30.4.2023 2.57.56	Hot (Inferred)	Block blob	1.09 KiB	Available
58.json	30.4.2023 2.59.56	Hot (Inferred)	Block blob	556 B	Available

Kuva 9: Osa master datasta


Yksittäiset tiedot olivat muodossa

```
{
  "EnqueuedTimeUtc": "2023-04-29T23:57:32.3100000Z",
  "Properties": {},
  "SystemProperties": {
    "to": "/devices/MikanArduinoUno/messages/events",
    "connectionDeviceId": "MikanArduinoUno",
    "connectionAuthMethod": "{ \"scope\": \"device\", \"type\": \"sas\", \"issuer\": \"iothub\", \"acceptingIpFilterRule\": null }",
    "connectionDeviceGenerationId": "638183779830879721",
    "contentType": "application/json",
    "contentEncoding": "utf-8",
    "enqueuedTime": "2023-04-29T23:57:32.3100000Z"
  },
  "Body": {
    "deviceId": "MikanArduinoUno",
    "location": "Työhuone",
    "celsius": 22,
    "fahrenheit": 71.6,
    "kelvin": 295.15
  }
}
```

ja niitä kertyi kaiken kaikkiaan


**Blob containers**

Number of containers	1
Number of blobs	244
Total data stored	448.97 KiB


**Tables**

Number of tables	8
Number of entities	74
Total data stored	250.56 KiB

Data Factoryssa tiedoista johdettiin Timestamp-sarake

Derived column's settings
Optimize
Inspect
Data preview

Output stream name \*
[Learn more](#)

Description
[Reset](#)

Incoming stream \*

+ Add
Clone
Delete
Open expression builder

Columns \*

Column	Expression
<input type="checkbox"/> Timestamp	toTimestamp(EnqueuedTimeUtc, 'yyyy-MM-dd\...' <a href="#">+</a> <a href="#">-</a> )

ja tiedot aggregoitiin sekä päivämäärän että tunnin mukaan

Aggregate settings   Optimize   Inspect   Data preview ●

---

Output stream name \*  [Learn more](#)

Description 

Aggregating data by 'Date, Hour' producing columns 'maxTempCelsius'

[Reset](#)

Incoming stream \*

**Group by**   Aggregates

Columns	Name as
<input type="text" value="toDate(Timestamp)"/>	<input type="text" value="Date"/>
<input type="text" value="hour(Timestamp)"/>	<input type="text" value="Hour"/>

eli tarkoituksena oli kerätä jokaiselta päivältä tunneittain korkein lämpötila Celsius-asteina

Aggregate settings   Optimize   Inspect   Data preview ●

---

Output stream name \*  [Learn more](#)

Description 

Aggregating data by 'date' producing columns 'maxTempCelsius'

[Reset](#)

Incoming stream \*

**Group by**   **Aggregates**

Grouped by: date

Add   Clone   Delete   Open expression builder

<input type="checkbox"/> Column	Expression
<input type="checkbox"/> <input type="text" value="maxTempCelsius"/>	<input type="text" value="max(Body.celsius)"/>

ja tallentaa tiedot CosmosDB:hen.

Data Preview:ssä kaikki näyttäisi olevan kunnossa,

Aggregate settings   Optimize   Inspect   **Data preview**

Number of rows + **INSERT** 9 \* **UPDATE** 0

Refresh

Typecast

Modify

Map drifted

Statistics

Remove

<div>↑↓</div>	Date	<div><div></div></div>	<div>↑↓</div>	Hour	123	<div>↑↓</div>	maxTempCelsius
<div>+</div>	2023-04-29			15			24
<div>+</div>	2023-04-29			16			24
<div>+</div>	2023-04-29			20			23
<div>+</div>	2023-04-29			21			23
<div>+</div>	2023-04-29			22			23
<div>+</div>	2023-04-30			4			22
<div>+</div>	2023-04-30			5			22
<div>+</div>	2023-04-30			3			21
<div>+</div>	2023-04-29			17			22

mutta kun samoja tietoja mennään katsomaan CosmosDB:stä huomataan, että Date-sarakkeen tiedoille tapahtuu jotain omituista:

