



Osaamista  
ja oivallusta  
tulevaisuuden  
tekemiseen

Tuomas Kekki

# Sääasemajärjestelmän suunnittelu ja toteutus

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Sähkö- ja automaatiotekniikan tutkinto-ohjelma

Insinöörityö

16.5.2021

Tekijä Otsikko	Tuomas Kekki Sääasemajärjestelmän suunnittelu ja toteutus
Sivumäärä Aika	41 sivua + 3 liitettä 16.5.2021
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	sähkö- ja automaatiotekniikka
Ammatillinen pääaine	automaatiotekniikka
Ohjaajat	lehtori Kai Virta
<p>Työn tavoitteena oli rakentaa säähavaintojärjestelmä, joka koostui aurinkoenergialla toimivasta IOT-sääasemasta, sääaseman tietokantapalvelimesta sekä havaintojen tarkasteluun tarkoitettuasiakasohjelmasta.</p> <p>Työhön kuuluivat sääaseman rakentaminen, aseman mikrokontrollerin ohjelmoiminen, tietokantapalvelimen asennukset ja kannan ja käyttäjien määrittäminen, asiakasohjelman käyttöliittymän suunnittelu ja ohjelmointi. Lisäksi toteutukseen kuuluivat mikrokontrollerin ja palvelimen välisten rajapintojen ohjelmointi PHP:llä. Tietokantapalvelimena toimi LAMP-palvelin, jossa yhdistyy Linux-käyttöjärjestelmä, Apache-palvelinsovellus, MySQL-tietokanta sekä PHP-ohjelmointikielellä toteutetut rajapinnat. Aurinkoenergialla toimiva sääasema lähettää GPRS-yhteyden avulla keräämänsä tiedot palvelimelle kutsumalla rajapintaa. Rajapinta siirtää tiedot tietokantaan. Asiakas ohjelmalla voidaan tarkastella tietokantaan vietyjä tietoja sekä määrittää tiedon keruu väli, jolla sääasema kerää dataa. Asiakasohjelman lähdekoodi löytyy githubista.</p> <p>Teoriaosuudessa käydään läpi säätiedon keräämiseen liittyviä seikkoja, joita tulee ottaa huomioon mittauspaikkaa valittaessa.</p> <p>Sääasemaa testattiin neljänä eri ajankohtana, joista yhdessä sääaseman toimintaedellytykset olivat siinä rajalla, että selviääkö se toiminnassa montaa päivää sen hetkisen akustonsa varassa. Kolme muuta suotuisampaa ajankohtaa sääasema pärjasi mainiosti.</p> <p>Työn tuloksena saatiin sääasemajärjestelmä, jonka aurinkoenergialla toimiva sääasema kykenee toimimaan suurimman osan vuodesta itsenäisesti eteläsuomen kaltaisissa sää- ja valoisuusolosuhteissa.</p>	
Avainsanat	Aurinkoenergia, sää-asema, IoT, LAMP

Author Title	Tuomas Kekki Design and implementation of a weather station system
Number of Pages Date	38 pages + 3 appendices 16 August 2021
Degree	Bachelor of Engineering
Degree Programme	Electrical and Automation Engineering
Professional Major	Automation Engineering
Instructors	Kai Virta, Senior Lecturer
<p>The objective of this thesis work was to build a weather observation system, which consisted of a solar-powered IoT weather station, a weather station database server and a client application for viewing gathered weather data.</p> <p>The work included building the weather station, programming weather station microcontroller, installing database server and configuring database and users for it. Designing and programming of user interface and client application were also included. Programming of application programming interfaces (API) between weather station and database was also part of work.</p> <p>The database server was a LAMP server, which combines a Linux operating system, an Apache server application, a MySQL database and interfaces implemented in the PHP programming language.</p> <p>A solar-powered weather station sends the data it collects via a GPRS connection to a server by calling the API. The API sends the data to the database. The client program can be used to view the data of the database and to specify the time interval at which the weather station collects data.</p> <p>The theoretical part reviews the issues related to the collection of weather data, which should be taken into account when choosing a site for weather station.</p> <p>The weather station was tested at four different test events. During one test period, the operating conditions of the weather station were on the verge of whether it would survive many days of operation on its current battery. The other three more favorable events the weather station did well.</p> <p>The result of the work is a weather station system whose solar-powered weather station is able to operate independently for most of the year in weather and light conditions of southern Finland.</p>	
Keywords	Solar-powered, weather station, IoT, LAMP

## Sisällys

### Lyhenteet

1	Johdanto	1
2	Sään seurannan perusteita	2
3	Säähavaintojärjestelmän osat	5
3.1	Sääasema	6
3.1.1	Akusto	6
3.1.2	Aurinkopaneeli ja lataussäädin	7
3.1.3	Arduino Uno	9
3.1.4	GPRS-shield	11
3.1.5	Kosteus- ja lämpötila-anturi	11
3.2	Tietokantapalvelin	12
3.2.1	Rasperry Pi	12
3.2.2	Apache	13
3.2.3	MySQL	13
3.2.4	PHP	14
3.3	Käyttäjän PC	14
3.3.1	Tietokannan asiakasovellus	14
3.4	Reititin	14
4	Järjestelmän toteuttaminen	15
4.1	Sääasema	15
4.1.1	Sähköjärjestelmä	15
4.1.2	Mikrokontrolleri	16
4.1.3	GSM-lisäkortti	18
4.1.4	Sensorit	20
4.2	Tietokantapalvelin	26
4.3	Tietokanta	29
4.4	Käyttöliittymä/Client	32
5	Tuloksien tarkastelu	34

6	Yhteenveto	36
	Lähteet	38
	Liitteet	
	Liite 1. PHP-lähdekoodit	
	Liite 2. Arduino lähdekoodi	
	Liite 2. Sääaseman keräämä data	

## Lyhenteet

ORM	Object-relational mapping. Oliomallin mukaisen esityksen kuvaus relaatiomallin mukaiseksi esitykseksi.
UI	User Interface. Laitteen/ohjelman käyttöliittymä.
AGM	Akkutyyppejä. Lyijyhappoakku, jossa elektrolyytti on imeytettynä lasivillaserottimeen eikä ole vapaana nesteenä akun sisällä.
SQL	Structured Query Language. Standardoitu kyselykieli, jolla suoritetaan erilaisia hakuja, muutoksia tai lisäyksiä relaatiotietokantoihin.
GSM	Global System for mobile Communications. Digitaalinen matkapuhelinjärjestelmä.
IDE	Integrated development environment. Ohjelmointiympäristö.
GPRS	General Packet Radio Service. GSM-verkossa toimiva pakettikytkentäinen tiedonsiirtopalvelu.
JSON	Java Script Object Notation. Avoimen standardin tiedostomuoto tiedon välitykseen.
Linux	Avoimen lähdekoodin käyttöjärjestelmä tietokoneille.
Apache	Apache HTTP-server on avoimeen lähdekoodiin perustuva HTTP-palvelinohjelma.
PHP	Hypertext Preprocessor. Ohjelmointikieli jota käytetään erityisesti web-palvelinympäristössä.
MySQL	Relaatiotietokantaohjelmisto.
LAMP	Lyhenne sanoista Linux, Apache, MySQL ja PHP.

RPI

Rasperry Pi-mikrotietokone.

## 1 Johdanto

Insinööriyön tavoitteena oli toteuttaa säähavaintojärjestelmä, johon kuuluva sääasema saisi tarvitsemansa energian aurinkosähköstä. Työlle ei ollut toimeksiantajaa, vaan insinööriyön ”tilaajana” ja ”toimittajana” toimi opiskelija. Järjestelmä koostui säätietoa keräävästä sääasemasta, kerättyä tietoa säilyttävästä tietokantapalvelimesta, tietokoneesta, jolla säädataa voidaan tarkastella sekä lähiverkosta ja reitittimestä, joka yhdistää laitteet. Sääasema keräsi ilman lämpötilan ja suhteellisen kosteuden sille annetuin aikaväleihin. Lisäksi akun jännitettä mitattiin samassa yhteydessä ja nämä tiedot toimitettiin internetin kautta kotiverkossa sijaitsevalle tietokantapalvelimelle. Palvelimena toimi kotiverkkoon liitetty Rasperry Pi -mikrotietokone, jolle oli asennettu MySQL-tietokantapalvelin. Tietokantaa voitiin katsella PC-sovelluksella, joka oli toteutettu Windows 10 -käyttöjärjestelmälle. Tavoitteena oli, että sääaseman voisi jättää pidemmäksi aikaa keräämään dataa paikkaan, jossa sille ei ole saatavilla sähköä verkosta. Työtä varten tarvittavista komponenteista valmiina oli Arduino, Rasperry Pi ja GPRS-kortti. Loput osat hankittiin sitä mukaa, kun työ edistyi.

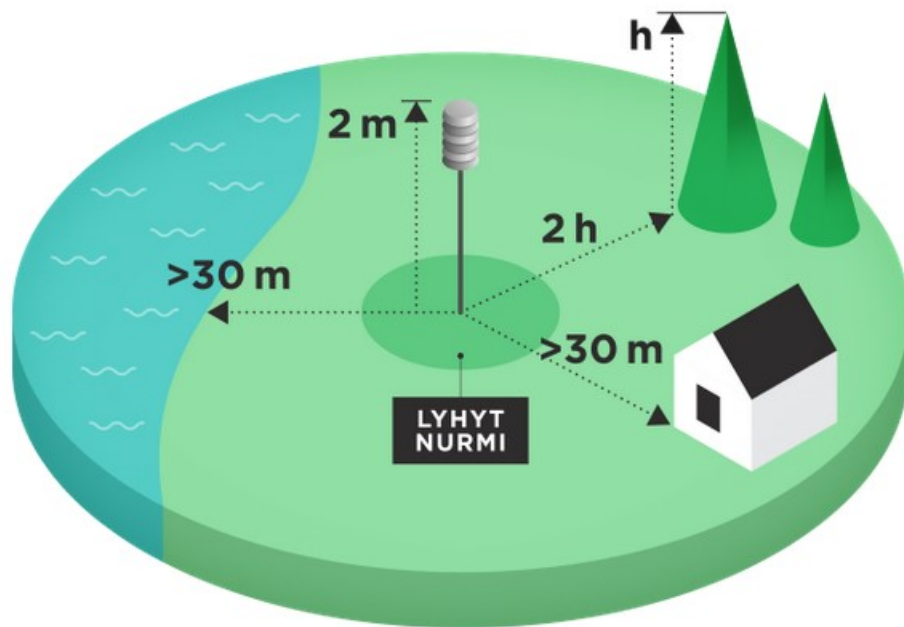
Työ aloitettiin tietokantajärjestelmän asennuksilla ja kannan rakenteen suunnittelulla ja konfiguroinnilla. Sen jälkeen valittiin ja hankittiin komponentit sääaseman sähköjärjestelmään ja mikrokontrollerille. Asiakasohjelman ja mikrokontrollerin koodin kehitystä tehtiin rinnakkain projektin aikana.

Työn aikana pääsin tekemään laitteen teknisen toteutuksen, ohjelmistoympäristön pysyttämisen, asiakasohjelman ja rajapintojen sekä mikrokontrollerin ohjelmoinnin. Työ sisälsi myös hieman sähköistä suunnittelua laitteen sähköjärjestelmän osalta sekä säähavaintoanturin valmistuksen ja kaikkien edellä mainittujen komponenttien yhteensovittamisen.



## 2 Sään seurannan perusteita

Kuka tahansa voi alkaa seuraamaan säätä ja tuottamaan säädataa, vaikka omalla kotipihallaan. Ilmatieteenlaitoksen www-sivuilta löytyy tietoa siitä, mitä kannattaa ottaa huomioon, jos aikoo perustaa oman sääaseman. Sivuille on koottu ohjeita siitä, kuinka voidaan tehdä säähavaintoja ja mittauksia, jotka täyttävät tietyt edustavuusvaatimukset. Edustavuusvaatimukset perustuvat Maailman ilmatieteen järjestön (WMO) kokoamaan liitteeseen. Tällä pyritään siihen, että eri puolilla maailmaa tehdyt säähavainnot olisivat vertailukelpoisia. Huomiota kiinnitetään mittalaitteiden sijoitteluun, jotta mittalaitteisiin vaikuttavat häiriötekijät saadaan minimoitua. Ideaalitulanteessa sääasema on sijoitettu tasaiselle avaralle alueelle, jonka läheisyydessä ei ole asfalttia, betonia tai kivikkoa. Tällöin nämä toimivat lämmön lähteinä ja vaikuttavat siten mittaustuloksiin. Myös vesialueet vaikuttavat tuloksiin siten, että keväisin vesistöllä on viilentävä vaikutus ja syksyisin vaikutus on lämmittävä. Myös rakennukset tai metsä haittaavat sääaseman toimintaa vaikuttamalla tuulen suuntaan ja nopeuteen. Mikäli sääaseman läheisyydessä on kuitenkin esteitä, tulisi niiden etäisyyden olla vähintään kaksi kertaa niiden korkeus. Ilman lämpötilan ja kosteuden mittauskorkeus on 2 metriä. Lämpömittari tulisi suojata säteilysuojalla, jonka on kuitenkin mahdollistettava ilman sekoittuminen suojan sisällä. Suojalla vältetään auringon valon ja sateen aiheuttamia mittavirheitä. Kosteusmittarin sijoittelussa pätevät samat ohjeet. (1.)



Kuva1. Sääaseman sijoittelussa huomioitavia etäisyyksiä (1).

Tuulta mitattaessa Ilmatieteenlaitos pyrkii mittaamaan 10 metriä ympäristön esteiden yläpuolelta. Kotioloissa talon katolta voi löytyä hyvä paikka. Katolle asennettaessa mittari olisi hyvä saada ainakin muutaman metrin katon yläpuolelle, koska räystäät aiheuttavat tuulen pyörteisyyttä. Tuulimittari täytyy suunnata todelliseen pohjoiseen. Mikäli suunnauksessa käytetään kompassia tai GPS:ää, on suunnauksessa huomioitava eranto, joka on kulma maantieteellisen ja magneettisen pohjoisnavan välillä. Myös mittarin vaakatason suuntaukseen on kiinnitettävä huomiota, jotta ei aiheutuisi mittavirheitä. Ilmanpaine-mittarin sijoituksessa olisi huolehdittava, että se pääsee suojaan vedolta, auringolta ja lämmöltä. Lämpötilan tulisi olla tasainen, minkä vuoksi ilmanpaineanturit suositellaan asennettavaksi sisätilaan. (1.)

Mikäli sääasemaan kuuluu sademäärän tai lumen korkeuden mittaus, niin niitä ajatellen olisi hyvä, jos mittarin ympärillä olisi pensaikkoa, joka hillitsisi tuulen nopeutta. Jo 2 metriä sekunnissa liikkuva tuuli aiheuttaa sen, että osa sateesta menee mittarin ohi. Sade-

mittarin tulee olla vakaatasossa. Lumen syvyyttä voidaan mitata kepillä, johon on merkitty senttimetrin välein merkit. Paras paikka lumen mittaukselle olisi kohta, johon tuuli ei vaikuttaisi. (1.)



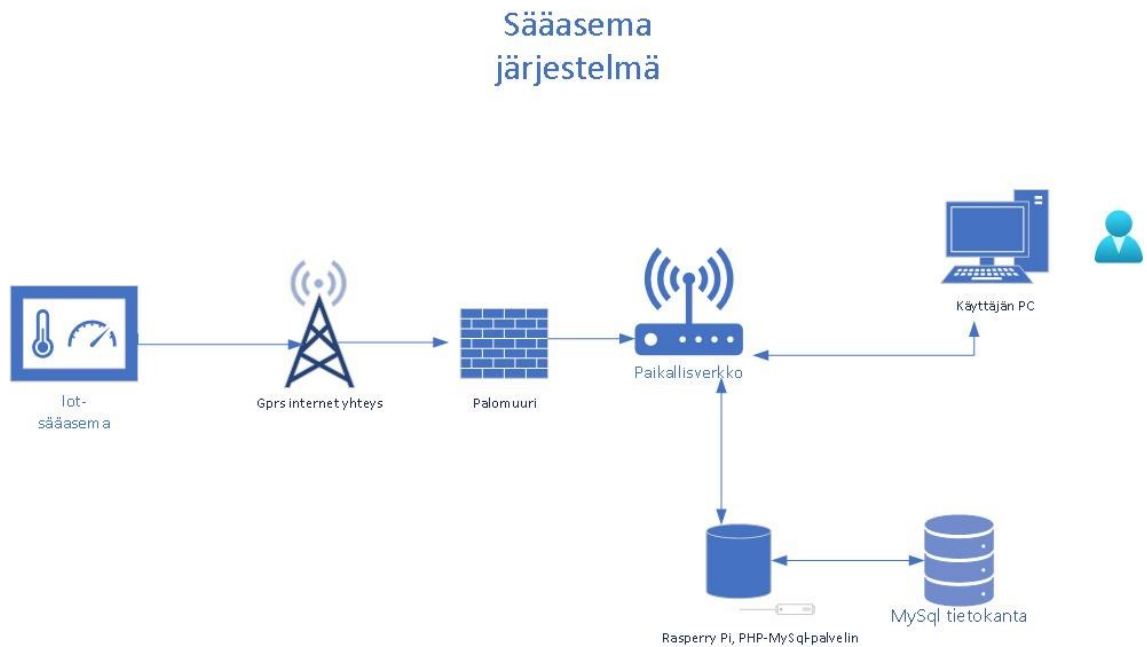
Kuva 2. Vaisalan siirrettävä sääasema MAWS201 (2).

Sääasema on laitteisto, jonka avulla voidaan mitata säähän liittyviä suureita kuten esimerkiksi lämpötilaa, ilmankosteutta, tuulen suuntaa ja nopeutta sekä esimerkiksi sademäärää. Opinnäytteen sääaseman rakenteen suunnitteluun hain vaikutteita Vaisalan MAWS201-sääasemasta. Insinööriyön sääaseman mittaamiksi suureiksi valittiin ilman lämpötila ja suhteellinen kosteus.

### 3 Säähavaintojärjestelmän osat

Säähavaintojärjestelmä koostuu seuraavista osista, joita tarkastellaan lähemmin edempänä:

IOT-Sääasema, paikallisverkko + palomuuuri, tietokantapalvelin ja käyttäjä-PC.



Kuva 3. Järjestelmän eri osat.

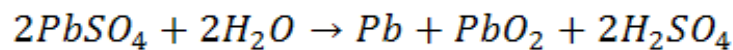
### 3.1 Sääasema

Järjestelmän sääasemaan kuuluvat seuraavat osat: akusto, aurinkopaneeli ja latauspiiri, Arduino-mikrokontrolleri, GPRS-kortti sekä kosteus- ja lämpötila-anturi.

#### 3.1.1 Akusto

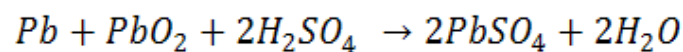
Akku on laite, johon voidaan varastoida sähköenergiaa kemialliseksi energiaksi. 12 voltin lyijyakku koostuu kuudesta sarjaan kytketystä kennosta, jotka kukin koostuvat kahdesta elektrodista. Positiivista elektrodia kutsutaan katodiksi ja negatiivista anodiksi. Lyijyakussa molemmat elektrodit ovat lyijyä, ja niiden välissä on elektrolyyttinä rikkihappoa, joka kuljettaa sähkövarauksia elektrodilta toiselle.

