

Arduino mikrokontrolleri paikallisena verkkoserverinä



Ammattikorkeakoulututkinnon opinnäytetyö
Sähkö- ja automaatiotekniikka, insinööri (AMK)

Syksy 2023

Olli-Pekka Laukkanen

Sähkö- ja automaatiotekniikka, insinööri (AMK)

Tekijä Olli-Pekka Laukkanen

Työn nimi Arduino mikrokontrolleri paikallisena verkkoserverinä

Ohjaaja Juha Sarkula

Tiivistelmä

Vuosi 2023

Tämän opinnäytetyön tarkoituksena oli toteuttaa Arduino-pohjainen verkkoserveri, jota voidaan käyttää erilaisten mittausarvojen tarkkailuun ajoneuvossa, joka toimii koirien oleskelutilana pitkillä matkoilla. Ajoneuvon sisälämpötilan tarkkailu etänä paikallisverkossa olevan tiedon voi käydä lukemassa sijainnista riippumatta. Ulkoista virtalähdettä käyttäessä Arduino voitiin sijoittaa haluttuun paikkaan WiFi -yhteyden kantaman sisällä. Tässä projektissa WiFi -yhteys luotiin Android tabletilla, joten WiFi-yhteys on myös täysin mobilisoitavissa. Projektissa käytettyyn Android tablettiin otettiin yhteys TeamViewer ohjelmalla, jolla saadaan yhteys etänä, tämä mahdollistaa Arduinon etäkäytön sijainnista riippumatta. Työssä tutustuttiin Arduino-kehitysalustaan, C-, C++- sekä HTML -kieliin ja etäyhteyksiin. Opinnäytetyö on tuotettu omaan käyttöön.

Opinnäytetyön lopputuloksena luotiin Arduino pohjainen verkkoserveri, joka on muokattavissa halutulla tavalla toimimaan datankeruu alustana sekä mikrokontrollerin ohjaukseen soveltuvana verkkosivustona. Verkkoserveri on käytettävissä paikallisverkossa, VPN-yhteyden kautta sekä TeamViewerin avulla halutuilla laitteilla myös julkisen verkon kautta. HTML-sivustosta luotiin valmis pohja, jota voi hyödyntää tulevaisuudessa halutulla tavalla laitteistojen ohjaukseen tai mittaus arvojen tarkasteluun.

Avainsanat Ohjelmointi, arduino, elektroniikkakomponentit, elektroniikkalaitteet.

Sivut 30 sivua ja liitteitä 3 sivua

Electrical and Automation Engineering

Author Olli-Pekka Laukkanen

Subject Arduino microcontroller as local web server.

Supervisors Juha Sarkula

Abstract

Year 2023

The purpose of this thesis was to create Arduino based web server, that can be used to monitoring various measurement values in a vehicle functioning as a long-distance dog living space. Remote monitoring of the vehicle's interior temperature can be accessed from any location within a local network. when using an external power source, the Arduino could be placed in a desired location within the WiFi range. In this project, the WiFi connection was established using an Android tabled, making it fully mobile. A VPN was added to the android used in the project, allowing remote access to the arduino regardless of location. The VPN also serves as strong protection for the network connection. The thesis involved becoming familiar with the Arduino development platform, C-, C++, HTML -languages, and VPN connectivity. The thesis work was produced for personal use.

The arduino microcontoller is a circuit based on the ATMEL AVR microcontroller board. In Arduino, operations take place on communication ports as well as input and output ports, to which necessary additional devices such as sensors can be connected.

Keywords Programming, arduino, electronic components, electronic gadgets.

Pages 30 pages and appendices 3 pages

Sisällys

1	Johdanto	1
2	Arduino-mikrokontrolleri	2
2.1	Historia	2
2.2	Atmel	4
2.2.1	STK500	4
2.2.2	ATmega128	5
2.2.3	ATmega8	6
2.3	Toimintaperiaate	7
2.4	Arduino IDE	7
2.5	Kirjastot	9
2.6	DHT11 – Lämpötila- ja kosteusanturi	9
3	Ohjelmointikielät	10
3.1	C-ohjelmointikieli	10
3.2	C++-ohjelmointikieli	11
3.3	HTML-kieli	12
3.3.1	Rakenne	12
3.3.2	Erikoismerkit	12
4	WiFi – Langaton verkko	13
5	SSL-sertifikaatti	13
5.1	Extended Validation	14
5.2	Organization Validation	14
5.3	Domain Validation	15
5.4	Wildcard-sertifikaatti	15
6	Etäyhteys	15
6.1	VPN	15
6.1.1	Host-to-site VPN	16
6.1.2	Site-to-site	16
6.1.3	SSL VPN	17
6.1.4	VPN:n hyödyt	17
6.2	TeamViewer	17

7	Käytännön toteutus.....	18
7.1	Arduino WiFi Rev2.....	18
7.2	Kytkenät	18
7.3	Ohjelmointi	19
7.3.1	WiFi-yhteys.....	20
7.3.2	Turvallisuus.....	20
7.3.3	Ohjelmakoodi	21
7.4	Etäohjaus.....	23
7.4.1	VPN.....	24
7.4.2	TeamViewer	25
7.5	Toiminnallisuus	25
7.6	Testaus	28
8	Yhteenveto	28
	Lähdeluettelo.....	29

Kuvat, taulukot ja kaavat

Kuva 1.	Ensimmäinen virallinen Arduino.....	4
Kuva 2.	ATmega128 mikrosiru (Microchip, ATmega128, n.d.-a).....	5
Kuva 3.	Atmega8 mikrosiru (Microchip, ATmega8, n.d.-b)	6
Kuva 4.	Arduino IDE:n oletusikkuna (Olli-Pekka Laukkanen, 2023)	8
Kuva 5.	DHT11 Lämpötila- ja kosteusanturi (Olli-Pekka Laukkanen, 2023)	10
Kuva 6	SSL-sertifikaatin tarkastus ikkuna (Olli-Pekka Laukkanen, 2023)	14
Kuva 7.	DHT11 anturin kytkentä mikrokontrolleriin (Olli-Pekka Laukkanen, 2023).....	19
Kuva 8.	Arduino Ide Tools alasvetovalikko	20

Kuva 9. Ohjelman toimintadiagrammi (Olli-Pekka Laukkanen, 2023)	23
Kuva 10 Serverin toiminta (Olli-Pekka Laukkanen, 2023).	26
Kuva 11 Toiminnallinen painike aktivoimatta (Olli-Pekka Laukkanen, 2023).	27
Kuva 12 Toiminnallinen painike aktivoitu (Olli-Pekka Laukkanen, 2023).	27