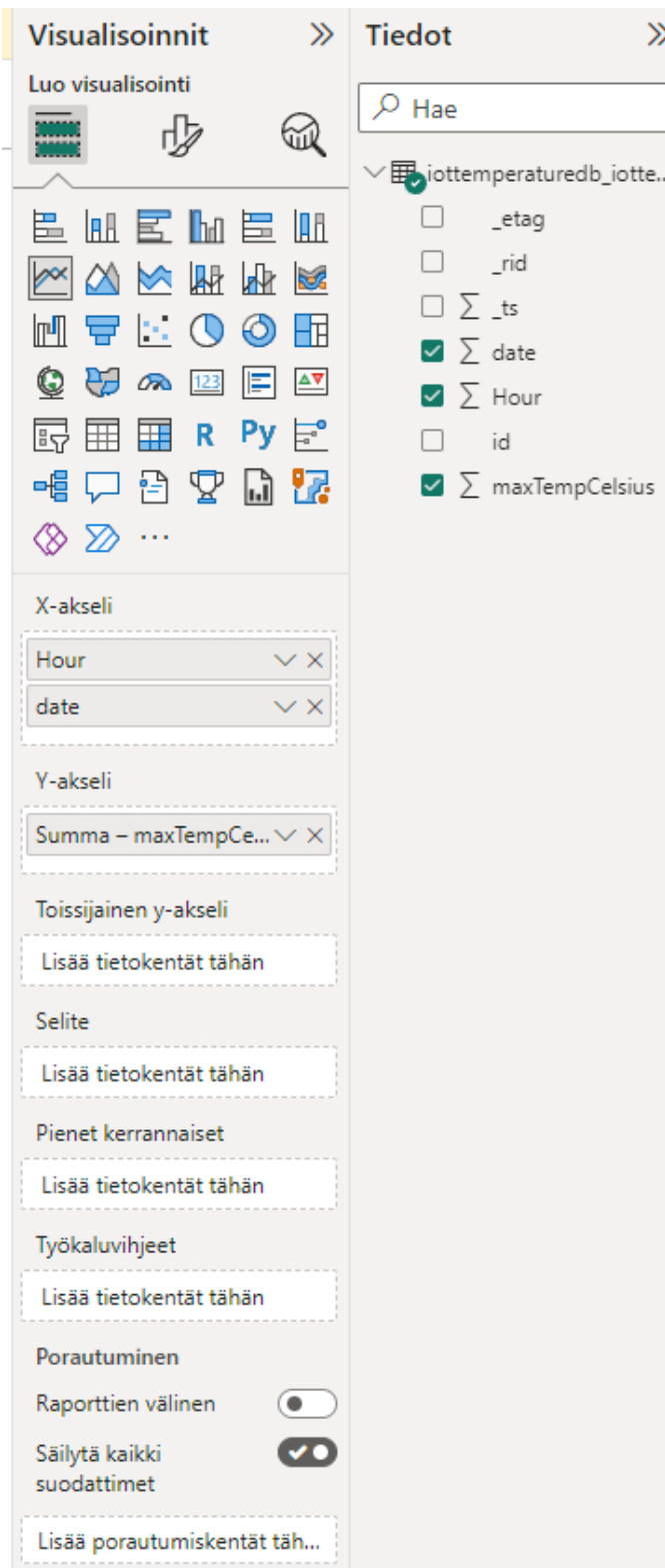
```

    "maxTempCelsius": 22,
    "Hour": 3,
    "id": "a9a64c27-6dda-4cf5-ac59-e3b2e88b780f",
    "Date": 1682812800000,
    "_rid": "1dswAMBFru8IAAAAAAAAAA==",
    "_self": "dbs/1dswAA==/colls/1dswAMBFru8=/docs/1dswAMBFru8IAAAAAAAAAA==/",
    "_etag": "\"00002925-0000-0c00-0000-644e8f280000\"",
    "_attachments": "attachments/",
    "_ts": 1682870056
  }

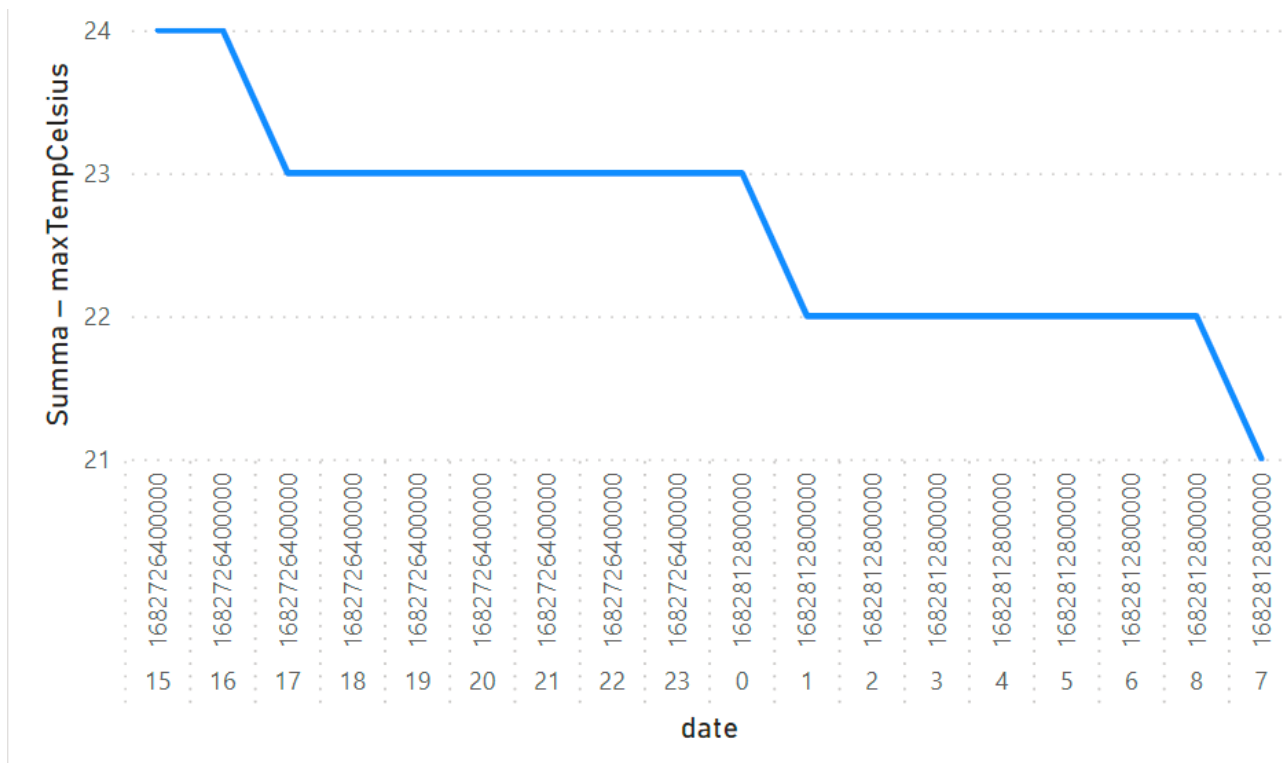
```