Lyijyakun kemiallinen kokonaisreaktio purettaessa akkua:



Kuva 4. Reaktio purkautuessa (3).

Lyijyakun kemiallinen kokonaisreaktio ladattaessa akkua:



Kuva 5. Reaktio ladattaessa (3)

Akun kapasiteettia merkitään kirjaimella C. Kapasiteetillä tarkoitetaan akun varauskykyä, joka ilmoitetaan ampeeritunteina (Ah). Insinööriyössä käytetyn akun tapauksessa yhtä akkua voitaisiin kuormittaa 1 ampeerin virralla 9 tuntia. Akkuun mahtuva energiamäärä voidaan laskea kertomalla akun kapasiteetti akun jännitteellä.  $9 \text{ Ah} \times 12 \text{ V} = 108 \text{ Wh}$ . (4.)

Sääaseman akustona toimii 2 kpl 12 voltin ja 9 ampeeritunnin AGM-lyijyakkua. Akut ovat tyypiltään AGM-akkuja, joissa happo on imeytettynä lasikuitumattoihin. Se on huoltovapaa ja kestää hyvin kylmiä olosuhteita. AGM-akkua voidaan ladata 14 voltin jännitteellä, mutta sen ylläpitojännitteen tulisi olla maksimissaan 13,8 voltia. (5.)



Kuva 6. Akku (6)

### 3.1.2 Aurinkopaneeli ja lataussäädin

Akustoa ladataan 12 voltin ja 15 watin SolarXon ES-P-aurinkopaneelilla. Akun latautumista säädellään Steca-lataussäätimellä. Paneelin mitat ovat 350 x 350 x 25 mm ja paino 1,5 kg.

Lataussäätien tehtävä on estää akun yllilataantuminen ja toisaalta akun liiallinen purkaantuminen. Säädin huolehtii että akkuja ladataan sille sopivalla latausjännitteellä. Säädintä voidaan käyttää myös ohjaamaan 12/24 V:n käyttölaitteita. Lataussäädin on useimmiten joko PWM-peritaattella toimiva (Pulse Width Modulation) tai MPPT-periaatteella (Maximum Power Point Tracking) toimiva. PWM-säädin katkoo ja kytkee aurinkopaneelilta saatavaa virtaa sopivassa suhteessa ja alentaa näin akun latausjännitettä. MPPT-säädin seuraa jatkuvasti paneelilta saatavaa jännitettä ja kohdistaa sen perusteella paneelille kuormitusta. Paneelin jännitteen noustessa säädin nostaa latausjännitettä. Jos teho kasvaa, nostaa säädin lisää jännitettä. Mikäli teho laskee jännitteen noston seurauksena, osaa säädin laskea latausjännitettä alaspäin. Tällä tavoitellaan maksimaalista tehoa. (4.)

Halvempien PWM-säätimien hyötysuhde on noin 50 prosentista 60 prosenttiin, kun taas hiukan kalliimmilla MPPT-säätimillä voidaan päästä 95-98 %:n hyötysuhteeseen. (7)



Kuva 7. PWM-Lataussäädin.

Aurinkosähköpaneeli on laite, joka muuttaa auringosta tulevaa sähkömagneettista säteilyä sähköenergiaksi. Aurinkopaneelien kehitys on aloitettu 50-luvulla avaruustekniikan tarpeisiin. Aurinkopaneeli koostuu useasta yhteenkytetyistä aurinkokennosta. Aurinkokennon raakaaineena on yleisimmin käytetty piitä. Auringon säteily synnyttää kennoissa sähkövirtaa, jota voidaan hyödyntää erilaisiin tarpeisiin. (8.)

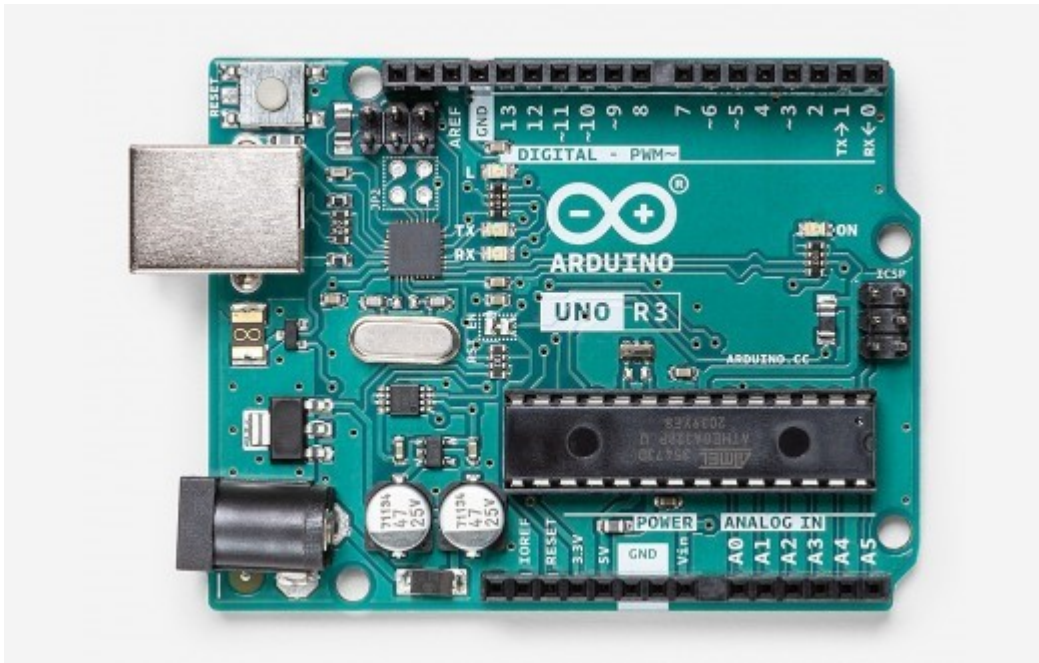


Kuva 8. Aurinkopaneeli.

### 3.1.3 Arduino Uno

Mikro-ohjaimet eli mikrokontrollerit ovat elektronisia laitteita, joihin voidaan ohjelmoida toimintoja. Mikrokontrolleri on ikään kuin yksinkertainen tietokone, joka suorittaa sille ohjelmoitua ohjelmaa. Se on tarkoitettu osaksi sulautettua järjestelmää pikemmin kuin itsenäiseksi laitteeksi.





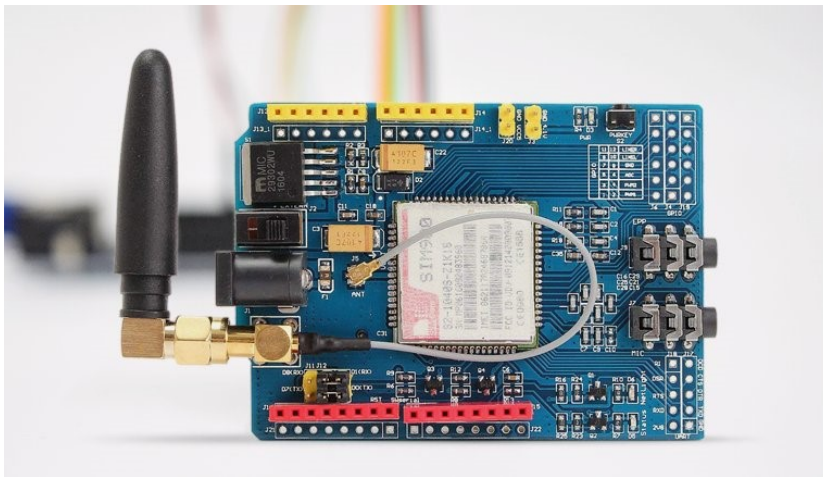
Kuva 9: Arduino Uno mikrokontrolleri. (10)

Arduino Uno on harrastajien parissa suosittu mikrokontrolleri, jota on alun perin lähdetty kehittämään edulliseksi ja helposti lähestyttäväksi tavaksi aloittelijoille tutustua ohjelmointiin logiikoihin. Tähän työhön mikrokontrolleriksi valikoitui Arduino Uno, koska sellainen oli jo valmiiksi saatavilla. Uno sisältää digitaalisia sisään- ja ulostuloja sekä analogisia sisääntuloja, joiden toimintaa voidaan ohjata sille ohjelmoidulla ohjelmalla. Uno perustuu ATmega328P-piiriin. Siinä on 14 digitaalista in/out-pinniä, joista kuusi voi toimia PWM (Pulse Width Modulation) ulostuloina. Lisäksi siinä on kuusi analogista sisääntuloa. Arduinoa voidaan ohjelmoida C- ja C++-kielillä. (9.)

Ohjelmointiin on tarjolla Arduinon oma IDE, jolla pääsee helposti luomaan omia ohjelmia. Arduino.cc-sivustolta saa myös paljon tietoa siitä, mitä ja miten Arduinoa voidaan ohjelmoida sekä keskusteluketjuista apua ohjelmointiongelmien. Arduinolle on myös saatavilla useita erilaisia lisäkortteja, kuten esimerkiksi moottorinohjauskortteja tai relekortteja. Tässä opinnäytetyössä käytettiin GSM-lisäkorttia, jolla voidaan tehdä GPRS-tiedonsiirtoa. (9.)

### 3.1.4 GPRS-shield

GPRS-shield vastaa GSM-verkon yli tapahtuvasta tiedonsiirrosta. Shield-sanalla tarkoitetaan Arduinolle suunniteltuja lisäkortteja, joita voidaan helposti kytkeä Arduinoon pi-noamalla shield-kortteja Arduinon päälle. GPRS-kortilla voidaan tuoda Arduinolle GSM-tekniikkaan perustuvia toimintoja. Sen avulla Arduino voi lähettää tai vastaanottaa tekstiviestejä tai sen avulla voitaisiin tehdä Arduinosta vaikkapa puhelin. Tässä työssä sitä käytetään GPRS-tiedonsiirtoon.



Kuva 10. GPRS-kortti

### 3.1.5 Kosteus- ja lämpötila-anturi

Kosteuden ja lämpötilan mittaamiseksi sääasemaan on valittu DHT22-kosteusanturi. Se kykenee mittaamaan  $-40$  Celsius-asteesta  $+80$  asteeseen saakka  $\pm 0,5$  asteen tarkkuudella. Ilman suhteellisen kosteuden se mittaa 5 prosentista 99 prosenttiin saakka  $\pm 2$  prosentin tarkkuudella. (11.)



Kuva 11. DHT22-sensori (11)

Item	Parameter	Min	Norm	Max	Unit
VCC	-	3.3	-	6	V
Measuring Current Supply	-	1	-	1.5	mA
Standby Current Supply	-	40	-	50	uA
Measuring range	Humidity	5%	-	99%	RH
	Temperature	-40	-	80	°C
Accuracy	Humidity	-	-	±2%	RH
	Temperature	-	-	±0.5	°C
Resolution	Humidity	-	-	0.1%	RH
	Temperature	-	-	0.1	°C
Repeatability	Humidity	-	-	±0.3%	RH
	Temperature	-	-	±0.2	°C
Long-term Stability	-	-	-	±0.5%	RH/year
Signal Collecting Period	-	-	2	-	S
Respond Time	1/e(63%)	6	-	20	S

Kuva 12. DHT22-sensorin arvoja (11)

### 3.2 Tietokantapalvelin

Tietokantapalvelin koostuu Raspberry Pi -tietokoneelle asennetusta Linux-pohjaisesta käyttöjärjestelmästä, Apache-palvelinohjelmasta, MySQL-tietokantaohjelmistosta.

#### 3.2.1 Raspberry Pi

Raspberry Pi on mikrotietokone, johon voidaan asentaa käyttöjärjestelmäksi erilaisia Linux-käyttöjärjestelmiä. Tässä työssä käytettiin palvelimena Raspberry Pi 3B (tästä eteenpäin RPI) -mikrotietokonetta. RPI 3B -mallissa on 64-bittinen neliydinprosessori, joka toimii 1,4 Gigahertsin kellotaajuudella. Muistia prosessorissa on 1 Gigabitti. Siinä on tuki

langattoman lähiverkon käyttöön, jolla se on myös tässä työssä liitetty kotiverkkoon. RPI:stä löytyy myös 4 USB-porttia, HDMI-portti ja 40 ohjelmoitavaa pinniä. (12.)

### 3.2.2 Apache

Apache on avoimen lähdekoodin HTTP- / web-palvelinohjelma. Se on saataville useille eri alustoille ja se on lisäksi ilmainen (13). Se on vuonna 2020 ollut käytetyin palvelinohjelmisto aktiivisina olleille sivustoille. Sen markkinaosuus on tuolloin ollut noin 27 %. Vertailuna Microsoftin osuus oli noin 4,5 % ja Googlen noin 9,8 %. (14.)

Tässä työssä Apache toimii palvelimena tietokannan, käyttöliittymän ja mikrokontrollerin välillä.

### 3.2.3 MySQL

Tietokanta on kokoelma loogisesti yhteenkuuluvaa tietoa, joka on luotu jotain tiettyä tarvetta varten. Tietokannalle tyypillistä on, että sen käyttäjät tunnetaan. Näin voidaan määrittellä oikeudet käyttäjille, jotka määrittävät, mitä kukin käyttäjä voi tehdä tietokannassa. Tietokantaa hallinnoidaan tietokannanhallintajärjestelmällä (TKHJ, englanniksi Database management system, DBMS). TKHJ:n avulla käyttäjä voi tehdä tietokannalle kyselyjä SQL-kielillä (Structured query language) lausekkeilla. SQL-kieli on kehitetty käyttäjän ja TKHJ:n väliseen kommunikointiin. SQL-kieltä käytetään monissa tietokantajärjestelmissä, joskin ne poikkeavat hieman toisistaan. Relaatiotietokannan perusyksikkö on yksi taulu eli relaatio. Relaatio voi olla osa suurempaa kokonaisuutta, joka koostuu useista relaatioista. Taulun sarakkeiden otsikot ovat attribuutteja ja vaakarivit ovat tietueita. Tietueet koostuvat soluista, jotka sisältävät tallennetun tiedon. Jotta tietoa voidaan yhdistää relaatiotietokannassa, tulee jokaisella taululla olla määritettynä pääavain, joka on jokin taulun attribuutti. Pääavain yksilöi tietueen. (15.)

MySQL Community edition on avoimen lähdekoodin relaatiotietokantaohjelmisto. Se valittiin tähän työhön, koska sitä käytetään LAMP-palvelimissa tietokantaohjelmistona.

### 3.2.4 PHP

PHP on yleiskäyttöinen ohjelmointikieli, jota on käytetty laajalti web-kehityksessä. Sitä voidaan sisällyttää HTML-koodiin ja saada aikaan dynaamista sisältöä. PHP-skriptit tekevät palvelimen päässä erilaisia asioita, eivätkä ne näy asiakkaan suuntaan. Tässä työssä PHP:tä on täytetty ns. kuuntelijan toteuttamisen. Kuuntelija ottaa vastaan HTTP-pyyntön ja käsittelee sen. Jos pyyntö on validi, lähettää skripti SQL-kyselyn tietokantaan. Myös sääaseman tiedonkeruuvälin määrittämiseen ja lukemiseen on tehty rajapinnat PHP:llä. (16.)

## 3.3 Käyttäjän PC

Käyttöliittymää käytetään tässä työssä Lenovo Legion -merkkiseltä kannettavalta tietokoneelta, jossa on asennettuna Windows 10 -käyttöjärjestelmä.

### 3.3.1 Tietokannan asiakasovellus

Tietokannan tarkasteluun on tässä työssä luotu Winformsilla yksinkertainen graafinen käyttöliittymäsovellus, jolla voidaan katsoa tietokantaan tallennettuja tietoja määritellyltä aikaväliltä. Sovelluksella voidaan myös tarkistaa ja asettaa sääaseman tiedonkeruun aikaväli. Tiedot haetaan MySQL-tietokannasta ja esitetään taulukoituna sovelluksessa. Taulukosta voidaan myös piirtää pylväskaaviot lämpötilasta ja ilmankosteudesta.

## 3.4 Reititin

Reitittimenä käytettiin Telia hybrid router Fast 5370 Air -nimistä laitetta. Kyseessä on hybridireititin, joka hyödyntää sekä 4G- että ADSL-tekniikkaa. ADSL tarjoaa alhaisemman mutta vakaan tiedonsiirron, jota täydennetään nopeammalla mutta tiedonsiirtonopeudeltaan vaihtelevalla 4G-yhteydellä.

## 4 Järjestelmän toteuttaminen

### 4.1 Sääasema

Sääaseman tehtävä on kerätä säädataa sille määritellyin aikaväleihin sekä toimittaa data edelleen tietokantapalvelimelle. Liikenne sääaseman ja palvelimen välillä on kaksisuuntaista. Kaksisuuntaisuudella tarkoitetaan tässä sitä, että samalla, kun sääasema ottaa yhteyttä palvelimelle ja lähettää keräämänsä tiedot, käy se lukemassa tiedon käynnistymisvälistä palvelimen ylläpitämästä rajapinnasta. Sallituiksi aikaintervaleiksi on määritetty 5 - 240 minuuttia. Intervalliasetus toimii siten, että jos ajaksi on asetettu 30 tai 60 minuuttia, tahdistaa sääasema päivitysvälin siten, että se päivittää tiedot joko tasatunnein tai tasatunteihin nähden puolen tunnin välein. Sääaseman mikrokontrollerina toimii Arduino Uno.

Itse sääaseman rakenne voidaan jakaa neljään osaan: sähköjärjestelmään, mikrokontrolleriin, GSM-korttiin ja sensoreihin.

#### 4.1.1 Sähköjärjestelmä

Aurinkosähköjärjestelmä koostuu aurinkopaneelistä ja akustosta sekä lataussäätimestä, joka huolehtii akun latauksesta. Järjestelmäksi valikoitui 12 voltin järjestelmä helpon saatavuuden ja yleisyytensä vuoksi. Aurinkopaneelin teho on 15 wattia ja yhden akun kapasiteetti 9 Ah / 20 h. Lataussäädin on rautakaupasta saatava, kuluttajakäyttöön tarkoitettu lataussäädin.

Järjestelmän vaatimukset sähkönsaannin osalta ovat seuraavanlaiset: Arduino-mikrokontrolleri tarvitsee 5 tai 9 voltia (V) tasajännitettä (DC) ja virrantarve voi olla maksimissaan 0,5 ampeeria. GSM-lisäkortti tarvitsee 5 voltia tasajännitettä ja lisäksi sen hetkellinen huippuvirrantarve voi olla 2 A.