Liitteet

Liite 1. Arduino koodi

1 Johdanto

Tämän opinnäytetyön tarkoituksena on toteuttaa Arduino-pohjainen verkkoserveri, joka ylläpitää HTML sivua paikallisverkossa. Verkkosivun tavoitteena on luoda valmis pohja mikrokontrollerin ohjaukseen. Projektia käytetään myös ajoneuvon lämpötilan tarkkailuun ajoneuvon ollessa parkissa. Ajoneuvossa on matkojen aikana koiria, joten lämpötilan tarkkailu on kriittinen osa eläinten turvallisuuden kannalta. Paikallisverkossa olevassa HTML-sivussa olevia tietoja voi käydä lukemassa sijainnista riippumatta. Valituilla laitteilla, joihin luodaan etäyhteys, voidaan lukea tiedot sijainnista riippumatta. Ulkoista virtalähdettä käyttäessä Arduino voidaan sijoittaa haluttuun paikkaan WiFi-yhteyden kantaman sisällä. Laajemman käyttöalueen näkökulmasta työssä tutustutaan VPN-yhteyteen, sekä TeamViewerin etäyhteyden hyödyntämiseen. Työssä tutustutaan myös Arduino-kehitysalustaan ja C-, C++-, HTML-kieliin sekä tietoturvaan paikallisessa verkkoyhteydessä. Simuloitaessa paikallista verkkosivua käytetään DHT11 -lämpötila- ja kosteusanturia tulostamaan mittausdataa sivulle.

Työ toteutetaan itsenäisesti henkilökohtaisena projektina ja tarkoitus jatkojalostaa kiinteistöautomaatioon soveltuvaksi kokonaisuudeksi. Tavoitteena on hyödyntää mahdollisimman laajasti jo opittuja asioita, sekä oppia elektroniikasta, sen toiminnasta, ohjelmoinnista sekä langattomista verkoista.

2 Arduino-mikrokontrolleri

Arduino on avoimen lähdekoodin kehitysalusta, jota ohjelmoidaan Arduino IDE -ohjelmistolla. Arduinosta puhutaan myös mikrokontrolleri nimellä. Arduinon avulla pystytään luomaan sulautettuja järjestelmiä eli tiettyyn ennalta määritettyyn toimintoon kehitetty laite, esimerkiksi hissien ohjauspiiri. Järjestelmää ohjataan koodilla, joka luodaan ja ladataan mikrokontrollerille Arduino IDE -ohjelmistolla. (Arduino-mikrokontrollerin ja koekytkentälevyn perusteet, n.d.)

Arduino laite perustuu 8 bittiseen Atmel AVR mikro -ohjaimeen. Laite on edullinen ja helppokäyttöinen myös aloittelijoille, mutta haastaa myös kokeneempia ohjelmoijia käyttämään laitetta. Arduinon hinta alkaa 30 eurosta, mutta tarvittaessa kaikki piirustukset ja lähdekoodit ovat saatavilla internetistä, tämä mahdollistaa oman alustan valmistamisen ”romusta”.

2.1 Historia

Arduino on avoimen lähdekoodin helppokäyttöinen kehitysalusta, jonka suunniteltu ja toteutus on aloitettu opiskelijan toimesta Interaction Design Institute Ivreassa vuonna 2003. (Armenta, 2022)

Hernando Barragán nimisen oppilaan diplomityön tarkoituksena oli luoda helppokäyttöinen kehitysalusta suunnittelijoille elektroniikan parissa poistamalla haastavat yksityiskohdat, jotta he voivat keskittyä omiin tavoitteisiinsa. Projektin nimi oli ”Wiring”. Projektiin kuului myös Massimo Banzi, sekä Casey Reas, joka tunnetaan ”Processing” ohjelmointikielestä.

Diplomityö sai paljon huomiota kyseisessä koulussa ja sitä on käytetty useissa muissa projekteissa vuodesta 2004 vuoteen 2005 kunnes koulu lakkautettiin. (Barragán, n.d.)

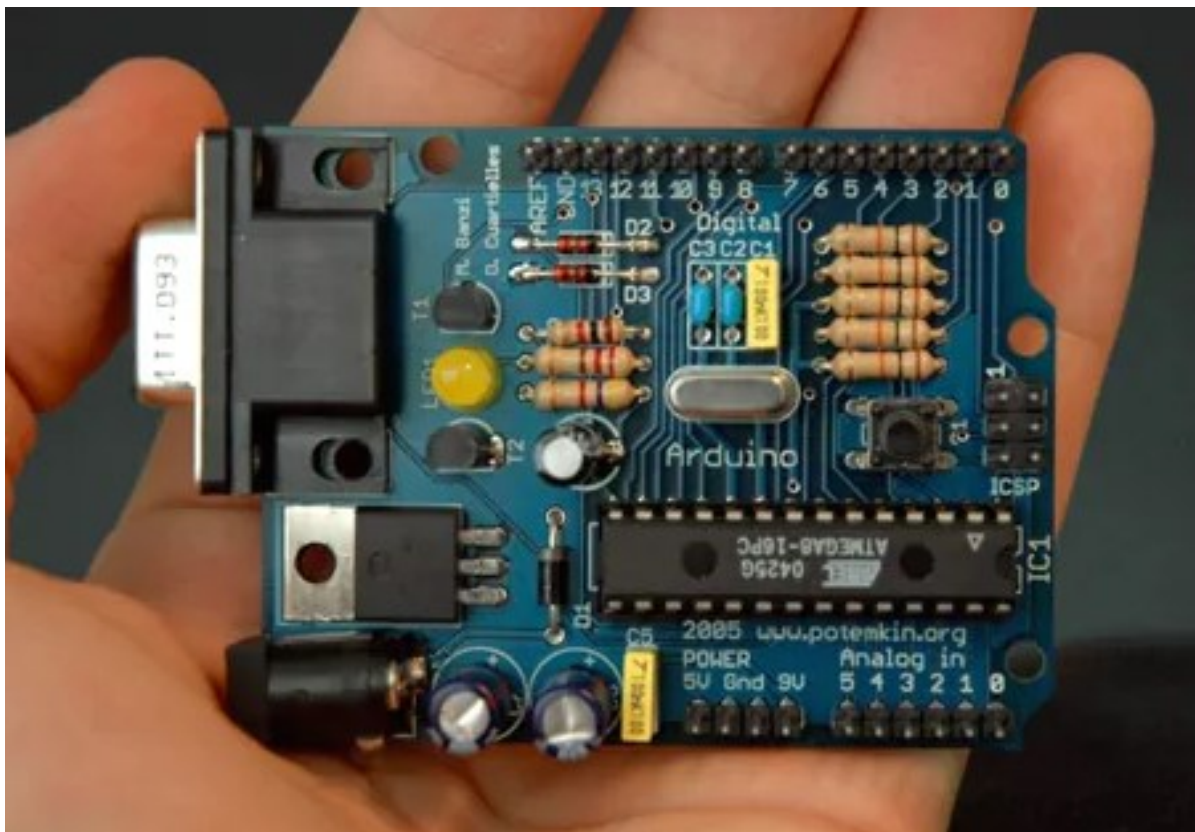
Ensimmäistä Wiring prototyyppiä yritettiin saada toimimaan Parallaxin valmistamalla Javelin Stamp mikrosirulla, mutta se ei ollut toteuttamiskelpoinen käyttäjien ohjelman lataamisen tuomien haasteiden vuoksi. (Barragán, n.d.)