Tämä ongelma jäi valitettavasti ratkaisematta, mutta luultavasti päivämäärä pitäisi tallentaa jossain toisessa muodossa. Sama ongelma näytti luentotallenteen perusteella olevan tunnilla esitellyssä mallissakin.

Lopuksi kytkin CosmosDB:n tiedot PowerBI:hin seuraamalla näitä [ohjeita](#) (Microsoft: Visualize Azure Cosmos DB data using Power BI, 2022). Itse prosessi oli varsin yksinkertainen ja meni täysin ohjeiden mukaisesti. Yritin tehdä PowerBI:llä viivakaaviota päivittäisestä ja tunnittaisesta korkeimmasta lämpötilasta



ja PowerBI kyllä saa tiedot CosmosDB:stä ihan oikein, mutta sama Date-sarakkeen ongelma päättyy kuvaajaankin:



Kuvaajasta on kuitenkin hyvin nähtävillä, miten työhuoneen lämpötila alkaa varsin kuumasta 24 asteesta. Työhuoneessa sijaitsevat sekä omani että tyttökaverini pöytäkoneet ja sitä lämmittää vielä iltapäivän aurinko. Lämpötila alkaa sitten vähitellen laskea, kun koneet jäävät vähemmälle käytöllä ja aurinko lakkaa paistamasta.



## 4 POHDINTA

Itse projekti oli tosi mielenkiintoinen ja mielelläni olisin käyttänyt siihen enemmänkin aikaa. Varsinkin tuo Wi-Fi moduuli olisi ollut hauska saada toimimaan ja sen avulla olisi ollut mielenkiintoista kerätä dataa pidemmältäkin aikajaksolta, jolloin sen visualisointi PowerBI:llä olisi ollut vielä mielenkiintoisempaa. Toisaalta, jos olisin tajunnut luovuttaa sen suhteen jo aiemmin, niin olisin voinut enemmän keskittyä datan käsittelyyn Azuren puolella ja koettaa ratkaista tuon Date-ongelman.

## LÄHTEET

- GitHub: Adding DHT11 or DHT22 sensor support.* (ei pvm). Haettu 30. 4 2023 osoitteesta <https://github.com/rwaldron/johnny-five/issues/112>
- J5: Thermometer - TMP36.* (ei pvm). Haettu 30. 4 2023 osoitteesta <http://johnny-five.io/examples/temperature-tmp36/>
- Joy-IT: KY-015 COMBI-SENSOR (TEMPERATURE & HUMIDITY).* (ei pvm). Haettu 30. 4 2023 osoitteesta <https://sensorkit.joy-it.net/en/sensors/ky-015>
- Microsoft: Streaming IoT Telemetry to Azure Event Hub : Part 1.* (4. 1 2016). Haettu 30. 4 2023 osoitteesta <https://learn.microsoft.com/en-us/archive/blogs/israelo/streaming-iot-telemetry-to-azure-event-hub-part-1>
- Microsoft: Visualize Azure Cosmos DB data using Power BI.* (13. 11 2022). Haettu 30. 4 2023 osoitteesta <https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/powerbi-visualize>
- Thinglabs.io: Sending Telemetry to the Cloud.* (ei pvm). Haettu 30. 4 2023 osoitteesta <http://thinglabs.io/setup-arduino/>
- Wikipedia: ESP8266.* (29. 3 2023). Haettu 30. 4 2023 osoitteesta <https://en.wikipedia.org/wiki/ESP8266>