Arduino-mikrokontrollerille voidaan syöttää 9 voltia dc-liittimen kautta tai tuoda suoraan 5 voltia 5 V:n pinniin. Jos 5 voltia tuodaan suoraan 5 V:n pinniin, on varmistettava, että syöttävä jännite ei kasva yli 5 voltin, koska tämä voi aiheuttaa vaurion mikrokontrollerille. Sääasemassa 5 voltin käyttöjännitettä tuotetaan kahdella eri menetelmällä. GSM-kortille

5 voltia tuotetaan hakkurityyppisellä DC-DC-muuntimella, joka alentaa 12 voltia 5 volttiin. Hakkurin etuna on, että se muuntaa jännitteen hyvällä hyötysuhteella ja kykenee tuottamaan kortin vaatiman hetkellisen 2 ampeerin virran ongelmitta (17).

Hakkuritehonlähde kuitenkin tuottaa piiriin niin paljon sähköistä häiriötä, että sitä käytetään vain GSM-kortin virransyöttöön silloin, kun korttia käytetään tietojen lähettämiseen. Muun aikaa käytetään 5 voltin regulaattoria tuottamaan mahdollisimman häiriötöntä 5 voltin käyttöjännitettä.

Virrankulutuksen minimoimiseksi Arduino tarjoaa erilaisia virransäästönsä tasoja. Tässä työssä käytetään Low-power-kirjastoa, jolla aktivoidaan mikrokontrollerista power down -virransäästötila. Arduino viettää virransäästötilassa sille määritellyn ajan, kunnes se taas suorittaa mittaukset ja datan siirron. Koko järjestelmän lepotilassa kuluttama virta oli yleismittarilla mitattuna noin 62 mA. Yksistään työssä käytetty lataussäädin kuluttaa jatkuvasti noin 15 mA, jolloin Arduinolle ja sen käyttämälle 5 voltin jänniteregulaattorille jää 47 mA. Alun perin koko 5 voltin järjestelmän jännite oli tarkoitus tuottaa DC-DC-jännitteenalentajalla. Ongelmaksi muodostui kuitenkin sen tuottamat sähköiset häiriöt mikrokontrollerille. Myös lepotilan virran kulutus oli kyseisessä kappaleessa lähes 50 mA, joka oli aivan liian suuri tässä käyttötarkoituksessa. Tämän vuoksi DC-DC-muunninta käytettiin ainoastaan silloin, kun GSM-korttiin tarvittiin virtaa. Kortti kytketään mosfet-transistorilla päälle vain siksi aikaa, kun kortti lähettää datan palvelimelle. Käyttöjännite Arduinolle tuotettiin 5 voltin regulaattorilla. Tällä ratkaisulla jännitteen mittaaminen onnistui ilman häiriöitä. Akuston jännitteenmittauksen aikana järjestelmä irrottaa myös aurinkopaneelilataussäätimen akusta. Tällä pyritään saavuttamaan realistisempi jännitelukema akulta, kun sitä ei samalla ladata. Akun varaustilan mittaaminen on hyvin haastava asia varsinkin laitteissa, joissa akku on käytännössä jatkuvassa kuormituksessa.

#### 4.1.2 Mikrokontrolleri

Arduinon ohjelmoiminen tapahtuu tietokoneelle asennetussa Arduino-ohjelmointiympäristössä. Ohjelmointiympäristöllä voidaan ohjelmoida eri Arduino-kortteja, joissa on eroja esimerkiksi liitännöiden määrässä ja käyttöjännitteissä. Arduino liitetään USB-yhteydellä tietokoneeseen ja valitaan ohjelmointiympäristössä oikean tyyppinen Arduino-kortti, jolle ohjelma aiotaan ladata. Ohjelma ladataan Flash-muistiin, jossa se säilyy, vaikka kortilta

katkaistaisiin virta. Arduinon ohjelmointikieli on teknisesti kokoelma yksinkertaisia C++-kielen funktioita. Kirjastot Arduinolle on toteutettu C++-kielen luokkina. Ohjelma koostuu ylimmällä tasolla 2 funktiosta. Nämä ovat Setup ja Loop. Setup-funktio ajetaan käynnistymisen yhteydessä ja siinä voidaan ajaa kertaluontoisesti ajettava koodi ja alustaa muuttujia sekä määrittää lähtöjä tai tuloja. Loop-funktio on päättymätön kierto, jossa funktion sisältö ajetaan uudelleen kerta toisensa perään. Mahdolliset viittaukset ulkoisiin kirjastoihin sekä globaalien muuttujien esittely tehdään ennen Setup-funktiota aivan koodin alkuun. Ohjelma tallennetaan Arduinon flash-muistiin, jonka koko Unossa on 32 kilobittiä (kB). Väliaikaisen tiedon taltioimiseen käytetään SRAM-muistia, jonka koko on Unossa 2 kB. SRAM-muisti pyyhkiytyy virran katketessa. (9, s.77-90.)

Luodun ohjelman sykli on seuraavanlainen:

Mittaa lämpötila ja kosteus -> kytke lataussäädin irti akusta -> Mittaa akun jännite -> kytke lataussäädin takaisin akkuun -> Muodosta lähetettävä JSON teksti -> Kytke virta GSM-korttiin -> Yhdistä kortti ja lähetä JSON palvelimelle ->Hae päivitysvälin asetus palvelimelta -> Hae paikallisaika verkosta -> Kytke virta pois GSM-kortilta -> Laske odotusaika seuraavaan käynnistymiseen -> Siirry virransäästötilaan -> odota määritetty aika -> Aloita sykli alusta.

Arduino kerää mittaustiedot muuttujiin ja muodostaa niistä palvelimelle lähetettävän JSON-muotoillun tekstin.

```
{apikey:*****, sensor:temp_hum, location:outside, temperature:temp, humidity:humi, battery:batt}
```

JSON-muotoinen viesti alkaa ja päättyy aaltosulkuihin. Yksinkertaisimmillaan yllä oleva viesti voisi olla {apikey:\*\*\*\*\*}, jossa apikey on muuttujan nimi ja \*\*\*\*\* on muuttujan arvo. Jos arvoja on useampi, sijoitetaan ne aaltosulkujen sisään ensimmäisen parin jatkoksi pilkulla erottaen. JSON-formaatin etu on sen helppo luettavuus ihmisen kannalta, ja sitä on helppo parsia takaisin erillisiksi nimi / arvo pareiksi.

Mikrokontrolleri siis toteuttaa ikuista ohjelmakiertoa, ja välissä se viettää määritellyn ajan virransäästötilassa. Mikrokontrollerin virrankulutukseen voi vaikuttaa monella tapaa. Jos mikrokontrolleria käytetään esimerkiksi matalammalla kellotaajuudella tai matalammalla



3,3 voltin jännitteellä päästään alhaisempaan energiankulutukseen. Myös käytettävällä mikrokontrollerilla on suuri merkitys siihen, kuinka pienellä virralla voidaan selvitä.

#### 4.1.3 GSM-lisäkortti

Tässä työssä käytettiin GSM-lisäkorttia SIM900 Quad Band GSM GPRS Shield.

Toimiakseen GSM-verkossa, tulee kortille hankkia teleoperaattorin sim-kortti, joka mahdollistaa yhdistämisen verkkoon sekä datan siirron internetin yli. Tässä työssä käyttötaroitukseen riittää edullinen Prepaid-kortti, jonka kautta yhteys palvelimeen voidaan muodostaa. Kortti tarvitsee toimiakseen 5 voltin jännitteen sekä melko suuren lyhytaikaisen virran. Kun GPRS-yhteyttä luodaan verkkoon, on kortin virrantarve lähes 2 ampeeria. Jos virtaa ei tule tarpeeksi, ei kortti saa yhdistettyä itseään matkapuhelinverkkoon. Arduino kommunikoi kortin kanssa sarjaliikenteen välityksellä. Kortin vaatimuksina on, että tiedonsiirtonopeutena käytetään 19200 baudinopeutta. Baudinopeudella tarkoitetaan sitä, kuinka monta bittiä sekunnissa voidaan siirtää. Kortille lähetetään sarjaliikenteessä AT-komentoja, joilla kortin internetyhteys määritetään ja käynnistetään.

Korttia ohjataan AT-komennoilla, joista esimerkkejä alla.

SAPBR-komento liittyy Bearer service -palveluun. Sillä määritetään yhteyden datansiirron mekanismit.

```
AT+SAPBR=3,1 "Contype","GPRS"
```

Määrittää yhteystyypiksi GPRS-yhteyden.

```
AT+SAPBR=3,1 "Apn","internet"
```

Määrittää APN:n (Access point name).

```
AT+SAPBR=1,1
```

Avaa Bearer-yhteyden.

```
AT+SAPBR=2,1
```

Tarkistaa että Bearer on avattuna.

```
AT+CLTS=1
```

Kytkee päälle aikaleiman. Aikaleima on paikallisaikaa.

```
AT+CENG=3
```

Kytkee Engineering moden päälle. Engineering mode -tilassa GPRS-kortin liikenteeseen lisätään tietoja enemmän. Tämän tilan avulla saadaan haettua kellonaika AT+CCLK?-komennolla.

```
AT+HTTPIPINIT
```

Käynnistää HTTP-palvelun.

```
AT+HTTTPARA="CID",1
```

Asettaa HTTP-yhteyden Bearer-profiilin tunnisteeseen.

```
AT+HTTTPARA="URL","HTTP://xx.xxx.xx.xx:xxxx/jsonpost.php/"
```

Määrittää kohdeosoitteen.

```
AT+HTTTPARA="CONTENT\","application/json"\r
```

Määrittää sisältötyypin

```
AT+HTTTPDATA=" + String(reading.length())+",100000
```

Määrittää lähtevän viestin merkkien määrän sekä lähetysajan pituuden millisekunteina. Komennon kirjoittamisen jälkeen kirjoitetaan ohjelmassa viestin sisältävän muuttujan "reading" sisältö GPRS-sarjaliikenteeseen.

```
AT+HTTTPACTION=1
```

Käynnistää HTTP-toiminnon (parametrit: 0=Read, 1=POST, 2=HEAD).

```
AT+HTTPREAD
```

Lukee palvelimen vastauksen.

```
AT+HTTPTERM
```

Päättää HTTP-yhteyden.

```
AT+SAPBR=0,1
```

Sulkee Bearerin.

#### 4.1.4 Sensorit

Lämpötilan ja kosteuden mittaamiseen käytetään DHT22-sensoria. Sensori mittaa lämpötilaa -40 celsiusasteesta +80 asteeseen +/- 0,5 celsiusuksen tarkkuudella. Ilman suhteellisen kosteusprosentin sensori mittaa 0...100 % väliltä 2...5 prosentin tarkkuudella. Sensori on pienikokoinen ja sille on tehty tässä työssä suojarakenne, jonka tehtävänä on suojata sensoria vesisateelta sekä vähentää suoran auringonvalon aiheuttamaa virhettä mittaustulokseen.



Kuva 13. Lämpötila-anturin kotelo.



Kuva 14. Lämpötila-anturi paikoillaan kotelossa.

Suojan rakenteeseen on otettu mallia kaupallisten sääasemien anturin rakenteesta. Se on koottu muovisista lautasen mallisista levyistä, joihin on porattu reikiä, jotta ilma pääsisi vaihtumaan mahdollisimman esteettömästi anturin ympärillä. DHT22:lle löytyy ohjelma-kirjasto, josta löytyy valmiita funktioita anturin käyttöön. Anturia luettaessa täytyy ottaa huomioon anturin vaatima palautumisaika mittausten välissä, joka on noin 1...2 sekuntia.

Anturi on kiinnitetty kolmijalkaiseen kuvausjalustaan, jolla anturin korkeus on säädettävissä 180 cm:n korkeuteen asti. Lämpötilan ja suhteellisen kosteuden lisäksi sääasemat usein mittaavat tuulen nopeutta sekä ilmanpainetta. Nämä ominaisuudet on mahdollista lisätä sääasemaan tarvittaessa helposti jälkeinpäin.



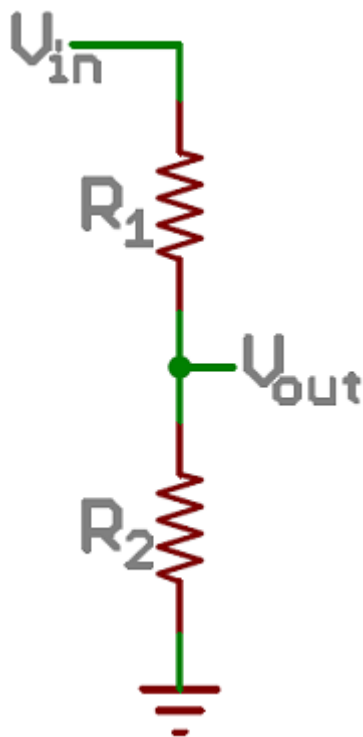
Kuva 15. Säasema mittauspisteellään.

## Jännitteen mittaus

Jännitteen mittaukseen on käytetty ulkoista 2,5 voltin referenssijännitettä sekä vastuksista tehtyä jännitteen jakajaa, joka alentaa akulta mitatun jännitteen siten, että se ei ylitä analogitulon maksimijännitettä 5 voltia.

$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

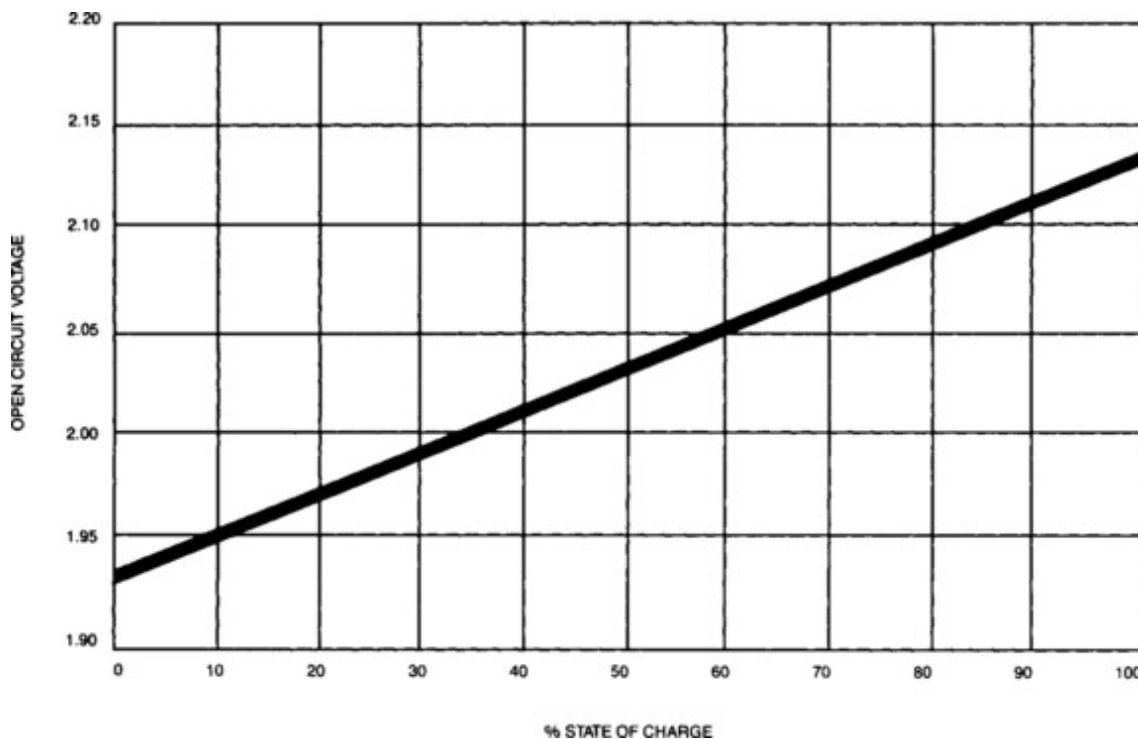
Kuva 16. Jännitteenjakajan kaava (18).



Kuva 17. Jännitteenjakajan kytkentäkaavio (18).

Alennettu mittausjännite mitataan Arduinon analogitulopinnistä A0. Pinniin tuleva jännite 0...2,5 voltia skaalataan arvoksi väliltä 0...1023, jonka perusteella Arduino laskee jännitteen. Yhdellä mittauksella otetaan 100 näytettä, joista lasketaan keskiarvo mittaukselle. Näin pyritään saamaan luotettavampi mittaustulos.

Lyijyakun varaustilaa voidaan arvioida tarkkaan mittaamalla akun napojen jännite ns. avoimen piirin mittauksena. Lyijyakun jännite muuttuu lähes lineaarisesti varaustason laskiessa. Avoimen piirin jännitteen mahdollisimman tarkka mittaus tulisi toteuttaa siten, että akulla ei ole kuormaa kytkettynä ja että akun varaamisesta tai kulutuksesta olisi kulunut 24 tuntia. (19)



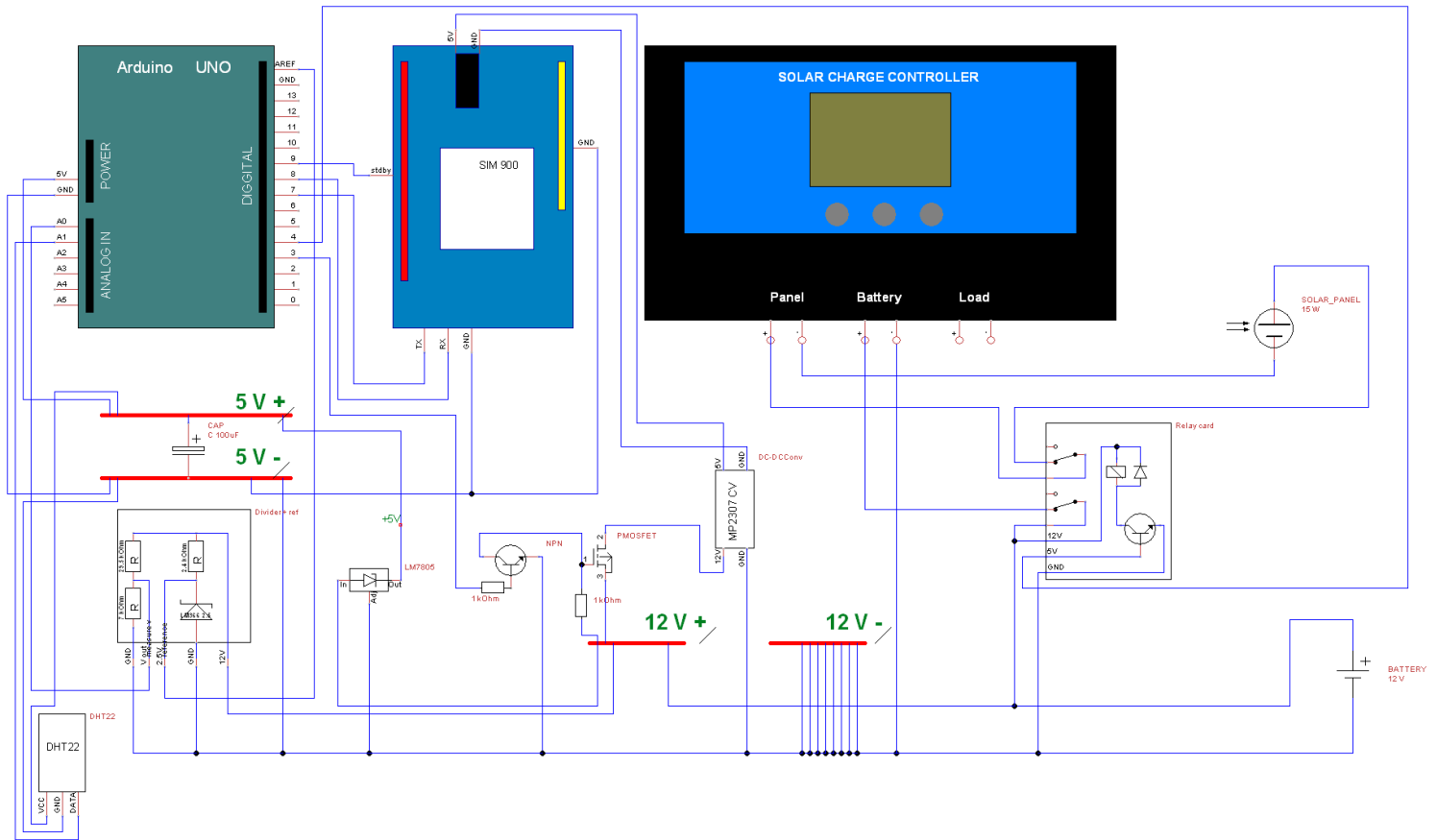
Kuva 18. Lyijyakkukennon jännitteen suhde varaustilaan (20)

Tässä insinööriyössä ei voida kuitenkaan mitata akun jännitettä avoimesta piiristä, koska mikrokontrolleri tarvitsee jatkuvan käyttöjännitteen akulta. Jännitteenmittauksen ajaksi aurinkopaneeli ja lataussäädin erotetaan piiristä, jotta ne eivät vääristäisi lataamalla akun jännitettä mittauksen aikana. Akun jännitteeseen vaikuttaa mittaushetkellä hiukan lataussäätimen ja paneelin erotuksessa käytettävän releen kelan kuluttama virta sekä Arduinon käyttämä virta. Tässä työssä akun varaustilaa arvioidaan suljetusta piiristä. Tällä tavalla mitatuilla tuloksilla saatiin riittävän tarkka kuva akun varaustilasta.