Toisessa prototyypissä käytettiin Atmelin AMR -pohjaista AT91R40008 mikrosirua, joka johti erinomaisiin tuloksiin. Tällä kontrollerilla luotiin ensimmäiset esimerkit ohjelmistoon. (Barragán, n.d.)

Kolmas prototyyppi toteutettiin käyttämällä Atmelin ATmega128 mikrosirua. ATmega128 mikrosirua kokeiltiin Atmelin STK500:n kytkettynä, joka oli menestys. Hernando osti MAVRICin BDMICRO piirilevyn, johon oli kytketty ATmega128. Brian Deanin työstämä ohjelmisto, jolla hän latsin ohjelmia MAVRIC kontrollerille tunnetaan nykyisin nimellä avrdude (AVR Download/Upload). Com-porttien hävitessä tietokoneista Hernando valitsi FTDI:n levyn, joka kommunikoi Usb-portin kautta. (Barragán, n.d.)

Vuonna 2005 Massimo Banzi, David Mellis sekä David Cuartielles aloittivat uuden projektin, jossa käytettiin Wiring projektin lähdekoodia mutta vanhan mikrokontrollerin tilalle vaihdettiin ATmega8. Arduino projektitiimiin kuului lisäksi Gianluca Martino, joka auttoi tiimiä tuotannollisissa tehtävissä kontrollerien valmistuksessa, sekä Tom Igoe. Hernando Barragánia ei koskaan pyydetty Arduino tiimiin mukaan ja hänelle on jäänyt epäselväksi, miksi lähdekoodi erotettiin Wiring projektista Arduinoksi. Alla Kuva 1 ensimmäinen virallinen Arduino mikrokontrolleri. (Barragán, n.d.)

Kuva 1. Ensimmäinen virallinen Arduino



2.2 Atmel

Atmel Corporation on 1984 perustettu, yksi nopeimmin kasvaneista puolijohde teollisuuden yrityksistä Yhdysvalloissa. Atmel suunnittelee, kehittää ja valmistaa ohjelmitavia logiikkalaitteita, sovelluskohtaisia integroituja piirejä, sekä muisti- ja mikro-ohjainlaitteita. Atmel on haalinut mainetta ja kunniaa 1980- ja 1990-luvuilla kehitettyjen haihtumattomien muistipiirien parissa. Atmelin tunnettuja asiakkaita ovat olleet mm. Motorola, Nokia ja Ericsson. (Atmel Corporation, n.d).

2.2.1 STK500

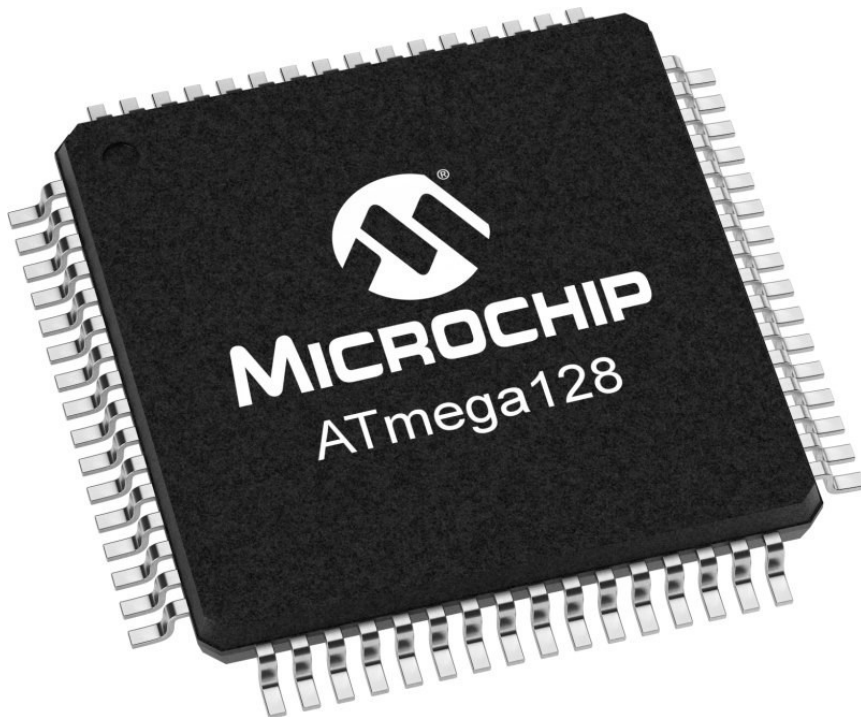
STK500 on mikro-ohjainten kehitysjärjestelmän aloituspakkaus, jonka avulla suunnittelijat voivat aloittaa AVR-koodin (Alf and Vegard's RISC processor) kehittämisen, prototyypit sekä

uusien mallien testaamisen nopeasti. STK-protokolla on liitäntä Atmel Studion integroidun kehitysympäristön kanssa koodin kirjoittamista ja virheenkorjausta varten. (STK500, n.d).