Releen ohjaamiseen käytetään transistoria. Arduinon ohjauspinnien maksimivirta on noin 40 mA, jolloin sillä tehtävät ohjauskytkennät kannattaa toteuttaa transistorien avulla Arduinon lähtöjen suojaamiseksi.



## KYTKENTÄKAAVIO



Kuva 19. Sääaseman kytkentäkaavio

## 4.2 Tietokantapalvelin

Järjestelmän palvelinpään toteutus on LAMP-serveri. LAMP on lyhenne, jolla kuvataan laajasti käytössä olevaa palvelinteknologian kokoonpanoa. Siinä käyttöjärjestelmänä toimii Linux ja webpalvelinohjelmistona Apache, tietokantapalvelinohjelmistona MySQL ja vuorovaikutuksen luovana skriptikielenä tyypillisesti PHP (21).

Tässä työssä LAMP-palvelin on pystytetty Raspberry Pi -mikrotietokoneelle RPi:lle on tässä työssä 3d-tulostettu kotelo passiivisella ilmanvaihdolla. Tulosteen malli on haettu Thingiversen web-sivulta, josta löytyy runsaasti sivuston käyttäjien mallintamia 3d-malleja. Käyttöjärjestelmäksi valittiin Raspbian, joka on RPi:lle optimoitu Debian Linuxiin

pohjautuva käyttöjärjestelmä (22). Asennukseen käytettiin NOOBS-asennuspakettia, joka mahdollistaa käyttöjärjestelmän asentamisen vaivattomasti muistikortin avulla. Muistikortin tulee olla vähintään 8 GB:n suuruinen. NOOBS:in asennustiedostot kopioidaan muistikortin juureen ja tämän jälkeen muistikortti asetetaan RPI:hin ja käynnistetään RPI. Loput asennuksesta opasti asennusohjelma (23). Debianissa eri ohjelmien asentamista helpottamaan on tehty Advanced package tool (myöhemmin Apt). Aluksi Apt päivitettiin sen tuoreimpaan versioon kirjoittamalla komentoriville seuraava komento: (24)

```
pi@raspberrypi:~ $ sudo apt update && sudo apt upgrade -y
```

Tämän jälkeen asennettiin Apache-webpalvelin. Apachen asennus tapahtui komentorivin kautta seuraavasti:

```
pi@raspberrypi:~ $ sudo apt install apache2 -y
```

Asennuksen jälkeen RPI:n kansioon `/var/www/html/` ilmestyi `index.html`-sivu. Sivun testaamista varten selvitettiin RPI:n ip-osoite seuraavalla komennolla: (24)

```
pi@raspberrypi:/var/www/html $ hostname -I
```

(24)

PHP:tä käytetään tässä työssä rajapintojen luomiseen. Rajapintoja tehtiin yhteensä kolme. Yksi rajapinta vastaa sääaseman lähettämän datan siirtämisestä tietokantaan, toinen rajapinta ottaa vastaan mittausaikavälipyntöjä ja kolmas kertoo tämänhetkisen mittausaikavälin. PHP:n asentamiseksi RPI:lle suoritettiin seuraava komento:

```
pi@raspberrypi:/var/www/html $ sudo apt install php -y
```

Seuraavaksi asennettiin tietokantapalvelin. Tietokantapalvelin toimii tietokannan ylläpitäjänä. Sen asennus tehtiin seuraavalla komennolla:

```
pi@raspberrypi:/var/www/html $ sudo apt install mariadb-server php-mysql -y
```

Asennuksen jälkeen suoritettiin apache-palvelimen uudelleen käynnistys seuraavalla komennolla:

```
pi@raspberrypi:/var/www/html $ sudo service apache2 restart
```

Komennon jälkeen annettiin salasana, jolla suojataan tietokanta sekä vastattiin myönteästi suojauksiin liittyviin jatkokysymyksiin.

PhpMyAdmin on graafinen sovellustyökalu, jolla voidaan hallinnoida MySQL-tietokantoja. Se asennettiin seuraavalla komennolla:

```
pi@raspberrypi:/var/www/html $ sudo apt install phpmyadmin -y
```

Asennusohjelma kysyi seuraavaksi kysymyksiä:

Configuring phpmyadmin? → Yes

MySQL application password for phpMyAdmin à MySQL:n salasana.

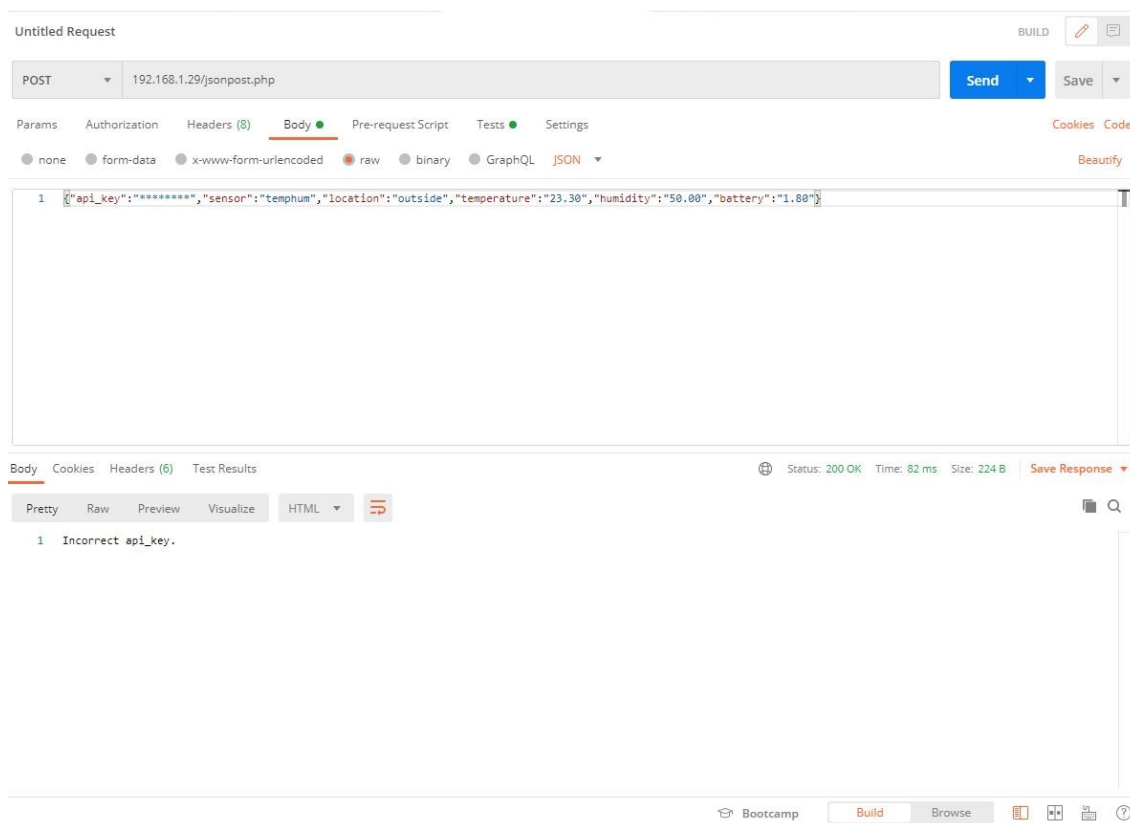
Seuraavaksi otettiin käyttöön PHP MySQLi -lisäosa ja uudelleen käynnistettiin Apache:

```
pi@raspberrypi:/var/www/html $ sudo phpenmod mysqli  
pi@raspberrypi:/var/www/html $ sudo service apache2 restart
```

Lopuksi vielä siirrettiin phpMyAdmin-kansio /var/www/html/ kansioon:

```
pi@raspberrypi:/var/www/html $ sudo ln -s /usr/share/phpmyadmin  
/var/www/html/phpmyadmin
```

Postman sovellus on API (application programming interface) -kehitystyökalu. Sillä voidaan testata rajapintoja kuten esimerkiksi tässä työssä käytettyä PHP-rajapintaa. PHP-rajapinnan testaus tapahtui siten, että Postman-ohjelmalla lähetettiin HTTP POST-request vastaanottavalle palvelimelle. Palvelimen päässä PHP-skripti validoi lähetetyn pyynnön ja lähettää vastauksen. Postman näyttää vastausviestin sekä tietoja, kuten kuinka kauan vastauksessa kesti ja onnistuiko yhteys kohdepalvelimeen.



Kuva 20. Kuvassa palvelimelta saatu vastausviesti, kun tunnuksen "api\_key"-arvo on väärä.

### 4.3 Tietokanta

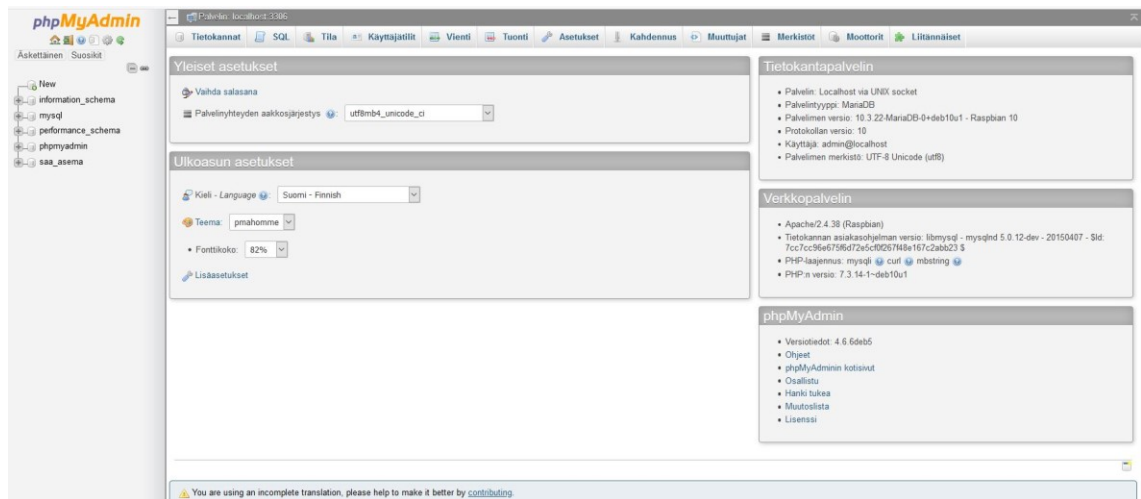
Tietokannan rakenne on tässä insinööriyössä hyvin yksinkertainen. Kanta koostuu yhdenlaisesta taulusta, johon sääasemalta tietoa vastaanottava rajapinta lisää tietueita. Taulun nimi on SensorData. Tietueen avainattribuuttina toimii id-arvo. Tämän insinööriyön tietokannassa se tarkoittaa, että monesko lisätty tietue on kyseessä. Sensor-arvoksi voidaan kirjoittaa sensorin tyyppi. Tässä työssä kenttään merkitään "temp\_hum", jolla viitataan lämpötilan ja kosteuden mittaukseen. Location-kenttää voitaisiin käyttää merkitsemään eri paikoissa olevia sääasemia. Tässä työssä siinä on merkintä "outside". Temperature-kenttään merkitään lämpötila 2 desimaalin tarkkuudella. Humidity\_rh-kenttään merkitään ilman suhteellinen kosteusprosentti 2 desimaalin tarkkuudella. Battery-

kenttään merkitään akun jännite 2 desimaalin tarkkuudella. Viimeisenä taulussa merkitään aikaleima timestamp-nimiseen kenttään. Sen arvon lisää tietokantapalvelin, joten sitä ei lähetetä sääasemalta.

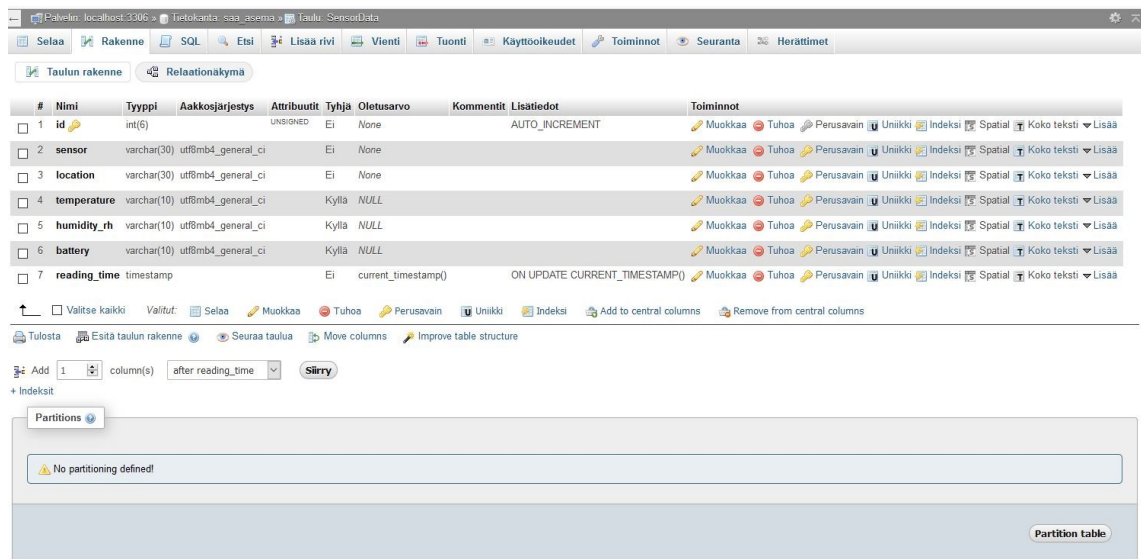
SensorData
<b>id</b>
sensor
location
temperature
humidity_rh
battery
reading_time

Kuva 21. SensorData-tietotaulu. Lihavoitu teksti tarkoittaa avainattribuuttia.

Tietokanta ja tietokannan taulu luotiin PhpMyAdmin-sovelluksella. Se on PHP-kielillä toteutettu ilmainen ohjelma, jolla voidaan hallinnoida MySQL-tietokantoja webin välityksellä.



Kuva 22. PhpMyAdmin-sovellus.



Kuva 23. Taulu "SensorData" PhpMyAdmin-sovelluksessa.

PHP:lla toteutettu kuuntelija API vastaanottaa json-muotoon muotoillun pyynnön. Kuuntelija tarkastaa, että vastaanotettava tiedosto on json-muodossa. Kun tieto on vastaanotettu, puretaan json-formaatti ja tarkastetaan api key. Jos api key on oikea, avaa kuuntelija yhteyden tietokantaan ja lähettää pyynnön kannalle. Rajapinta vastaa Arduinolle onnistuneen SQL-kyselyn päätteeksi "Sql query OK."

```
$sql = "INSERT INTO SensorData (sensor, location, temperature, humidity_rh,
battery) VALUES ('" . $sensor . "', '" . $location . "', '" . $temperature .
"', '" . $humidity . "', '" . $battery . "')";
```

Yllä luodaan sql-kysely, jolla viedään SensorData-taulun tietueeseen Arduinolta vastaanotetut arvot.

#### 4.4 Käyttöliittymä/Client

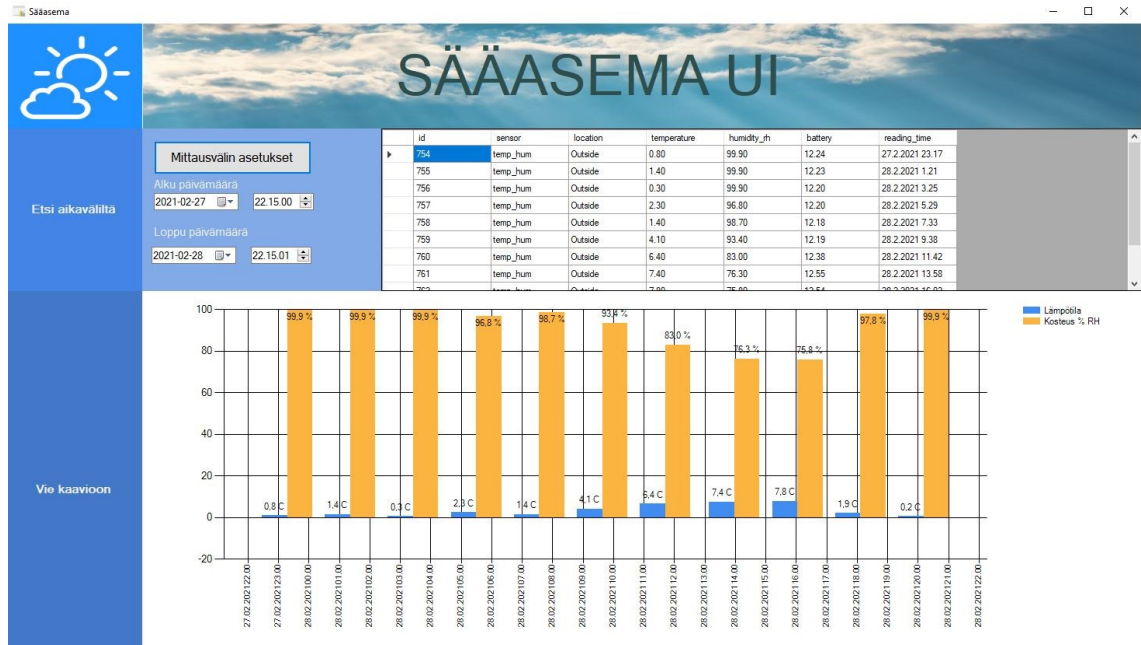
Halusin luoda järjestelmälle yksinkertaisen Windows-sovelluksen, jolla sääaseman keräämiä tietoja voidaan katsella tietokannasta. Vaatimukset sovellukselle olivat, että se kykenee hakemaan MySQL-tietokannasta arvoja halutulta aikaväliltä ja että niistä saadaan muodostettua kaavio. Lisäksi sovelluksesta voidaan asettaa sääaseman tiedonkeruuväli. Sovelluksen lähdekoodi löytyy githubista (25).

Ohjelmointikielenä sovelluksen tekoon käytettiin C Sharp (C#) -kieltä. Ohjelmointikehitys tehtiin Microsoft Visual Studiolla käyttäen Windows Formsia, joka on osa .NET-kirjastoja. WinFormsia on käytetty paljon käyttöliittymien tekemiseen. WinFormsissa Form on visuaalinen pinta, jolle voidaan liittää helposti erilaisia toiminnallisuuksia, kuten nappeja ja tekstinsyöttökenttiä. Formille lisätään halutut komponentit, jonka jälkeen niille ohjelmoidaan halutut toiminnallisuudet. (26.)

Sääasema ohjelmassa syötetään aloitus- ja lopetuspäivämäärät ja kellonajat jaksolle, jonka aikana kerättyä säädataa halutaan katsella. Kun ajat on määritetty, painetaan "Etsi aikaväliltä" -nappia. Napin painallus käynnistää yhteyden tietokantapalvelimelle ja välittää SQL-pyyntöä annetuilla aikaparametreilla.

```
("Select * from saa_asema.SensorData WHERE reading_time BETWEEN " +
    "" + queryStart + "" and "" + queryEnd + """, myConn)
```

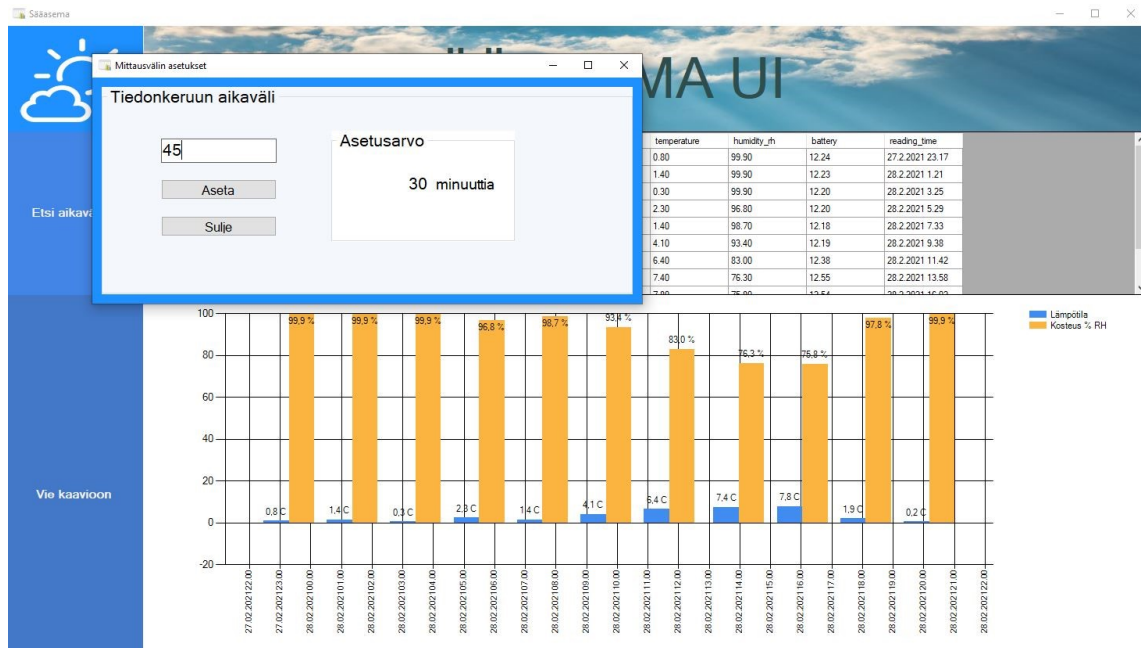
Muuttuja "myConn" sisältää palvelimen, tietokannan nimen, käyttäjän tunnuksen ja salasanan. Jos kysely onnistuu, kirjoitetaan haetut arvot datataulukkoon. Kun taulukkoon on viety dataa, voidaan painaa "Vie kaavioon" -nappia, jolloin ohjelma piirtää kaavion taulukosta.



Kuva 24. Asiakasohjelmassa säätietoa esitettyä kaaaviona ja taulukossa.

Sääseman toimintaa ohjaava aikaväli voidaan asettaa painamalla "Mittausvälin asetukset" -nappia. Napista painamalla avautuu uusi pienempi ikkuna, joka näyttää, montako minuuttia aikaväli on nyt. Syöttämällä tekstikenttään minuuttimäärän väliltä 5-240 ja painamalla "Aseta"-nappia viedään uusi asetusrvo rajapinnalle, joka tallentaa sen tiedostoon. Tästä tiedostosta sääsema käy pyytämässä aikavälin toisen rajapinnan avulla.





Kuva 25. Mittausaikavälin asettaminen.

## 5 Tuloksien tarkastelu

Sähköjärjestelmän toimintakykyä koestettiin neljänä eri ajankohtana. Elokuussa 2020 järjestelmää koekäytettiin 22.8. - 14.9.20, jona aikana auringonvalon määrä riitti lataamaan yhteen akkuun riittävästi varausta, jotta järjestelmä pysyi toimintakunnossa. Päivän pituus vaihteli tuona aikana 15h 05min–12h 58min. Toinen testijakso suoritettiin helmikuussa 2021, jolloin jakson (26.2. - 3.3.21) aikana oli aurinkoisia päiviä, mutta päivän pituus (10h 10min–10h 38min) oli lyhyempi sekä akkujen olosuhteet kylmyyden vuoksi haasteellisemmat. Tuolla jaksolla käytettiin kahta 9 Ah:n akkua rinnan. Niiden varausta järjestelmä kykeni tuossa vaiheessa vuotta pitämään heikosti, jonka vuoksi akkujen jännite laski päivä päivältä alhaisemmaksi. Kolmas testijakso oli 3. - 10.4.21. Tuona aikana laitteen akku varautui jokaisen päivän aikana täyteen päivien pituuden vaihdellen 13 h 32 min – 14 h 11 min. Mittaus suoritettiin Janakkalassa (60.783°N 24.533°E), jossa tuona aikana auringon nousu- ja laskuaika oli alussa 6:39 - 20:12 ja lopussa 6:18 - 20:30. Viimeinen testijakso suoritettiin 28.4. - 14.5.21. Viimeisen testijakson aikana auringon nousu- ja laskuaika oli alussa 5:23 – 21:14 ja lopussa 4:40-21:55. Koejaksojen perusteella näyttää siltä, että tällä sijainnilla sääasema kykenisi pysymään toimintakuntoisena

luultavasti maaliskuusta syys- tai lokakuuhun. Suljettujen akkujen vaihto autonakkuun voisi olla kylmempään aikaan parempi vaihtoehto niiden hieman paremman latautuvuuden vuoksi kylmissä olosuhteissa. Lisäksi laitteen virrankulutus olisi hyvä saada 60 mA:n tasolta esimerkiksi 10 mA:n lukemiin.

Järjestelmälle alkuun 5 voltin jännitteen tuottanut hakkuritekniikkaan perustuva DC-jännitteenalennin osoittautui ongelmalliseksi. Kun työssä edettiin vaiheeseen, jossa keskityttiin akun jännitteenmittauksen toteutukseen, huomattiin mittaustulosten vaihtelevan useita satoja millivoltteja. Vianetsintää varten kirjoitettiin yksinkertainen koodi Arduinoon, jossa Arduino tulostaa usb:n kautta IDE-sovelluksen monitoiroti-ikkunaan jatkuvasti jännitteen. Lopulta hakkurin vaihtaminen regulaattoriksi poisti häiriön, ja jännite tasaantui. DC-muunnin myös kulutti virtaa enemmän, kuin sen datalehdellä mainittiin, joten se on voinut olla viallinenkin. Jännitteen luennan tarkkuus suljetulla piirillä toteutettuna riittää mielestäni tähän sovellukseen hyvin. Tavoitteena oli saada esim. kolmeportainen High/Med/Low-mittari. Mitattu lukema kuitenkin kertoo trendistä tarkemmin kuin kolmeportainen lukema, vaikka absoluuttinen jännite olisikin hieman eri kuin mitattu.

Sääaseman koodin osalta minulla oli paljon haasteita. SRAM-muistin kokorajoitus tuli vastaan Arduino-ohjelman kehityksen loppupäässä, kun viimeisimpien toimintojen tuominen mukaan koodiin alkoi aiheuttaa ohjelman arvaamatonta käyttäytymistä. Tämä ohjasi minut tutkimaan tarkemmin muistin käyttöä käsitteleviä ongelmia Arduino.cc-foorumilta. Sen seurauksena karsin ohjelmasta turhia string-tekstin tulostuksia, joita oli käytetty kehityksen tukena kertomaan sarjamonitorissa kulloinkin meneillään olevasta ohjelman vaiheesta. String-muuttujat kuluttavat Arduinon muistikapasiteettia melko paljon, joten niiden karsiminen auttoi osaltaan tilannetta. Minulla oli myös varattuna aluksi useampi globaali char-taulukko vastaanotettavan tekstin käsittelyä varten. Muistin säästämiseksi ne muutettiin yhdeksi yhteiseksi globaaliksi taulukoksi, jota useampi funktio käyttää vuorollaan ja taulukon tyhjentämisestä huolehditaan sen käytön välissä. Arduino IDE -ohjelmointityökalu kertoo vapaaksi jääneen ohjelmamuistin ja dynaamisen muistin määrän sen jälkeen, kun ohjelma on onnistuneesti ladattu Arduinoon. Kun sain karsittua turhat muistia vievät muuttujat ohjelmasta, kasvoi dynaamisen vapaan muistin määrä riittäväksi, jolloin ohjelma alkoi toimia taas johdonmukaisesti.

Sääaseman lähettämät datat päätyivät pääosin palvelimelle. Testijaksojen aikana havaittiin, että noin kerran vuorokaudessa tietojen lähetys palvelimelle epäonnistui. Kun sääasema oli sijoitettuna kolmannen testijakson aikana paikkaan, jossa sen käyttämän sim-kortin operaattorin kuuluvuus on heikko, havaittiin, että epäonnistuneita lähetyksiä oli paljon useammin. Jatkokehityksenä tähän voisi sääaseman koodiin lisätä ominaisuuden, jolla se lukee palvelimen antaman vastauksen ja yrittää uudelleen lähettää tiedot, jos siirto on epäonnistunut.

## 6 Yhteenveto

Insinööriyössä pyrittiin rakentamaan sääasemajärjestelmä, joka sisältäisi aurinkovoimalla toimivan sääaseman, oman tietokannan palvelimiseen sekä tietokannan tarkasteluun asiakasohjelman. Työlle asetetut tavoitteet saavutettiin.

Työn haastavin osuus oli kenties sääaseman toteutus kokonaisuutena eli kaikkien osien saaminen toimimaan keskenään. Laitteen sähköisen kokoonpanon yhteydessä kohdatiin ongelmia, kun hakkurivirtalähteen aiheuttama sähköinen häiriö tuotti liikaa heittelyä akun jännitteenmittaamiseen. Lisäksi mikrokontrollerin ohjelman kehitys osoittautui huomattavasti haasteellisemmaksi. Kehityksen loppuvaiheessa ohjelman käyttämä muisti alkoi olla sen verran vähissä, että se alkoi aiheuttaa ohjelman suorituksen aikana odottamatonta käyttäytymistä.

Insinööriyön tekeminen oli minulle hyvin antoisaa. Verrattain yksinkertaisen säähavaintolaitteen ja järjestelmän kaikkien osien tekeminen ja yhteensovittaminen osoittautui työlläksi tehtäväksi.

Insinööriyön kesto venyi aiottua pidemmäksi erinäisistä syistä. Ohjelmointitaitausta minulla oli ennen työtä joidenkin koulukurssien verran C#-, C++-, SQL-skripteistä sekä pythonista ja Javasta. Insinööriyön aikana tutustuin PHP-kieleen rajapinnan muodostamiseksi sekä tietoliikenteessä käytettäviin AT-komentoihin. Minulla ei myöskään ollut ennestään kokemusta tehdä HTML-pyyntöjä, joita käytettiin API-rajapinnan kutsumiseen. Niin ikään SQL- ja web-palvelimen toteuttamisesta tai tietokannan ylläpidosta/hallinnoinnista minulla ei ollut ennestään kokemusta. Onnekseni asioita on helppo opiskella

internetin avulla löytyvien lukuisten tutoriaali- ja opastussivujen kautta, joita myös tuli hyödynnettyä työn aikana. Työn aikana opin myös paljon akkukäyttöisten laitteiden haasteista ja esimerkiksi akun varaustilan määrittämisen hankaluudesta. Lisäksi hakkuritehonlähteen aiheuttamat ongelmat lisäsivät ymmärrystä vaatimuksista jännitelähdettä kohtaan, varsinkin laitteessa, joka suorittaa mittauksia, joihin hakkurin aiheuttama häiriö voi vaikuttaa.

Insinööri työn tuloksena syntyneen laitteen jatkokehittämiseksi sääasemalle voisi lisätä ilmanpaine sekä tuulianturit. Myös mikrokontrollerin energiatehokkuutta voisi parantaa käyttämällä valmiin Arduino-kortin sijaan karsittua piiriä, jolloin virrankulutus saataisiin pienennettyä murto-osaan nykyisestä. Myös asiakasohjelmaan voisi tehdä lisää toiminnallisuutta ja sen käytettävyyttä parantaa. Websivulta käytettävä versio voisi myös olla nykypäiväinen ratkaisu. Työn palkitsevin kohta oli, kun huomasin, että sääasema pärjäsi päivästä toiseen ja kykeni jatkamaan toimintaansa ilman, että siihen tarvitsi puuttua.

## Lähteet

- 1 Oman sääaseman perustaminen. Ilmatieteenlaitos. Verkkoaineisto. <<https://www.ilmatieteenlaitos.fi/oma-saaasema>> Luettu 15.3.2021.
- 2 Maws201. Verkkoaineisto. <<https://www.vaisala.com/en/products/instruments-sensors-and-other-measurement-devices/weather-stations-and-sensors/maws201>>. Luettu 1.5.2021.
- 3 Vesterinen, Matias. 2015. Lyijyakuvertailu. Opinnäytetyö. MAMK. Theseus-tietokanta.
- 4 Sirola, Oskari. 2018. Aurinkosähköjärjestelmän toiminta ja aurinkopaneelin ominaiskäyrän mittaus. Insinööriö. Metropolia Ammattikorkeakoulu. Theseus-tietokanta.
- 5 BU-201a: Absorbent Glass Mat (AGM). Verkkoaineisto. <[https://batteryuniversity.com/learn/article/absorbent\\_glass\\_mat\\_agm](https://batteryuniversity.com/learn/article/absorbent_glass_mat_agm)> Luettu 15.3.2021.
- 6 FULBAT 12 V datasheet. Verkkoaineisto. <[https://fulbat.com/wp-content/uploads/2020/09/FULBAT\\_FP12-9\\_GeneralPurpose\\_2020.pdf](https://fulbat.com/wp-content/uploads/2020/09/FULBAT_FP12-9_GeneralPurpose_2020.pdf)>. Luettu 14.5.2021.
- 7 Lataussäätimet. Verkkoaineisto. <<https://www.aurinkosahko.net/category/11/lataussaatimet>>. Luettu 1.5.2021.
- 8 Tallqvist, Oskari. 2013. Sulautettu hybridiaurinkosähköjärjestelmä. Insinööriö. Metropolia Ammattikorkeakoulu. Theseus-tietokanta.
- 9 Silvonon Kimmo. 2018. Elektroniikka ja sähkötekniikka. S.77-91. Tallinna: Printon Trükikoda.
- 10 Arduino Uno Rev3. <<https://store.arduino.cc/arduino-uno-rev3>>. Luettu 1.5.2021.
- 11 DHT22. Verkkoaineisto. <[https://www.elecrow.com/wiki/index.php?title=Crowtail-AM2302\\_Humidity%26Temperature\\_Sensor](https://www.elecrow.com/wiki/index.php?title=Crowtail-AM2302_Humidity%26Temperature_Sensor)> Luettu 24.4.2021.

- 12 Raspberry Pi foundation. Verkkoaineisto. <<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>>. Luettu 1.5.2021.
- 13 Apache. Verkkoaineisto. <<https://www.2kmediat.com/apache/apachehistoria.asp>> Luettu 15.3.2021.
- 14 July 2020 Webserver survey. Verkkoaineisto. <<https://news.netcraft.com/archives/2020/07/27/july-2020-web-server-survey.html>> Luettu 15.3.2021.
- 15 Metropolia Tietokannat verkkokussi. 2020. Metropolia UAS:TX00CY40-3002 Tietokannat ja SQL.
- 16 PHP. Verkkoaineisto. <[https://hallstrom.fi/php\\_opas/php\\_1.html#otsikko1\\_1](https://hallstrom.fi/php_opas/php_1.html#otsikko1_1)> Luettu 15.3.2021.
- 17 Hakkuriteholähteet. Aalto yliopisto. Verkkoaineisto. <[https://wiki.aalto.fi/download/attachments/62723066/R6\\_k9\\_hakkurit.pdf?version=1&modification-Date=1333351184000](https://wiki.aalto.fi/download/attachments/62723066/R6_k9_hakkurit.pdf?version=1&modification-Date=1333351184000)> Luettu 11.5.2021.
- 18 Voltage divider. Verkkoaineisto. <<https://learn.sparkfun.com/tutorials/voltage-dividers/all>>. Luettu 1.5.2021.
- 19 O Reilly Rechargeable Batteries Applications Handbook. Section 4 4.3.2 State-Of-Charge indication.
- 20 O Reilly Rechargeable Batteries Applications Handbook. Appendix B. Sealed-Lead Cell and Battery Product Line Data.
- 21 LAMP. Verkkoaineisto. <[HTTPs://www.linux.fi/wiki/LAMP](http://www.linux.fi/wiki/LAMP)>. Luettu 2.3.2021.
- 22 Raspbian. Verkkoaineisto. <<https://www.raspbian.org/>> Luettu 4.3.2021.
- 23 Noobs. Verkkoaineisto. <<https://learn.adafruit.com/setting-up-a-raspberry-pi-with-noobs/overview>> Luettu 4.3.2021.

- 24 Raspberry Pi: Install Apache + MySQL + PHP (LAMP server). Verkkoaineisto. <<https://randomnerdtutorials.com/raspberry-pi-apache-mysql-php-lamp-server/>> Luettu 2.3.2021.
- 25 Kekki, Tuomas. 2021. Weathertracker UI käyttöliittymän lähdekoodi. Verkkodokumentti. <<https://github.com/DonBitti/WS>>. Luettu 16.5.2021.
- 26 WinForms. Verkkoaineisto. <<https://docs.microsoft.com/en-us/dotnet/desktop/winforms/overview/?view=netdesktop-5.0>> Luettu 15.3.2021.