2.2.2 ATmega128

ATmega128 on Atmel AVR perheeseen kuuluva 8-bittinen pienitehoinen mikro-ohjain (Kuva 2). RISC arkkitehtuuriin perustuvassa ohjaimessa on 64 nastan liitäntä. 133 komennon suorittamisen yhdellä ohjelmakierroksella ja 32 x 8 bittinen yleiskäyttöinen työrekisteri tekevät tästä laitteesta loistavan valinnan moniin sovelluksiin. ATmega128 tärkeimmät ominaisuudet ovat: Käyttöjännite 4,5–5,5V, Ohjelmamuisti 128KB, Ram 4KB, EEPROM 4KB, 53 I/O pinniä, kellotaajuus 16MHz. (Adnanaqeel, Introduction to ATmega128, 2018)

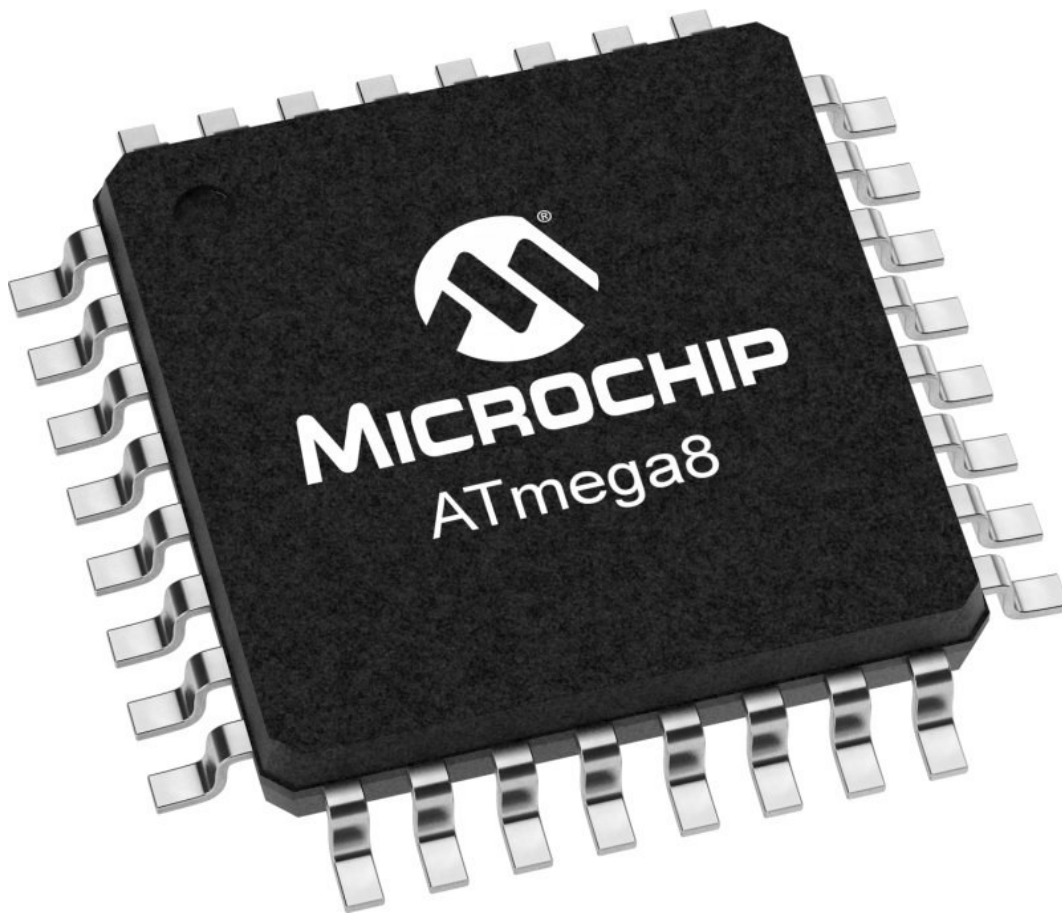
Kuva 2. ATmega128 mikrosiru (Microchip, ATmega128, n.d.-a)



2.2.3 ATmega8

ATmega8 on 28 pinninen, 8-bitin AVR mikrokontrolleri (Kuva 3), joka perustuu RISC arkkitehtuuriin. ATmega8 käytetään yleisimmin teollisuuden automaatioprojekteissa pienen kokonsa ja kustannustehokkuuden takia. ATmega128 tärkeimmät ominaisuudet ovat: Käyttöjännite 2,7–5,5V, Ohjelmamuisti 8KB, Ram 1KB, EEPROM 512B, 23 I/O pinniä, kellotaajuus 16MHz. (Adnanaqeel, Introduction to ATmega8, 2018)

Kuva 3. Atmega8 mikrosiru (Microchip, ATmega8, n.d.-b)



2.3 Toimintaperiaate

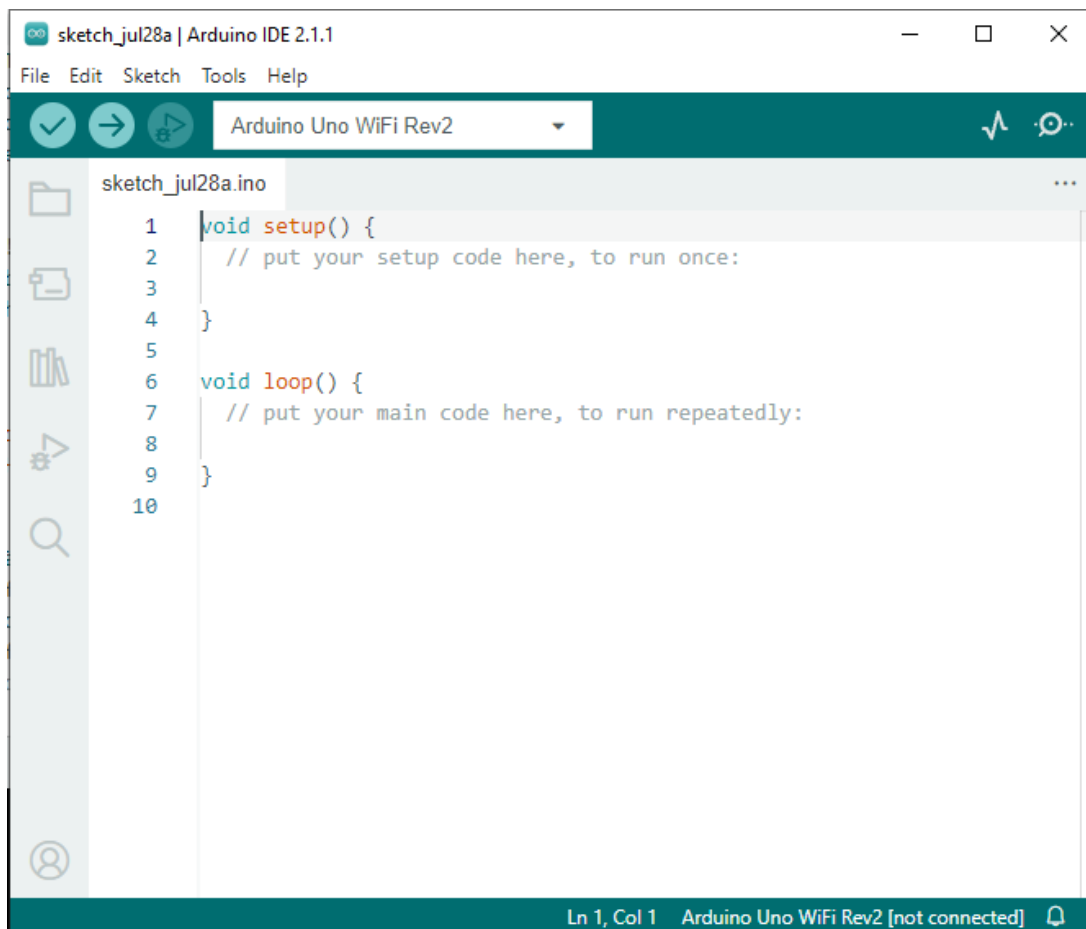
Arduino mikrokontrolleri on ATMEL AVR mikro-ohjainkorttiin perustuva piiri. Mikro-ohjaimet ovat integroituja piirejä, joihin voi tallentaa haluttuja toimintoja. Toiminnot ohjelmoidaan Arduino IDE (Integrated development environment) -ohjelmistossa, ohjelmointikoodi kielenä toimii C- ja C++ -ohjelmointikielien. Arduinossa toiminta tapahtuu tietoliikenne porteilla sekä tulo- ja lähtöporteilla, joihin voi kytkeä tarvittavia lisälaitteita kuten esimerkiksi antureita. Arduino kortteja on valmistettu moneen tarkoitukseen. Arduinoon on saatavilla laaja valikoima lisävarusteita, jotka laajentavat käyttömahdollisuuksia, näitä varusteita kutsutaan nimillä Shield tai Backpack. Laitteita ovat esimerkiksi GPS, kello, radioyhteys, kosketusnäyttö ja monia muita. (CircuitSchools Staff, 2023)

Arduinolla voidaan luoda yksittäisiä elementtejä, se voidaan yhdistää useampaan laitteeseen, sekä se voi olla vuorovaikutuksessa laitteiston, sekä ohjelmiston kanssa. Mahdollisuudet ovat valtavat, esimerkiksi sälekaihtimia ohjaavan moottorin ohjaus on mahdollista suorittaa Arduinolla hyödyntämällä Arduinoon kiinnitettyä valoanturia. Arduinolla on mahdollista luoda mitä tahansa itsenäisiä toimintoja. Käyttömahdollisuudet laajentuvat jatkuvasti Arduino yhteisön ansiosta. (CircuitSchools Staff, 2023)

2.4 Arduino IDE

Arduino IDE on ohjelmointiin tarkoitettu ympäristö, joka mahdollistaa Arduino mikrokontrollerin suorittaman koodin kirjoittamisen. Ohjelma tukee Windows, Linux ja Macintosh OSX -käyttöjärjestelmiä. Kirjoitushetkellä ohjelmiston uusimman versio on 2.1.1. Alla Kuva 4 Arduino IDE:n oletusikkuna. (Getting started, 2018)

Kuva 4. Arduino IDE:n oletusikkuna (Olli-Pekka Laukkanen, 2023)



Arduino IDE tukee ohjelmointikielinä C ja C++ kieliä. Ohjelmaa tai koodia kutsutaan nimellä Sketch. Arduino IDE:n tallentama sketch on tiedosto muotoa .ino (Javapoint, n.d).

Arduino IDE:n käytössä haasteena on virheellisen koodin korjaus. Arduino IDE ei kerro koodissa olevien virheiden sijaintia, vaan antaa yleispätevän ilmoituksen, kun koodia käännetään Arduino mikrokontrollerille. Virheiden korjaus vaatii käyttäjältä kokeilemisen kautta onnistumista.

Arduinon koodi koostuu kahdesta pääfunktioista. Setup funktio, joka suoritetaan, kerran kun laite käynnistetään. Setup funktiossa alustetaan tarvittavat kirjastot, pinnit ja muuttujat. Loop suoritetaan setupin jälkeen ja se kiertää kehää, kunnes laite sammutetaan. Loopin tehtävä on ohjata Arduinon piirilevyä ja siihen kytkettyjä lisälaitteita.

2.5 Kirjastot

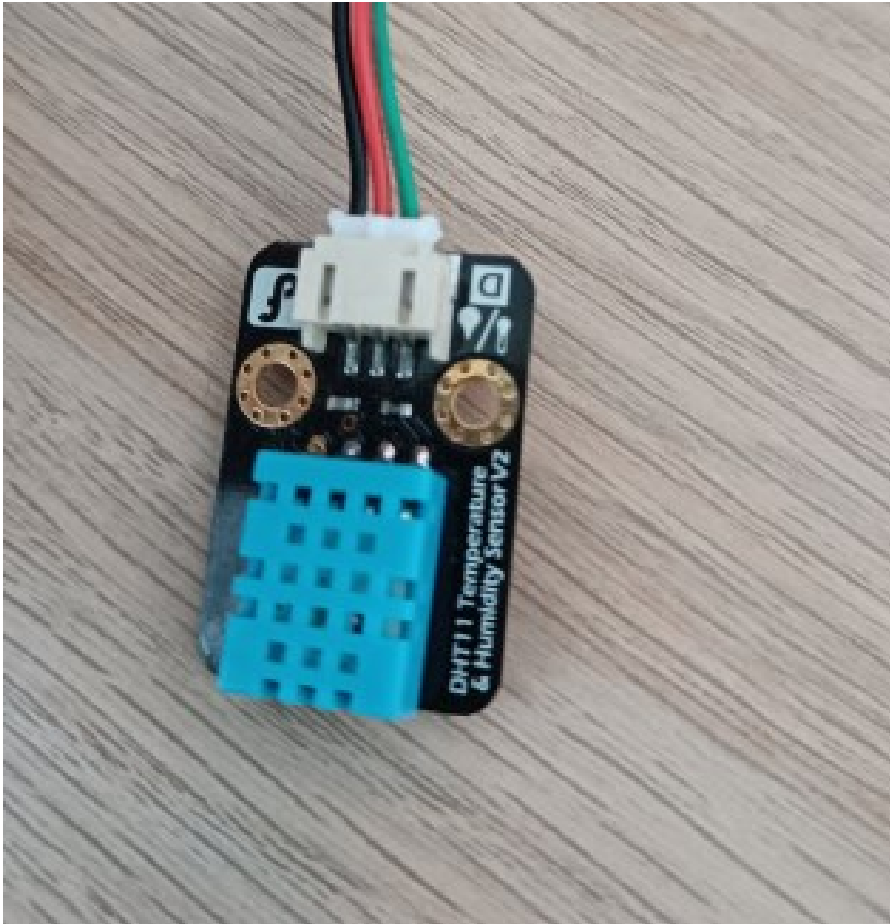
Kirjastot ovat tiedostoja, jotka mahdollistavat laajemman valikon toimintoja Arduino-laitteella, Kirjastot ovat kirjoitettu C++-kielellä. Kirjastoja on laaja valikoima valmiina Arduino IDE -ohjelmassa, mutta niitä on mahdollisuus luoda myös itse. Kirjaston käyttö Arduino-projektissa tapahtuu ”#include” komennolla. (Arduino, 2023)

Kirjaston tehtävä on kääntää määritetyt komennot koodissa Arduinolla tunnistettavaan muotoon esimerkiksi ”DHT11” joka kertoo anturin mallin tai DHT11 liittyvä ”TEMPTYPE”, joka määrittää käytettävän lämpötila asteikon.