## PHP kuuntelija json\_post.php

```
<?php

//Get input stream.
$json = file_get_contents("php://input");

//Check that request is a POST request.
if(strcasecmp($_SERVER["REQUEST_METHOD"], "POST") != 0){
    die("Request method is not POST");
}

//Check that the content type is application/json.
$contentType = isset($_SERVER["CONTENT_TYPE"]) ? trim($_SERVER["CONTENT_TYPE"]) : '';
if(strcasecmp($contentType, "application/json") != 0){
    die("Content type is not: application/json");
}

//Receive the RAW post data.
$content = trim(file_get_contents("php://input"));

//Attempt to decode the incoming RAW post data from JSON.
$decoded = json_decode($content, true);

//If json_decode failed, the JSON is invalid.

if(!is_array($decoded)){
    die("Invalid json form.");
}

$data = json_decode($json, true);
// Set your apikey to api_key_value.
$api_key_value = "*****";
$api_key = $data["api_key"];
if($api_key != $api_key_value) {
    die("Incorrect api_key.");
}

$sensor = $data["sensor"];
$location = $data["location"];
$temperature = $data["temperature"];
$humidity = $data["humidity"];
$battery = $data["battery"];

//Mysql database variables
$servername = "localhost";
$dbname = "saa_asema";
$username = "*****";
$password = "*****";

//Make connection to Mysql database
$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
```



```
        die("Connection failed: " . $conn->connect_error);
    }

    //Sql query
    $sql = "INSERT INTO SensorData (sensor, location, temperature, humidity_rh,
battery)
        VALUES ('" . $sensor . "', '" . $location . "', '" . $temperature .
        "', '" . $humidity . "', '" . $battery . "')";

    if ($conn->query($sql) === TRUE) {
        echo "Sql query OK.";
    }
    else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
    //close connection
    $conn->close();

?>
```

### PHP readinterval.php

```
<!DOCTYPE html>
<html><body>
<?php

$myFile = "minutes.txt";
$myFileLink = fopen($myFile, 'r');
$myFileContents = fread($myFileLink, filesize($myFile));
fclose($myFileLink);
echo '#' . $myFileContents . '#';
?>

</table>
</body>
</html>
```

## PHP setinterval.php

```
<!DOCTYPE html>
<html><body>

<?php

$minutes = $_GET['minutes'];
if ($minutes < 241===$minutes > 4){

$File = "minutes.txt";
$FileLink = fopen($File, 'w+') or die("Can't open file.");
$newContents = $minutes;
fwrite($FileLink, $newContents);
fclose($FileLink);
echo "Success";
}
else{
    echo "Failed";
}
?>

</html>
```

## Arduino ohjelma

```

#ifdef __arm__
// should use uinstd.h to define sbrk but Due causes a conflict
extern "C" char* sbrk(int incr);
#else // __ARM__
extern char *_brkval;
#endif // __arm__

#include <DHT.h>
#include <LowPower.h>
#include <SoftwareSerial.h>

SoftwareSerial GPRS(7, 8); //7 = TX, 8 = RX

int count = 0;
int i = 1; //if i = 0, send SMS.
const int buttonPin = 2;
int buttonState = 0;
int ledPin = 13;

String reading="";
String temp = "";
String humi = "";
String batt = "";

bool success = false;
int successCounter = 0;

float vOut = 0.0;
float vIn = 0.0;
#define R1 29600.0
#define R2 7050.0

//int value = 0;
int freeMem;

int minut=1; //new 27.3
int tunt=1; //new 27.3
int delaySeconds; //new 27.3
int delayRequest=31; //new 27.3
int pulses=450; // wait pulse for sleepmode

char minutes[3]; //new 27.3
char hours[3]; //new 27.3
char request[4]; //new 27.3
char buff [64];

#define ANALOG_IN_PIN A0
#define batteryRelay 4
#define simPower 3
#define DHTPIN A1
#define DHTTYPE DHT22

```

```

#define dhtpower 2

float ref_voltage = 2.47;          // Float for Reference Voltage
float h=0;
float t=0;
float b=0;
DHT dht(DHTPIN, DHTTYPE);
unsigned long time_now = 0;
unsigned long period = 3600000;

void setup() {
  pinMode(dhtpower, OUTPUT);
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(ANALOG_IN_PIN, INPUT);
  pinMode(simPower, OUTPUT);
  pinMode(batteryRelay, OUTPUT);
  analogReference(EXTERNAL);

  //muuttuja lämpötila-anturille
  GPRS.begin(19200);              // the GPRS baud rate
  Serial.begin(19200);            // the Serial port of Arduino baud rate.
  dht.begin();
  Serial.println("Go");
}

void loop() {

  //Start cycle...

  time_now = millis();
  digitalWrite(dhtpower, HIGH);
  readTempHum();                  //Updates temperature and humidity values to
global variables h and t.
  digitalWrite(dhtpower, LOW);

  digitalWrite(batteryRelay, HIGH); //Disconnect solarcharger from battery

  vIn = readBatteryVoltage();      //Measure voltage

  delay(1000);

  digitalWrite(batteryRelay, LOW); //re-connect Battery to solarcharger

  temp = String(t);
  humi = String(h);
  batt = String(vIn);
  reading="{\"api_key\": \"*****\", \"sensor\": \"temp_hum\", \"location\": \"Outside\", \"temperature\": \"\" + temp + \"\", \"humidity\": \"\" + humi +
  \"\", \"battery\": \"\" + batt + \"\"}"; //Generate json string
  delay(2000);

  Serial.println("Powering sim900.");
  SIM900power(true);
  delay(1000);
  Serial.println(reading);
  delay(2000);

  turnTransmissionLed(true);
}

```

```
connectGPRS();

sendData();

getIntervalGPRS();

getGPRSTime();

disconnectGPRS();

turnTransmissionLed(false);

clearBuffArray();

Serial.println("Tulostetaan tunt ja minut:");
Serial.println(tunt);
Serial.println(minut);
Serial.println("Tulostetaan request array:");
Serial.println(request);
Serial.println("Tulostetaan request minuutit:");
Serial.println(delayRequest);
Serial.println("Tulostus päättyi");

reading="";

Serial.println("Shutdown sim900.");

SIM900power(false);

delay(2000);

delaySeconds=calcDelay(minut,tunt,delayRequest);
Serial.println(delaySeconds);

int pulses=delaySeconds / 8;
Serial.println(pulses);
delay(2000);
int iTime=0;

while(iTime < pulses){

    LowPower.powerDown(SLEEP_8S, ADC_OFF, BOD_OFF);    //Power saving

    iTime= iTime+1;
}
}

void getHoursMinutes()
{
char date[50];
char *ptr1;
int indx=0;
int indx3=0;
while (GPRS.available())
{
    delay(10);
    date[indx3]=GPRS.read();
    indx3++;
    date[indx3]='\0';
}
while (indx<50){
```

```

if(date[indx]==' ')
{
    indx++;
    hours[0]=date[indx];
    Serial.print(date[indx]);
    indx++;
    hours[1]=date[indx];
    Serial.print(date[indx]);
    hours[2]='\0';
    indx++;
    indx++;
    minutes[0]=date[indx];
    Serial.print(date[indx]);
    indx++;
    minutes[1]=date[indx];
    Serial.print(date[indx]);
    minutes[2]='\0';
    //break;
}
indx++;
}
while (GPRS.available())
{
    GPRS.read();
}
tunt = strtol(hours, &ptr1,10);
minut = strtol(minutes, &ptr1,10);
}

int getIntervalRequest()
{
    //char requestResponse[64];
    int indx2=0;
    while (GPRS.available())
    {
        delay(10);
        buff[indx2]=GPRS.read();
        indx2++;
        buff[indx2]='\0';
    }

    int intervalMinutes=0;
    int iReq=0;
    int indx=0;
    char *ptr;

    while(indx<64)

    {
        Serial.print(buff[indx]);
        delay(10);
        if(buff[indx]=='#')
        {
            Serial.println("Löytyi!");
            indx++;
            Serial.println(buff[indx]);
            while(buff[indx]!='#')
            {
                request[iReq]=buff[indx];
                indx++;
                iReq++;
            }
        }
    }
}

```

```

    }
    delay(10);
    indx++;
}
intervalMinutes = strtol(request, &ptr,10); //make integer value from global
char array request
return intervalMinutes;

}

int calcDelay(int minutIn, int tuntIn, int intervalIn )
//inputs are given i minutes or hours. Output is given in seconds.
//Returns time interval to next start.
//if input interval is 30 or 60 or 120, then start time is matched to on the
hour.

{
    int hours=tuntIn;
    int minutes=minutIn;
    int interval=intervalIn;
    int delayOutput=0;

    switch (interval) {
        case 30:
            //case half hour interval
            //Calculate time before next interval point. In this case every half
hour
            if(minutes >= 30){
                Serial.print("Input minutes: ");
                Serial.println(minutes);
                delayOutput = 60*(60-minutes);
                Serial.println("Case 1.1");
            }
            else {
                Serial.print("Input minutes: ");
                Serial.println(minutes);
                delayOutput = 60*(30-minutes);
                Serial.println("Case 1.2");
            }
            Serial.println("Case 1 viive " );
            break;
        case 60:
            //case 1 hour interval
            Serial.print("Input minutes: ");
            Serial.println(minutes);
            delayOutput = 60*(60-minutes);
            Serial.println("Case 2");
            break;
        case 120:
            //case 2 hour interval
            Serial.print("Input minutes: ");
            Serial.println(minutes);
            delayOutput = 60*(120-minutes);
            Serial.println("Case 3");
            break;
        default:
            //case with only minutes interval
            delayOutput = 60*interval;
            Serial.println("Case default");
            break;
    }
}

```

```

return delayOutput; //Return seconds to sleep before next start
}

void turnTransmissionLed(bool state)
{
    if (state)
    {
        digitalWrite(ledPin, HIGH);
    }
    else
    {
        digitalWrite(ledPin, LOW);
    }
}

void readTempHum()
{
    delay(3000);
    h = dht.readHumidity();
    t = dht.readTemperature();
    delay(3000);
    h = dht.readHumidity();
    t = dht.readTemperature();
    delay(3000);
}

float readBatteryVoltage()
{
    delay(500);
    float adc_voltage = 0.0;
    float in_voltage = 0.0;
    int adc_value = 0;
    int i=0;
    float sumV = 0.0;
    float meanV = 0.0;
    while(i<100)
    {
        adc_value = analogRead(ANALOG_IN_PIN);
        adc_voltage = (adc_value * ref_voltage) / 1024.0; // Determine voltage at
ADC input
        in_voltage = adc_voltage / (R2/(R1+R2)); // Calculate voltage at divider
input
        sumV = sumV + in_voltage;
        i++;
    }
    meanV = (sumV/100);

    return meanV;
}

void clearSettingArray() { // function to clear buffer array
    for (int i = 0; i < 64; i++) {
        // setting[i] = NULL; // clear all index of array with com-
mand NULL
    }
}

void clearBuffArray() { // function to clear buffer array
    for (int i = 0; i < 64; i++) {
        buff[i] = NULL; // clear all index of array with command
NULL
}
}

```



```

    }
}

void ShowSerialData() {

    while (GPRS.available() != 0)
    {
        Serial.write(GPRS.read());
        delay(60);
    }
}

void printSerialData()
{
    const unsigned long timeout = 2000ul;
    unsigned long startChrono = millis();

    while (millis() - startChrono <= timeout) {
        if (GPRS.available()) {
            Serial.write(GPRS.read());
            startChrono = millis(); // reset the timeOut
        }
        delayMicroseconds(10);
    }
}

void SIM900power(bool up)
{
    if(up){
        digitalWrite(3,HIGH); //Switch power connection on
        delay(1000);
        pinMode(9, OUTPUT); // Turn sim900 on
        digitalWrite(9,LOW);
        delay(1000);
        digitalWrite(9,HIGH);
        delay(2000);
        digitalWrite(9,LOW);
        delay(3000);
    } else{
        pinMode(9, OUTPUT); // Turn sim900 off
        digitalWrite(9,LOW);
        delay(1000);
        digitalWrite(9,HIGH);
        delay(2000);
        digitalWrite(9,LOW);
        delay(3000);
        digitalWrite(3,LOW); //Switch power connection off
        delay(1000);
    }
}

void connectGPRS()
{
    Serial.println("Connect GPRS.");
    GPRS.print("AT+SAPBR=3,1,\"Contype\",\"GPRS\"\r");
    delay(1000);
    ShowSerialData();
    delay(1000);

    GPRS.print("AT+SAPBR=3,1,\"Apn\",\"internet\"\r");
    delay(2000);
}

```

```

ShowSerialData ();

GPRS.print ("AT+SAPBR=1,1\r");
delay (2000);
ShowSerialData ();

GPRS.print ("AT+SAPBR=2,1\r");
delay (1000);
ShowSerialData ();

GPRS.println ("AT+CLTS=1\r");
delay (1000);
GPRS.println ("AT+CENG=3\r");
delay (1000);
ShowSerialData ();
}

void sendData ()
{
  Serial.println ("start sending.");
  GPRS.print ("AT+HTTPINIT\r");
  delay (1000);
  ShowSerialData ();

  GPRS.print ("AT+HTTTPARA=\"CID\",1\r");
  delay (1000);
  ShowSerialData ();

  GPRS.println ("AT+HTTP-
  PARA=\"URL\", \"HTTP://XX.XXX.XX.XX:XXXX/jsonpost.php/\" \r"); //Public server IP
  address
  delay (1000);
  ShowSerialData ();

  GPRS.print ("AT+HTTTPARA=\"CONTENT\", \"application/json\" \r");
  delay (4000);
  ShowSerialData ();

  GPRS.print ("AT+HTTPDATA=" + String (reading.length ()) + ",100000\r"); //
  delay (1000);
  ShowSerialData ();

  GPRS.print (reading + "\r");
  delay (1000);
  ShowSerialData ();

  GPRS.print ("AT+HTTPACTION=1\r");
  delay (1000);
  ShowSerialData ();

  GPRS.print ("AT+HTTPREAD\r");
  delay (1000);

  Serial.print ("GPRS buffer:");
  Serial.println (GPRS.available ());
  delay (1000);

  char c = ' ';
  int len = 30;
  char termChar = '.';
  int index = 0;

```

```

while (GPRS.available())
{
    delay(10);
    buff[index]=GPRS.read();
    index++;
    buff[index]='\0';
}
delay(1000);

Serial.println(buff);
Serial.println(sizeof(buff));
index = 0;

char * p;
char message [14]= "Sql query OK.";
p = strstr (buff, message);
Serial.println(p);

clearBuffArray();

delay(1000);
ShowSerialData();
}

void getIntervalGPRS()
{
    //GET MINUTES PARAMETER

    GPRS.println("AT+HTTTPARA=\"URL\", \"HTTP:// XX.XXX.XX.XX:XXXX /readinter-
val.php/\"\\r");//Public server IP address
    delay(1000);
    ShowSerialData();

    GPRS.print("AT+HTTTPACTION=0\\r");
    delay(1000);
    ShowSerialData();

    GPRS.print("AT+HTTTPREAD\\r");
    delay(1000);
    delay(2000);
    delayRequest=getIntervalRequest();
    clearBuffArray();
    ShowSerialData();
    delay(1000);
}

void getGPRSTime(){
    //GET LOCAL TIME FROM NETWORK
    Serial.println("Aika");
    GPRS.print("AT+CCLK?\\r");
    delay(1000);
    Serial.print("GPRS buffer:");
    Serial.println(GPRS.available());

    char readChar;

    getHoursMinutes(); //Extract hours and minutes to global int variables tunt
& minut
    ShowSerialData();
}

```

```
void disconnectGPRS()
{
    GPRS.print("AT+HTTPTERM\r");
    delay(1000);
    ShowSerialData();

    GPRS.print("AT+SAPBR=0,1\r");
    delay(8000);
    ShowSerialData();
}

int freeMemory() {
    char top;
#ifdef __arm__
    return &top - reinterpret_cast<char*>(sbrk(0));
#elif defined(CORE_TEENSY) || (ARDUINO > 103 && ARDUINO != 151)
    return &top - __brkval;
#else // __arm__
    return __brkval ? &top - __brkval : &top - __malloc_heap_start;
#endif // __arm__
}
```

## Mittausdata

### Jakso 2

ID	sensor	location	Temperature	Humidity	Battery	DateTime
741	temp_hum	Outside	2.2	99.9	12.46	26.2.2021 18:19
742	temp_hum	Outside	2.9	99.9	12.45	26.2.2021 20:23
743	temp_hum	Outside	2.3	99.7	12.45	26.2.2021 22:27
744	temp_hum	Outside	1.5	99.7	12.41	27.2.2021 0:31
745	temp_hum	Outside	1.0	99.4	12.39	27.2.2021 2:35
746	temp_hum	Outside	0.1	95.6	12.34	27.2.2021 6:44
747	temp_hum	Outside	0.3	93.5	12.31	27.2.2021 8:48
748	temp_hum	Outside	1.6	90.0	12.33	27.2.2021 10:52
749	temp_hum	Outside	3.1	78.2	12.40	27.2.2021 12:56
750	temp_hum	Outside	5.0	68.6	12.83	27.2.2021 15:00
751	temp_hum	Outside	1.8	87.4	12.41	27.2.2021 17:05
752	temp_hum	Outside	-0.7	95.6	12.26	27.2.2021 19:09
753	temp_hum	Outside	1.0	93.3	12.25	27.2.2021 21:13
754	temp_hum	Outside	0.8	99.9	12.24	27.2.2021 23:17
755	temp_hum	Outside	1.4	99.9	12.23	28.2.2021 1:21
756	temp_hum	Outside	0.3	99.9	12.20	28.2.2021 3:25
757	temp_hum	Outside	2.3	96.8	12.20	28.2.2021 5:29
758	temp_hum	Outside	1.4	98.7	12.18	28.2.2021 7:33
759	temp_hum	Outside	4.1	93.4	12.19	28.2.2021 9:38
760	temp_hum	Outside	6.4	83.0	12.38	28.2.2021 11:42
761	temp_hum	Outside	7.4	76.3	12.55	28.2.2021 13:58
762	temp_hum	Outside	7.8	75.8	12.54	28.2.2021 16:02
763	temp_hum	Outside	1.9	97.8	12.21	28.2.2021 18:06
764	temp_hum	Outside	0.2	99.9	12.15	28.2.2021 20:11
765	temp_hum	Outside	0.0	99.9	12.11	28.2.2021 22:15
766	temp_hum	Outside	-0.2	99.9	12.09	1.3.2021 0:19
767	temp_hum	Outside	-1.0	99.9	12.08	1.3.2021 2:23
768	temp_hum	Outside	-1.5	99.9	12.05	1.3.2021 4:27
769	temp_hum	Outside	1.2	99.9	12.05	1.3.2021 8:35
770	temp_hum	Outside	4.6	88.2	12.23	1.3.2021 10:39
771	temp_hum	Outside	6.0	78.8	12.43	1.3.2021 12:43
772	temp_hum	Outside	6.2	78.2	12.41	1.3.2021 14:48
773	temp_hum	Outside	5.4	80.4	12.18	1.3.2021 16:52
774	temp_hum	Outside	3.9	88.0	12.08	1.3.2021 18:56
775	temp_hum	Outside	2.8	92.2	12.05	1.3.2021 21:01