2.6 DHT11 – Lämpötila- ja kosteusanturi

DHT11 (Digital Humidity & Temperature) on Kuva 5 esitetty edullinen digitaalinen kosteus- ja lämpötila-anturi. Toiminta perustuu kosteuden mittauksessa kapasitiiviseen kosteusanturiin ja lämpötilamittaukseen käytetään lämmöstä muuttuvaa vastusta eli termistoria. Anturi antaa arvot digitaalisena signaalina. Anturi toimii parhaimmillaan 2 sekunnin mittaussnopeudella. Anturin tarkkuus on hyvä 0-50 celsius asteen lämpötiloissa, jolloin mittaustoleranssi on +- 2 celsius astetta. Kosteusmittauksen tarkkuus on hyvä 20–80 % kosteudella 5 % tarkkuudella. (DIGITAL TEMPERATURE AND HUMIDITY SENSOR (DHT11), 2021)

Kuva 5. DHT11 Lämpötila- ja kosteusanturi (Olli-Pekka Laukkanen, 2023)



3 Ohjelmointikielät

3.1 C-ohjelmointikieli

C-Kieli on yleiskäyttöinen ohjelmointikieli, joka on helppo, joustava sekä erittäin yleinen ohjelmointikieli. C-kieli on strukturoitu ohjelmointi kieli, joka on laitteesta riippumaton ja laajasti käytetty kieli. Esimerkiksi Windows käyttöjärjestelmä on kirjoitettu suurimmilta osin C-kielillä. C-kieli on luotu vuonna 1972 Yhdysvalloissa. Dennis Ritchie loi C-kielen yhdistelemällä ALGOL, BCPL sekä B -kieliä, sekä lisäämällä muita ominaisuuksia kieleen. C-kieltä pidetään ohjelmoinnin, sekä koodauksen perustana. Kielen tuntemus on hyödyllistä kielen monikäyttöisyyden takia, jos tuntee C-kielen, on helppo vaihtaa ohjelmointikieltä

muiden kielten käyttäessä hyvin samankaltaista rakennetta, koska muut kielet ovat vahvasti C-johdannaisia. (Thompson, 2023)

C-kielen vahvuus perustuu kielen nopeuteen, joka perustuu siihen, että C-kielessä ei siivota koodia, joka ei tuota ylimääräistä laskenta aikaa tästä syystä C-kieli on nopeampi, kuin moni muu ohjelmointikieli. Ohjelmointikielet voi jakaa karkeasti kolmeen tasoon, matala taso, keskitaso ja korkeataso. Matalan tason ohjelmointikielet käsittelevät ohjelmia nopeammin, kun taas korkean tason ohjelmointikielet ovat käyttäjä- ja kehittäjäystävällisempiä. C-kieli on keskitaso ohjelmointikieli, jossa yhdistyvät matalan, sekä korkean tason kielten parhaat puolet. C-kieli on helppo ymmärtää, helppo käyttää, ja se käsittelee ohjelmia nopeammin kuin korkean tason ohjelmointikielet. (NIIT Editorial, 2021)

3.2 C++-ohjelmointikieli

C++ on yksi maailman suosituimmista ohjelmointikielistä. C++ on monialustainen kieli, jota on mahdollista käyttää tehokkaiden sovellusten kirjoittamiseen. C++ on kehitetty C-kielen lisäosaksi, mutta on saanut 4 suurta päivitystä vuosina 2011, 2014, 2017 ja 2020. C++-ohjelmointikieli toimii lähes samalla syntaksilla kuin C-ohjelmointikieli. (C++ Introduction, n.d).

C++-ohjelmointikieli sopii suuriin projekteihin. Monet projektit kuten kääntäjät, pilvitalennusjärjestelmät, tietokannat, pelinkehitys, graafiset suunnittelut ja monet muut ovat rakennettu C++-kielellä. Kieli sopii myös 3D-mallinnuksen projektien tekemiseen, nämä projektit vaativat valtavien tietomäärien tehokkaan hallinnan ja C++ on yksi parhaista valinnoista siihen. Myös monet sovellukset ja ohjelmistot, joita käytät tai käyt läpi jokapäiväisessä elämässäsi, on kehitetty C++-kielellä. Esimerkiksi Spotifyn, suosittu musiikin suoratoistosovelluksen, taustakoodi on kirjoitettu C++-kielellä. YouTube-videonkäsittelytoiminnon rakentamisessa käytetään C++ kieltä. Amazon-sovellus on myös kirjoitettu C++-kielellä. C++-kielen käyttö yleistyy vuosi vuodelta ja edellä mainittujen

sovellusten lisäksi samankaltaisten sovellusten kehitykseen on käytetty C++-kieltä. (Menon, 2023)

3.3 HTML-kieli

HTML (Hyper Text Markup Language) on verkkosivujen luomiseen tarkoitettu kieli, sitä ei yleisesti pidetä ohjelmointikielenä, koska sillä ei voi luoda dynaamisia toimintoja. HTML-lyhenne tulee sanoista "Hyper Text Markup Language". Html on aloittelija ystävällinen sekä, laajasti tuettu kieli. (Sarosa, 2023)

3.3.1 Rakenne

Jokainen HTML-dokumentti alkaa komennolla "`<!DOCTYPE html>`", joka kertoo verkkoselaimelle dokumentin tyyppin. Verkkosivu koostuu useammasta elementistä. HTML-elementtiin kuuluu kolme pääosaa, jotka ovat avaava tunniste, sisältö sekä päättävä tunniste. Avaavalla tunnisteella ilmoitetaan missä elementti alkaa ja aloitetaan komennolla "`<p>`". Sisältö on se osa, jonka käyttäjä näkee. Lopetustunniste on sama kuin avaava tunniste, joka päättää elementin komennolla "`</p>`". (Sarosa, 2023)

3.3.2 Erikoismerkit

HTML-kielessä erikoismerkit eivät toimi totuttuun tapaan vaan kirjoitetaan merkkiviittauksina, kuten esimerkiksi suomessa käytetty Ä kirjain kirjoitetaan html kielessä "`ä`". Erikoismerkkejä ovat mm. skandinaaviset kirjaimet, tarkkeelliset kirjaimet, valuuttojen merkit, matemaattiset merkit, latinalaisen merkistö lisäkirjaimet, symbolit, nuolet, erikoisvälimerkit, loogiset operaattorit. (Korpela, 2023)

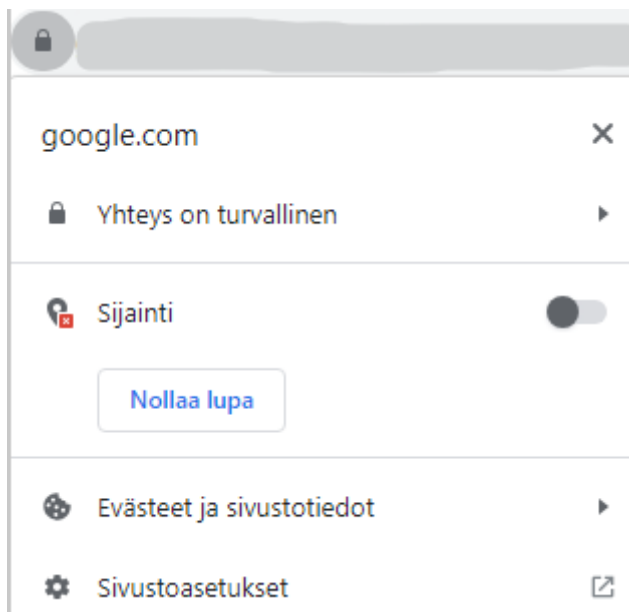
4 WiFi – Langaton verkko

WiFi (Wireless Fidelity) on IEEE 802.11 standardiin kuuluva langattoman verkon nimitys. WiFi-verkko toimii radioaalloilla, jonka taajuudet ovat joko 2,4 tai 5 gigahertsiä. WiFi-verkkoon lähetetty data muutetaan radioaalloiksi ja vastaanottava osa muuttaa radioaallot prosessorilla luettavaan muotoon. Verkon taajuus on huomattavasti muita taajuuksia suurempi, joka mahdollistaa suuremman datamäärän siirron. 2,4 gigahertsin kantama on noin 30 metriä, kun 5 gigahertsin kantama on maksimissaan noin 60 metriä. Tavallisesti reitittimet mahdollistavat verkon toimivuuden 30 metrin säteellä reitittimestä esteettömässä tilassa, esteitä ovat esimerkiksi seinät ja muut reitittimen ja laitteen välissä olevat esteet. (Brain & Homer, 2021)

5 SSL-sertifikaatti

SSL-sertifikaatti (Secure Socket Layer) on internetin liikenteen suojausmenetelmä standardi, jonka tarkoitus on luoda turvallinen yhteys kahden laitteen välille. SSL-yhteys pitää huolen, että data ja tieto, joka liikkuu internetissä selaimestasi web-palvelimeen, säilyy turvassa hakkereilta ja tahoilta, jotka mahdollisesti yrittävät vakoilla tai varastaa tiedot. SSL on laajalti levinnyt standardi ja sitä käytetään miljoonilla verkkosivuille arkaluontoisen materiaalin ja tiedon suojaamiseen. Yleisimpiä SSL-standardin käyttäjiä ovat verkkokaupat. SSL-standardi toimii myös sisäänkirjautumisen suojaamisessa. SSL-sertifikaatin toimiminen vaatii oikeaoppisen asennuksen. Sertifikaatin tulee olla yleisesti tunnetun ja hyväksytyin myöntäjän allekirjoittama, jolloin se hyväksytään sellaisenaan takeena siitä, että sivusto on turvallinen ja luotettava. Hyväksytyjä sertifikaatin antajia ovat esimerkiksi Firefox, Internet Explorer, Chrome ja Safari. Oikein asennuksen voi tarkistaa verkkosivun osoitekentän vasemmasta laidasta löytyvällä lukon kuvalla, joka on esitetty Kuva 6. Lukkoa klikkaamalla aukeaa varmenteen lisätiedot, kuten varmenteen myöntäjä ja voimassaolo aika. Kaikilla SSL-sertifikaateilla on erääntymispäivämäärä, jonka jälkeen varmenne ei ole enää voimassa. Varmenteen puuttuessa selain antaa yleensä selkeän varoituksen ja sivulle jatkaminen vaatii käyttäjältä aktiivisen hyväksymisen epäturvalliseen yhteyteen liittymiseen. Toinen keino salatun yhteyden tunnistamiseen on osoitteen alussa olevan ”https” lopun S-kirjain, joka ilmaisee suojauksen olemassaolon. (Aitoa, n.d).

Kuva 6 SSL-sertifikaatin tarkastus ikkuna (Olli-Pekka Laukkanen, 2023)



5.1 Extended Validation

Extended Validation varmenne on käytössä kaikenlaisilla tahoilla kuten valtiohallinnon organisaatiot, yhdistykset, yritykset ja säätiöt. Hakuprosessin kesto on noin 10 vuorokautta ja ne myönnetään, kun hakijan oikeus sivustoon on varmistettu. Extended Validation varmenteen myöntämisperusteet on määritetty vuonna 2007 ja siinä säädeltiin ne askeleet, jotka myöntäjäauktoriteetin tulee käydä läpi ennen myöntämistä. (Aitoa, n.d)

5.2 Organization Validation

Organization Validation varmenne on käytössä verkko-osoitteissa, jotka omaavat oikeuden hakemaansa verkko-osoitteeseen. Lisäksi organisaation oikeellisuus tutkitaan tietyllä tarkkuudella. OV-sertifikaatin saaneen organisaation tiedot näkyvät kappaleessa 5 mainitussa lukon kuvakkeen takaa aukeavassa tietovalikossa. (Aitoa, n.d)

5.3 Domain Validation

Domain Validation varmenne myönnetään, kun myöntäjäauktoriteetti on tarkistanut hakijan oikeuden kyseiseen verkko-osoitteeseen. Domain Validation varmenteeseen ei suoriteta minkäänlaista organisaation oikeellisuustarkistusta eikä kappaleessa 5 mainitun lukon takaa löydy organisaation tietoja. (Aitoa, n.d)

5.4 Wildcard-sertifikaatti

Normaalit varmenteet (Extended Validation, Organisation Validation, Domain Validation) suojaavat yhtä verkkosivustoa kerrallaan. Wildcard-sertifikaatilla voidaan suojata päädomain, joka sisältää monia alidomaineja, yleensä alidomaineja voi olla rajattomasti. (Aitoa, n.d)

6 Etäyhteys

6.1 VPN

VPN (Virtual Private Network) on kehitetty helpottamaan etäyhteyksien luomista. Nykypäivänä organisaatiot toimivat hajautetussa ympäristössä, jolloin etäyhteyksien käyttö on välttämätöntä. Usein organisaation edustajilla on välttämätöntä olla mahdollisuus päästä organisaation verkkoon liikkuvan työn takia. Sisäverkon käyttö ei kuitenkaan ole turvallista sellaisenaan, sillä se on suuri tietoturvariski. Julkisen verkon kautta voidaan lähettää tiedot ns. tunnelin läpi. Sitä kutsutaan VPN-yhteydeksi. Etäyhteydet ovat välttämättömyys yritysmaailmassa.