776	temp_hum	Outside	2.9	92.3	12.03	1.3.2021 23:05
777	temp_hum	Outside	2.5	93.4	12.00	2.3.2021 3:13
778	temp_hum	Outside	2.0	93.1	11.98	2.3.2021 5:17
779	temp_hum	Outside	2.4	90.5	11.98	2.3.2021 9:25
780	temp_hum	Outside	4.4	77.9	12.19	2.3.2021 11:29
781	temp_hum	Outside	6.9	65.9	12.63	2.3.2021 13:34
782	temp_hum	Outside	6.7	62.8	12.34	2.3.2021 15:38
783	temp_hum	Outside	3.6	73.4	12.06	2.3.2021 17:42
784	temp_hum	Outside	0.5	91.9	11.97	2.3.2021 19:47
785	temp_hum	Outside	0.2	92.4	11.94	2.3.2021 21:51
786	temp_hum	Outside	-0.8	95.2	11.92	2.3.2021 23:55
787	temp_hum	Outside	-2.2	99.4	11.90	3.3.2021 1:59
788	temp_hum	Outside	-2.6	99.9	11.87	3.3.2021 4:02
789	temp_hum	Outside	-3.3	99.9	11.85	3.3.2021 6:06
790	temp_hum	Outside	-2.6	99.9	11.86	3.3.2021 8:10
791	temp_hum	Outside	1.8	93.7	12.02	3.3.2021 10:14
792	temp_hum	Outside	5.8	73.6	12.11	3.3.2021 12:18
793	temp_hum	Outside	6.3	76.8	12.20	3.3.2021 14:23
794	temp_hum	Outside	5.6	85.2	12.00	3.3.2021 16:27

### Jakso 3

ID	sensor	location	Temperature	Humidity	Battery	DateTime
1007	temp_hum	Outside	4.5	71.5	12.60	3.4.2021 11:07
1008	temp_hum	Outside	5.2	75.6	12.83	3.4.2021 12:03
1009	temp_hum	Outside	8.1	57.6	12.83	3.4.2021 14:03
1010	temp_hum	Outside	5.5	53.0	12.75	3.4.2021 15:03
1011	temp_hum	Outside	7.9	44.0	12.83	3.4.2021 16:03
1012	temp_hum	Outside	5.9	47.4	12.74	3.4.2021 17:03
1013	temp_hum	Outside	5.9	46.9	12.59	3.4.2021 19:03
1014	temp_hum	Outside	4.6	56.7	12.54	3.4.2021 20:03
1015	temp_hum	Outside	3.1	62.8	12.51	3.4.2021 21:03
1016	temp_hum	Outside	2.1	66.0	12.49	3.4.2021 22:03
1017	temp_hum	Outside	1.0	77.2	12.45	3.4.2021 23:02
1018	temp_hum	Outside	0.5	89.9	12.41	4.4.2021 1:03
1019	temp_hum	Outside	0.9	90.2	12.39	4.4.2021 3:03
1020	temp_hum	Outside	1.1	91.3	12.38	4.4.2021 4:03
1021	temp_hum	Outside	1.3	93.2	12.36	4.4.2021 5:03
1022	temp_hum	Outside	1.2	95.1	12.35	4.4.2021 6:03
1023	temp_hum	Outside	1.5	95.4	12.34	4.4.2021 7:03
1024	temp_hum	Outside	2.0	95.3	12.36	4.4.2021 8:03
1025	temp_hum	Outside	3.1	93.1	12.47	4.4.2021 9:03

1026	temp_hum	Outside	7.1	75.3	12.83	4.4.2021 11:03
1027	temp_hum	Outside	8.4	66.6	12.83	4.4.2021 12:03
1028	temp_hum	Outside	8.9	46.6	12.83	4.4.2021 18:05
1029	temp_hum	Outside	6.3	54.7	12.79	4.4.2021 19:03
1030	temp_hum	Outside	4.8	64.3	12.73	4.4.2021 20:02
1031	temp_hum	Outside	4.0	76.9	12.70	4.4.2021 21:03
1032	temp_hum	Outside	3.2	86.6	12.68	4.4.2021 22:03
1033	temp_hum	Outside	3.3	90.8	12.65	4.4.2021 23:03
1034	temp_hum	Outside	3.7	92.6	12.64	5.4.2021 0:03
1035	temp_hum	Outside	3.8	94.0	12.63	5.4.2021 1:02
1036	temp_hum	Outside	3.9	94.9	12.61	5.4.2021 2:03
1037	temp_hum	Outside	4.0	98.2	12.55	5.4.2021 6:03
1038	temp_hum	Outside	3.5	98.1	12.54	5.4.2021 7:03
1039	temp_hum	Outside	3.3	98.2	12.54	5.4.2021 8:03
1040	temp_hum	Outside	3.4	98.1	12.58	5.4.2021 9:03
1041	temp_hum	Outside	3.5	98.1	12.59	5.4.2021 10:03
1042	temp_hum	Outside	3.2	98.3	12.60	5.4.2021 11:03
1043	temp_hum	Outside	3.9	97.8	12.82	5.4.2021 12:03
1044	temp_hum	Outside	3.8	94.3	12.77	5.4.2021 13:03
1045	temp_hum	Outside	3.9	92.3	12.82	5.4.2021 14:03
1046	temp_hum	Outside	3.1	88.7	12.74	5.4.2021 15:03
1047	temp_hum	Outside	3.1	88.9	12.74	5.4.2021 15:20
1048	temp_hum	Outside	3.1	86.5	12.70	5.4.2021 15:22
1049	temp_hum	Outside	3.3	88.9	12.72	5.4.2021 15:31
1050	temp_hum	Outside	3.4	91.1	12.79	5.4.2021 16:02
1051	temp_hum	Outside	3.1	88.3	12.78	5.4.2021 17:03
1052	temp_hum	Outside	3.0	89.3	12.70	5.4.2021 17:05
1053	temp_hum	Outside	3.7	77.9	12.83	5.4.2021 18:03
1054	temp_hum	Outside	2.5	73.6	12.64	5.4.2021 19:03
1055	temp_hum	Outside	2.4	73.7	12.58	5.4.2021 19:05
1056	temp_hum	Outside	1.4	78.9	12.53	5.4.2021 20:03
1057	temp_hum	Outside	0.7	74.8	12.50	5.4.2021 21:03
1058	temp_hum	Outside	0.2	98.9	12.38	6.4.2021 7:01
1059	temp_hum	Outside	0.4	99.1	12.36	6.4.2021 8:03
1060	temp_hum	Outside	1.2	96.4	12.38	6.4.2021 9:03
1061	temp_hum	Outside	1.2	97.2	12.34	6.4.2021 9:05
1062	temp_hum	Outside	3.0	91.4	12.83	6.4.2021 10:02
1063	temp_hum	Outside	2.6	88.2	12.83	6.4.2021 11:03
1064	temp_hum	Outside	3.4	83.4	12.83	6.4.2021 12:03
1065	temp_hum	Outside	5.0	70.6	12.83	6.4.2021 13:03
1066	temp_hum	Outside	5.2	70.1	12.83	6.4.2021 14:04
1067	temp_hum	Outside	4.9	68.4	12.83	6.4.2021 14:05
1068	temp_hum	Outside	1.8	95.3	12.83	6.4.2021 15:03

## Jakso 4

ID	sensor	location	Temperature	Humidity	Battery	DateTime
1220	temp_hum	Outside	0.7	95.0	12.45	28.4.2021 23:02
1221	temp_hum	Outside	1.3	95.3	12.43	29.4.2021 0:03
1222	temp_hum	Outside	0.8	98.8	12.40	29.4.2021 1:03
1223	temp_hum	Outside	-0.3	99.9	12.38	29.4.2021 2:03
1224	temp_hum	Outside	-1.1	99.9	12.35	29.4.2021 3:02
1225	temp_hum	Outside	-1.5	99.9	12.31	29.4.2021 5:03
1226	temp_hum	Outside	-2.1	99.9	12.30	29.4.2021 6:03
1227	temp_hum	Outside	-0.9	99.9	12.30	29.4.2021 7:03
1228	temp_hum	Outside	0.9	99.4	12.30	29.4.2021 8:03
1229	temp_hum	Outside	5.0	83.1	12.30	29.4.2021 9:03
1230	temp_hum	Outside	3.9	85.2	12.35	29.4.2021 10:02
1231	temp_hum	Outside	4.8	76.7	12.59	29.4.2021 11:03
1232	temp_hum	Outside	8.0	57.6	12.83	29.4.2021 12:03
1233	temp_hum	Outside	8.5	43.0	12.83	29.4.2021 13:03
1234	temp_hum	Outside	9.6	38.5	12.83	29.4.2021 14:03
1235	temp_hum	Outside	10.3	39.4	12.83	29.4.2021 15:03
1236	temp_hum	Outside	8.6	39.7	12.83	29.4.2021 16:03
1237	temp_hum	Outside	11.1	31.0	12.83	29.4.2021 18:03
1238	temp_hum	Outside	7.4	40.1	12.83	29.4.2021 19:03
1239	temp_hum	Outside	6.0	49.5	12.75	29.4.2021 20:03
1240	temp_hum	Outside	4.7	56.0	12.70	29.4.2021 21:03
1241	temp_hum	Outside	2.6	73.5	12.66	29.4.2021 22:03
1242	temp_hum	Outside	1.7	81.7	12.64	29.4.2021 23:03
1243	temp_hum	Outside	0.1	90.0	12.61	30.4.2021 0:03
1244	temp_hum	Outside	-1.1	92.7	12.59	30.4.2021 1:03
1245	temp_hum	Outside	-2.1	95.2	12.56	30.4.2021 2:03
1246	temp_hum	Outside	-3.0	95.9	12.54	30.4.2021 3:03
1247	temp_hum	Outside	-2.8	96.0	12.51	30.4.2021 4:03
1248	temp_hum	Outside	-3.0	97.2	12.50	30.4.2021 5:03
1249	temp_hum	Outside	-2.8	97.8	12.48	30.4.2021 6:02
1250	temp_hum	Outside	-1.3	96.1	12.48	30.4.2021 7:02
1251	temp_hum	Outside	6.0	70.4	12.49	30.4.2021 9:03
1252	temp_hum	Outside	6.0	62.7	12.55	30.4.2021 10:03
1253	temp_hum	Outside	7.5	54.8	12.83	30.4.2021 11:03
1254	temp_hum	Outside	9.8	32.6	12.83	30.4.2021 12:03
1255	temp_hum	Outside	10.8	28.6	12.83	30.4.2021 13:04
1256	temp_hum	Outside	9.3	32.9	12.83	30.4.2021 14:03
1257	temp_hum	Outside	11.2	31.8	12.83	30.4.2021 15:03
1258	temp_hum	Outside	9.0	36.1	12.83	30.4.2021 17:04



1259	temp_hum	Outside	8.6	37.4	12.83	30.4.2021 18:03
1260	temp_hum	Outside	8.6	37.7	12.83	30.4.2021 19:03
1261	temp_hum	Outside	7.7	41.0	12.80	30.4.2021 20:03
1262	temp_hum	Outside	6.0	49.0	12.75	30.4.2021 21:03
1263	temp_hum	Outside	5.3	55.0	12.72	30.4.2021 22:03
1264	temp_hum	Outside	0.9	85.4	12.65	1.5.2021 0:03
1265	temp_hum	Outside	-0.3	90.0	12.63	1.5.2021 1:03
1266	temp_hum	Outside	-1.2	91.9	12.60	1.5.2021 2:03
1267	temp_hum	Outside	-2.5	95.3	12.55	1.5.2021 4:03
1268	temp_hum	Outside	-2.6	95.4	12.53	1.5.2021 5:02
1269	temp_hum	Outside	-2.5	96.2	12.51	1.5.2021 6:03
1270	temp_hum	Outside	-0.6	93.2	12.51	1.5.2021 7:03
1271	temp_hum	Outside	1.8	90.2	12.51	1.5.2021 8:03
1272	temp_hum	Outside	6.3	70.0	12.54	1.5.2021 9:03
1273	temp_hum	Outside	7.0	62.4	12.64	1.5.2021 10:03
1274	temp_hum	Outside	6.5	60.6	12.77	1.5.2021 11:03
1275	temp_hum	Outside	7.5	56.6	12.83	1.5.2021 12:03
1276	temp_hum	Outside	10.1	41.1	12.83	1.5.2021 13:04
1277	temp_hum	Outside	9.4	36.4	12.83	1.5.2021 14:03
1278	temp_hum	Outside	7.8	40.1	12.83	1.5.2021 15:03
1279	temp_hum	Outside	8.7	34.1	12.83	1.5.2021 16:04
1280	temp_hum	Outside	8.1	36.4	12.83	1.5.2021 17:03
1281	temp_hum	Outside	9.4	34.3	12.83	1.5.2021 18:04
1282	temp_hum	Outside	7.6	36.4	12.83	1.5.2021 19:03
1283	temp_hum	Outside	6.7	40.2	12.79	1.5.2021 20:04
1284	temp_hum	Outside	5.7	44.9	12.74	1.5.2021 21:03
1285	temp_hum	Outside	3.6	57.1	12.70	1.5.2021 22:03
1286	temp_hum	Outside	2.2	61.6	12.66	1.5.2021 23:03
1287	temp_hum	Outside	1.0	69.9	12.63	2.5.2021 0:02
1288	temp_hum	Outside	-0.9	82.1	12.60	2.5.2021 1:03
1289	temp_hum	Outside	-2.0	89.3	12.56	2.5.2021 2:03
1290	temp_hum	Outside	-1.9	89.5	12.53	2.5.2021 3:03
1291	temp_hum	Outside	-1.9	90.1	12.50	2.5.2021 4:03
1292	temp_hum	Outside	-2.4	91.5	12.48	2.5.2021 5:03
1293	temp_hum	Outside	-0.9	88.7	12.48	2.5.2021 6:03
1294	temp_hum	Outside	0.9	82.4	12.49	2.5.2021 7:03
1295	temp_hum	Outside	2.4	71.2	12.48	2.5.2021 8:03
1296	temp_hum	Outside	7.1	42.2	12.49	2.5.2021 9:03
1297	temp_hum	Outside	6.3	37.0	12.58	2.5.2021 10:03
1298	temp_hum	Outside	7.1	31.1	12.83	2.5.2021 11:03
1299	temp_hum	Outside	10.5	27.2	12.83	2.5.2021 12:03
1300	temp_hum	Outside	11.2	26.1	12.83	2.5.2021 13:03
1301	temp_hum	Outside	12.4	25.7	12.83	2.5.2021 14:03

1302	temp_hum	Outside	10.8	26.6	12.83	2.5.2021 15:03
1303	temp_hum	Outside	12.8	24.4	12.83	2.5.2021 16:04
1304	temp_hum	Outside	10.1	28.7	12.83	2.5.2021 17:04
1305	temp_hum	Outside	12.6	25.3	12.83	2.5.2021 18:04
1306	temp_hum	Outside	10.9	26.9	12.83	2.5.2021 19:03
1307	temp_hum	Outside	8.7	31.5	12.83	2.5.2021 20:03
1308	temp_hum	Outside	7.4	37.9	12.83	2.5.2021 21:03
1309	temp_hum	Outside	4.5	52.2	12.80	2.5.2021 22:02
1310	temp_hum	Outside	2.3	64.7	12.77	2.5.2021 23:03
1311	temp_hum	Outside	1.0	74.8	12.74	3.5.2021 0:03
1312	temp_hum	Outside	-0.4	84.6	12.71	3.5.2021 1:03
1313	temp_hum	Outside	-1.0	90.0	12.68	3.5.2021 2:03
1314	temp_hum	Outside	-0.9	89.9	12.66	3.5.2021 3:03
1315	temp_hum	Outside	-1.7	92.1	12.64	3.5.2021 4:02
1316	temp_hum	Outside	-1.4	90.9	12.61	3.5.2021 5:03
1317	temp_hum	Outside	0.0	85.3	12.60	3.5.2021 6:03
1318	temp_hum	Outside	1.1	78.6	12.61	3.5.2021 7:03
1319	temp_hum	Outside	4.6	57.4	12.70	3.5.2021 8:03
1320	temp_hum	Outside	6.3	37.7	12.82	3.5.2021 9:03
1321	temp_hum	Outside	8.1	33.5	12.83	3.5.2021 10:03
1322	temp_hum	Outside	8.6	29.3	12.83	3.5.2021 11:03
1323	temp_hum	Outside	9.5	27.1	12.83	3.5.2021 12:04
1324	temp_hum	Outside	9.7	28.4	12.83	3.5.2021 13:03
1325	temp_hum	Outside	10.7	25.7	12.83	3.5.2021 14:03
1326	temp_hum	Outside	10.8	25.7	12.83	3.5.2021 15:04
1327	temp_hum	Outside	10.4	27.1	12.83	3.5.2021 17:03
1328	temp_hum	Outside	9.9	30.8	12.83	3.5.2021 18:04
1329	temp_hum	Outside	9.3	33.7	12.83	3.5.2021 19:03
1330	temp_hum	Outside	8.3	33.2	12.83	3.5.2021 20:03
1331	temp_hum	Outside	7.4	39.0	12.83	3.5.2021 21:03
1332	temp_hum	Outside	6.5	41.2	12.80	3.5.2021 22:03
1333	temp_hum	Outside	6.0	43.7	12.78	3.5.2021 23:03
1334	temp_hum	Outside	5.3	45.0	12.74	4.5.2021 0:03
1335	temp_hum	Outside	4.4	48.1	12.72	4.5.2021 1:03
1336	temp_hum	Outside	3.4	49.7	12.69	4.5.2021 2:03
1337	temp_hum	Outside	3.0	52.8	12.65	4.5.2021 3:03
1338	temp_hum	Outside	2.5	59.4	12.63	4.5.2021 4:03
1339	temp_hum	Outside	2.6	65.0	12.60	4.5.2021 6:03
1340	temp_hum	Outside	3.8	60.8	12.65	4.5.2021 7:03
1341	temp_hum	Outside	5.4	51.7	12.69	4.5.2021 8:03
1342	temp_hum	Outside	9.0	38.2	12.77	4.5.2021 9:03
1343	temp_hum	Outside	9.4	37.1	12.83	4.5.2021 10:03
1344	temp_hum	Outside	9.8	36.5	12.83	4.5.2021 11:03

1345	temp_hum	Outside	11.8	31.3	12.83	4.5.2021 12:03
1346	temp_hum	Outside	11.3	33.4	12.83	4.5.2021 13:04
1347	temp_hum	Outside	10.8	35.7	12.83	4.5.2021 15:03
1348	temp_hum	Outside	10.7	37.6	12.83	4.5.2021 16:03
1349	temp_hum	Outside	10.0	39.3	12.83	4.5.2021 17:04
1350	temp_hum	Outside	9.4	42.0	12.83	4.5.2021 18:03
1351	temp_hum	Outside	8.0	49.7	12.83	4.5.2021 19:04
1352	temp_hum	Outside	7.6	54.1	12.83	4.5.2021 20:03
1353	temp_hum	Outside	5.9	66.3	12.82	4.5.2021 21:03
1354	temp_hum	Outside	5.2	73.6	12.79	4.5.2021 22:03
1355	temp_hum	Outside	5.1	75.4	12.77	4.5.2021 23:03
1356	temp_hum	Outside	4.7	76.8	12.71	5.5.2021 1:03
1357	temp_hum	Outside	4.2	75.7	12.68	5.5.2021 2:03
1358	temp_hum	Outside	3.8	81.5	12.65	5.5.2021 3:03
1359	temp_hum	Outside	3.4	87.1	12.63	5.5.2021 4:03
1360	temp_hum	Outside	3.1	90.6	12.61	5.5.2021 5:03
1361	temp_hum	Outside	2.4	92.7	12.60	5.5.2021 6:03
1362	temp_hum	Outside	2.6	90.7	12.63	5.5.2021 7:03
1363	temp_hum	Outside	3.2	89.5	12.64	5.5.2021 8:03
1364	temp_hum	Outside	4.5	80.2	12.78	5.5.2021 9:03
1365	temp_hum	Outside	4.9	73.2	12.83	5.5.2021 10:03
1366	temp_hum	Outside	5.2	71.6	12.83	5.5.2021 11:03
1367	temp_hum	Outside	5.8	65.9	12.83	5.5.2021 12:03
1368	temp_hum	Outside	6.6	62.1	12.83	5.5.2021 13:03
1369	temp_hum	Outside	5.8	70.4	12.83	5.5.2021 14:03
1370	temp_hum	Outside	4.9	85.6	12.83	5.5.2021 15:03
1371	temp_hum	Outside	2.6	97.4	12.83	5.5.2021 16:03
1372	temp_hum	Outside	1.9	99.1	12.77	5.5.2021 17:03
1373	temp_hum	Outside	2.2	99.3	12.69	5.5.2021 18:03
1374	temp_hum	Outside	1.6	99.9	12.60	5.5.2021 20:02
1375	temp_hum	Outside	1.9	99.9	12.59	5.5.2021 21:03
1376	temp_hum	Outside	2.3	99.9	12.58	5.5.2021 22:03
1377	temp_hum	Outside	2.8	99.9	12.56	5.5.2021 23:03
1378	temp_hum	Outside	3.2	99.9	12.55	6.5.2021 0:03
1379	temp_hum	Outside	3.5	99.9	12.55	6.5.2021 1:03
1380	temp_hum	Outside	3.9	99.9	12.54	6.5.2021 2:03
1381	temp_hum	Outside	4.1	99.9	12.54	6.5.2021 3:03
1382	temp_hum	Outside	4.1	99.9	12.53	6.5.2021 4:03
1383	temp_hum	Outside	4.0	99.9	12.51	6.5.2021 5:03
1384	temp_hum	Outside	4.4	99.9	12.53	6.5.2021 6:03
1385	temp_hum	Outside	4.6	99.9	12.53	6.5.2021 7:03
1386	temp_hum	Outside	5.2	99.9	12.54	6.5.2021 8:03
1387	temp_hum	Outside	5.0	99.9	12.53	6.5.2021 9:03

1388	temp_hum	Outside	5.2	99.9	12.55	6.5.2021 10:03
1389	temp_hum	Outside	5.9	97.7	12.65	6.5.2021 11:03
1390	temp_hum	Outside	5.8	97.4	12.66	6.5.2021 12:03
1391	temp_hum	Outside	6.9	94.4	12.78	6.5.2021 13:03
1392	temp_hum	Outside	7.0	93.5	12.65	6.5.2021 14:04
1393	temp_hum	Outside	8.2	88.8	12.83	6.5.2021 15:03
1394	temp_hum	Outside	8.2	85.5	12.82	6.5.2021 16:03
1395	temp_hum	Outside	6.9	90.0	12.65	6.5.2021 18:02
1396	temp_hum	Outside	5.5	93.2	12.55	6.5.2021 19:03
1397	temp_hum	Outside	4.7	94.5	12.53	6.5.2021 20:03
1398	temp_hum	Outside	4.3	94.6	12.51	6.5.2021 21:03
1399	temp_hum	Outside	3.9	94.3	12.50	6.5.2021 22:03
1400	temp_hum	Outside	3.5	95.8	12.48	7.5.2021 0:03
1401	temp_hum	Outside	3.3	93.2	12.46	7.5.2021 1:03
1402	temp_hum	Outside	3.2	94.2	12.45	7.5.2021 2:03
1403	temp_hum	Outside	3.0	94.3	12.44	7.5.2021 3:03
1404	temp_hum	Outside	2.8	93.2	12.43	7.5.2021 4:04
1405	temp_hum	Outside	2.7	93.9	12.42	7.5.2021 5:03
1406	temp_hum	Outside	2.5	96.1	12.41	7.5.2021 6:03
1407	temp_hum	Outside	2.4	96.6	12.43	7.5.2021 7:03
1408	temp_hum	Outside	4.4	91.6	12.56	7.5.2021 8:03
1409	temp_hum	Outside	4.9	91.1	12.58	7.5.2021 9:03
1410	temp_hum	Outside	5.7	91.2	12.79	7.5.2021 10:03
1411	temp_hum	Outside	5.8	95.0	12.82	7.5.2021 11:04
1412	temp_hum	Outside	5.3	96.9	12.68	7.5.2021 12:03
1413	temp_hum	Outside	5.7	98.5	12.65	7.5.2021 13:03
1414	temp_hum	Outside	5.7	98.7	12.58	7.5.2021 14:03
1415	temp_hum	Outside	7.0	95.5	12.73	7.5.2021 15:03
1416	temp_hum	Outside	7.3	95.9	12.66	7.5.2021 16:04
1417	temp_hum	Outside	7.4	92.8	12.54	7.5.2021 17:03
1418	temp_hum	Outside	7.6	96.0	12.66	7.5.2021 18:03
1419	temp_hum	Outside	6.5	93.4	12.55	7.5.2021 19:03
1420	temp_hum	Outside	5.1	94.6	12.44	7.5.2021 21:03
1421	temp_hum	Outside	3.8	99.5	12.43	7.5.2021 22:03
1422	temp_hum	Outside	3.6	99.7	12.41	7.5.2021 23:03
1423	temp_hum	Outside	3.0	99.2	12.40	8.5.2021 0:03
1424	temp_hum	Outside	2.7	98.9	12.39	8.5.2021 1:03
1425	temp_hum	Outside	2.5	99.9	12.38	8.5.2021 2:03
1426	temp_hum	Outside	2.2	99.9	12.36	8.5.2021 3:03
1427	temp_hum	Outside	2.3	99.9	12.35	8.5.2021 4:03
1428	temp_hum	Outside	2.3	99.9	12.34	8.5.2021 5:03
1429	temp_hum	Outside	2.2	99.9	12.34	8.5.2021 6:03
1430	temp_hum	Outside	2.8	99.9	12.34	8.5.2021 7:02

1431	temp_hum	Outside	3.2	99.7	12.35	8.5.2021 8:03
1432	temp_hum	Outside	4.5	93.6	12.60	8.5.2021 10:03
1433	temp_hum	Outside	6.6	86.3	12.83	8.5.2021 11:03
1434	temp_hum	Outside	8.0	67.6	12.83	8.5.2021 12:03
1435	temp_hum	Outside	11.5	39.7	12.83	8.5.2021 14:04
1436	temp_hum	Outside	12.1	37.8	12.83	8.5.2021 15:03
1437	temp_hum	Outside	9.8	38.1	12.83	8.5.2021 16:03
1438	temp_hum	Outside	10.5	39.2	12.83	8.5.2021 17:04
1439	temp_hum	Outside	7.0	69.3	12.83	8.5.2021 18:04
1440	temp_hum	Outside	7.8	61.6	12.83	8.5.2021 20:03
1441	temp_hum	Outside	5.4	78.9	12.80	8.5.2021 21:03
1442	temp_hum	Outside	3.4	90.8	12.77	8.5.2021 22:03
1443	temp_hum	Outside	1.6	94.2	12.73	8.5.2021 23:03
1444	temp_hum	Outside	0.6	94.3	12.70	9.5.2021 0:03
1445	temp_hum	Outside	-0.3	94.9	12.68	9.5.2021 1:03
1446	temp_hum	Outside	-0.8	96.5	12.65	9.5.2021 2:03
1447	temp_hum	Outside	-1.4	98.1	12.63	9.5.2021 3:03
1448	temp_hum	Outside	-2.0	99.9	12.59	9.5.2021 5:02
1449	temp_hum	Outside	-1.8	99.9	12.58	9.5.2021 6:03
1450	temp_hum	Outside	0.7	96.6	12.58	9.5.2021 7:03
1451	temp_hum	Outside	3.6	89.4	12.63	9.5.2021 8:03
1452	temp_hum	Outside	6.0	74.0	12.65	9.5.2021 9:03
1453	temp_hum	Outside	8.2	62.8	12.73	9.5.2021 10:03
1454	temp_hum	Outside	9.5	56.0	12.83	9.5.2021 11:03
1455	temp_hum	Outside	10.5	67.3	12.83	9.5.2021 12:03
1456	temp_hum	Outside	12.3	70.8	12.83	9.5.2021 13:03
1457	temp_hum	Outside	12.7	72.2	12.83	9.5.2021 14:03
1458	temp_hum	Outside	14.6	43.3	12.83	9.5.2021 15:04
1459	temp_hum	Outside	13.4	40.3	12.83	9.5.2021 16:04
1460	temp_hum	Outside	12.8	35.7	12.83	9.5.2021 17:04
1461	temp_hum	Outside	10.9	57.8	12.83	9.5.2021 19:03
1462	temp_hum	Outside	9.3	63.2	12.83	9.5.2021 20:03
1463	temp_hum	Outside	8.4	76.0	12.83	9.5.2021 21:03
1464	temp_hum	Outside	7.8	85.0	12.82	9.5.2021 22:03
1465	temp_hum	Outside	6.1	96.0	12.77	10.5.2021 0:03
1466	temp_hum	Outside	5.5	99.4	12.74	10.5.2021 1:03
1467	temp_hum	Outside	5.6	99.9	12.72	10.5.2021 2:03
1468	temp_hum	Outside	5.7	99.9	12.70	10.5.2021 3:03
1469	temp_hum	Outside	5.7	99.9	12.68	10.5.2021 4:03
1470	temp_hum	Outside	5.8	99.9	12.66	10.5.2021 5:03
1471	temp_hum	Outside	5.9	99.9	12.65	10.5.2021 6:03
1472	temp_hum	Outside	6.3	99.9	12.65	10.5.2021 7:03
1473	temp_hum	Outside	7.7	99.4	12.70	10.5.2021 8:03

(12)

1474	temp_hum	Outside	8.6	97.3	12.72	10.5.2021 9:03
1475	temp_hum	Outside	8.8	99.2	12.72	10.5.2021 10:03
1476	temp_hum	Outside	9.4	99.0	12.73	10.5.2021 11:03
1477	temp_hum	Outside	9.7	99.4	12.70	10.5.2021 12:03
1478	temp_hum	Outside	10.0	98.1	12.72	10.5.2021 13:03
1479	temp_hum	Outside	10.8	95.5	12.83	10.5.2021 14:03
1480	temp_hum	Outside	12.1	93.1	12.83	10.5.2021 15:04
1481	temp_hum	Outside	12.4	93.3	12.75	10.5.2021 16:03
1482	temp_hum	Outside	17.0	80.6	12.83	10.5.2021 17:03
1483	temp_hum	Outside	17.3	73.7	12.83	10.5.2021 18:04
1484	temp_hum	Outside	18.6	71.3	12.83	10.5.2021 19:03
1485	temp_hum	Outside	16.3	77.0	12.83	10.5.2021 20:03
1486	temp_hum	Outside	14.9	82.9	12.75	10.5.2021 21:03
1487	temp_hum	Outside	12.2	91.3	12.69	10.5.2021 22:03
1488	temp_hum	Outside	10.9	92.5	12.66	10.5.2021 23:03
1489	temp_hum	Outside	10.4	92.7	12.64	11.5.2021 0:03
1490	temp_hum	Outside	10.0	92.9	12.63	11.5.2021 1:03
1491	temp_hum	Outside	9.6	92.7	12.60	11.5.2021 2:03
1492	temp_hum	Outside	9.3	93.2	12.59	11.5.2021 3:03
1493	temp_hum	Outside	8.4	95.2	12.58	11.5.2021 4:03
1494	temp_hum	Outside	7.6	96.4	12.56	11.5.2021 5:03
1495	temp_hum	Outside	8.5	95.6	12.56	11.5.2021 6:03
1496	temp_hum	Outside	11.8	88.6	12.56	11.5.2021 7:03
1497	temp_hum	Outside	13.6	81.2	12.58	11.5.2021 8:03
1498	temp_hum	Outside	17.2	67.5	12.60	11.5.2021 9:03
1499	temp_hum	Outside	17.7	65.2	12.68	11.5.2021 10:04
1500	temp_hum	Outside	23.6	51.0	12.83	11.5.2021 12:03
1501	temp_hum	Outside	24.1	44.2	12.83	11.5.2021 13:04
1502	temp_hum	Outside	23.0	42.9	12.83	11.5.2021 14:03
1503	temp_hum	Outside	23.9	42.6	12.83	11.5.2021 15:04
1504	temp_hum	Outside	25.0	37.9	12.83	11.5.2021 16:04
1505	temp_hum	Outside	25.6	34.5	12.83	11.5.2021 17:04
1506	temp_hum	Outside	26.5	33.7	12.83	11.5.2021 18:03
1507	temp_hum	Outside	25.7	33.8	12.83	11.5.2021 19:03
1508	temp_hum	Outside	22.2	39.0	12.83	11.5.2021 20:03
1509	temp_hum	Outside	19.4	50.7	12.83	11.5.2021 21:03
1510	temp_hum	Outside	16.7	63.8	12.83	11.5.2021 22:03
1511	temp_hum	Outside	15.3	71.3	12.83	11.5.2021 23:03
1512	temp_hum	Outside	14.5	73.5	12.83	12.5.2021 0:03
1513	temp_hum	Outside	13.4	75.3	12.83	12.5.2021 1:03
1514	temp_hum	Outside	11.4	84.4	12.83	12.5.2021 2:03
1515	temp_hum	Outside	9.9	90.6	12.83	12.5.2021 3:03

(12)

1516	temp_hum	Outside	9.1	92.2	12.80	12.5.2021 4:03
1517	temp_hum	Outside	9.3	91.1	12.78	12.5.2021 5:03
1518	temp_hum	Outside	10.4	83.9	12.78	12.5.2021 6:03
1519	temp_hum	Outside	12.9	70.3	12.78	12.5.2021 7:03
1520	temp_hum	Outside	15.2	64.2	12.78	12.5.2021 8:04
1521	temp_hum	Outside	19.4	47.6	12.79	12.5.2021 9:03
1522	temp_hum	Outside	19.8	45.2	12.83	12.5.2021 10:03
1523	temp_hum	Outside	22.0	35.9	12.83	12.5.2021 11:03
1524	temp_hum	Outside	25.7	30.4	12.83	12.5.2021 12:04
1525	temp_hum	Outside	26.0	25.2	12.83	12.5.2021 13:04
1526	temp_hum	Outside	27.1	23.5	12.83	12.5.2021 14:03
1527	temp_hum	Outside	27.3	25.5	12.83	12.5.2021 15:04
1528	temp_hum	Outside	28.2	25.2	12.83	12.5.2021 16:04
1529	temp_hum	Outside	27.3	27.2	12.83	12.5.2021 17:04
1530	temp_hum	Outside	27.4	28.2	12.83	12.5.2021 18:04
1531	temp_hum	Outside	26.3	29.9	12.83	12.5.2021 19:03
1532	temp_hum	Outside	23.0	37.7	12.83	12.5.2021 20:03
1533	temp_hum	Outside	21.1	40.1	12.83	12.5.2021 21:03
1534	temp_hum	Outside	18.3	50.0	12.83	12.5.2021 22:04
1535	temp_hum	Outside	16.7	57.2	12.83	12.5.2021 23:03
1536	temp_hum	Outside	15.3	61.1	12.83	13.5.2021 0:03
1537	temp_hum	Outside	14.4	66.6	12.83	13.5.2021 1:03
1538	temp_hum	Outside	13.8	68.2	12.83	13.5.2021 2:03
1539	temp_hum	Outside	13.2	72.7	12.83	13.5.2021 3:03
1540	temp_hum	Outside	12.8	75.2	12.83	13.5.2021 4:03
1541	temp_hum	Outside	12.6	77.2	12.83	13.5.2021 5:03
1542	temp_hum	Outside	12.2	81.0	12.83	13.5.2021 6:03
1543	temp_hum	Outside	14.3	71.3	12.83	13.5.2021 7:03
1544	temp_hum	Outside	16.6	64.9	12.83	13.5.2021 8:03
1545	temp_hum	Outside	20.7	54.9	12.83	13.5.2021 9:03
1546	temp_hum	Outside	22.3	51.0	12.83	13.5.2021 10:03
1547	temp_hum	Outside	24.3	41.0	12.83	13.5.2021 11:03
1548	temp_hum	Outside	27.5	33.5	12.83	13.5.2021 12:04
1549	temp_hum	Outside	27.6	32.9	12.83	13.5.2021 13:04
1550	temp_hum	Outside	28.4	29.4	12.83	13.5.2021 14:04
1551	temp_hum	Outside	28.1	30.0	12.83	13.5.2021 15:04
1552	temp_hum	Outside	27.5	31.5	12.83	13.5.2021 16:04
1553	temp_hum	Outside	26.2	33.9	12.83	13.5.2021 17:04
1554	temp_hum	Outside	25.0	37.2	12.83	13.5.2021 18:04
1555	temp_hum	Outside	23.6	43.4	12.83	13.5.2021 19:03
1556	temp_hum	Outside	21.1	53.4	12.83	13.5.2021 20:03
1557	temp_hum	Outside	19.5	59.0	12.83	13.5.2021 21:04

(12)

1558	temp_hum	Outside	17.1	71.2	12.83	13.5.2021 22:03
1559	temp_hum	Outside	15.3	75.3	12.83	13.5.2021 23:04
1560	temp_hum	Outside	13.6	80.6	12.83	14.5.2021 0:03
1561	temp_hum	Outside	12.0	86.8	12.83	14.5.2021 1:03
1562	temp_hum	Outside	10.7	90.4	12.83	14.5.2021 2:03
1563	temp_hum	Outside	9.8	92.7	12.83	14.5.2021 3:03
1564	temp_hum	Outside	9.3	94.6	12.83	14.5.2021 4:03
1565	temp_hum	Outside	8.9	96.4	12.82	14.5.2021 5:03
1566	temp_hum	Outside	9.2	96.5	12.80	14.5.2021 6:03