VPN yhteydessä tiedot kulkevat tunneliverkossa, jolloin tietoja on mahdoton kalastella. VPN yhteys toimii kahden tai useamman verkon välillä julkisen verkon kautta. Turvallisuuden näkökulmasta tiedon on pysyttävä muuttumattomana sekä luottamuksellisuuden tulee säilyä. Käyttäjien ja heidän käyttöoikeuksien hallinnalla kasvatetaan turvallisuutta.

VPN-yhteyden mahdollisia käyttökohteita ovat esimerkiksi työpaikan sisäisen verkon käyttö etätöissä tai matkustaessa. Yksityishenkilöllä kotiverkon käyttö matkustaessa.

Selaustoiminnon piilottaminen julkisessa verkossa. Julkista WiFi-verkkoa käyttäessä selaustoiminta näkyy muille, mikäli vierailee verkkosivuilla, joita ei ole suojattu SSL-varmenteella.

Julkisessa verkossa data kulkee monien palveluntarjoajien verkkojen ja laitteiden läpi, tämä ei kuitenkaan edellytä palveluntarjoajilta minkäänlaista turvallisuutta. Virtuaalinen tunneli takaa turvallisuuden näiden verkkojen ja laitteiden läpi. Tällaisissa tilanteissa palveluntarjoaja toimii vain IP-liikenteen kuljettajana. (Raatikainen, 2016, ss. 16-24)

6.1.1 Host-to-site VPN

Etäyhteys VPN yhdistää yksityisen henkilön laitteen ja organisaation sisäisen verkon.

Esimerkiksi etätöitä tekeväälle työntekijälle mahdollistetaan organisaation sisäisen verkon käytön turvallisesti julkisen verkon yli. Käyttäjä tarvitsee tunnelin luomiseen VPN-ohjelmiston. Organisaation sisäverkon VPN-yhdyskäytävä pitää huolen käyttäjän tunnistamisesta, sekä luo suojatun tunneliyhteyden. Kun tunneliyhteys on luotu, käyttäjä voi siirtää dataa oman laitteen ja organisaation verkon välillä turvallisesti. Etäyhteys tarjoaa turvallisen ja salatun keinon kommunikoida kahden osapuolen välillä, jotka ovat yhteydessä internetiin. (Raatikainen, 2016, s. 16)

6.1.2 Site-to-site

Site-to-site VPN yhdistää lähiverkon julkisen verkon yli toiseen lähiverkkoon esimerkiksi yhdistäessä yrityksen kahden toimipisteen verkot yhdeksi kokonaisuudeksi. Tämä tarkoittaa sitä, että VPN-tunneli on luotu kahden VPN-yhdyskäytävän väliin. Toimipisteiden VPN-yhdyskäytävät kommunikoivat keskenään ja muodostaa suojatun tunnelin. VPN-yhdyskäytävä on vastuussa tiedon salaamisesta, käyttäjien ja verkon tunnistamisesta sekä tiedon eheydestä. (Raatikainen, 2016, s. 17)

6.1.3 SSL VPN

Nykyään yleisimmin käytetty VPN-yhteys on SSL-pohjainen. SSLVPN (Secure Socket Layer Virtual Private Network) käyttää salaukseen SSL/TSL-protokollaa. SSL VPN voi käyttää joko selaimen kautta tai erillisellä ohjelmistolla kuten Microsoftin SSTP. SSL VPN käyttää samaa verkkoprotokollaa kuin HTTPS salaus. (Raatikainen, 2016, s. 23)

6.1.4 VPN:n hyödyt

VPN on halpa vaihtoehto verrattuna perinteisiin kiinteisiin yhteyksiin kahden pisteen välillä. VPN poistaa pitkien matkojen kiinteiden yhteyksien tarpeen käyttämällä julkista verkkoa turvallisesti. VPN käyttää salausta, todentamista ja valtuuttamista taatakseen turvallisen, eheän ja luottamuksellisen tiedonsiirron. Internet pohjainen VPN mahdollistaa yrityksen sisäverkon kehityksen ja kasvamisen, kun liiketoiminta tarvitsee muutosta ilman suuria laitehankintoja. VPN tarjoaa helpon yhteyden kahden toimipisteen välille koska internet on laajalti levinnyt yhteys. VPN tunnelin etäisyys on rajaton, mikäli molemmilla osapuolilla on yhteys internettiin. Etätyöskentely organisaation sisäisessä verkossa mahdollistuu. (Raatikainen, 2016, ss. 23-24)

6.2 TeamViewer

TeamViewer on langattoman yhteyden mahdollistava ja tietoteknisen etätuen antamiseen suunniteltu sovellus. Sovellus tarjoaa langattoman turvallisen yhteyden kahden tai useamman laitteen välille. Sovelluksen käyttö on ilmaista yksityishenkilölle, mutta kaupalliseen käyttöön sovellukseen myydään erillistä lisenssiä, jonka hinta muotoutuu käyttötarkoituksen mukaan. TeamViewer mahdollistaa pääsyn laitteelta toiselle etäyhteydellä, mutta tämän lisäksi kyseistä laitetta on mahdollista ohjata toiselta laitteelta. Tiedostojen siirto onnistuu myös TeamViewerin kautta. TeamViewer tukee monia alustoja kuten Windows, MacOS, Linux, Android ja IOS pohjaiset laitteet. (Gratas, 2022)

7 Käytännön toteutus

7.1 Arduino WiFi Rev2

Projektissa käytetty Arduino WiFi Rev2 on mikrokontrollerina vastaava kuin kolmannen sukupolven Arduino Uno R3, mutta siinä on sisäänrakennettuna langattoman yhteyden mahdollistava moduuli, joka sisältää WiFi, Bluetooth ja Bluetooth low energy yhteydet.

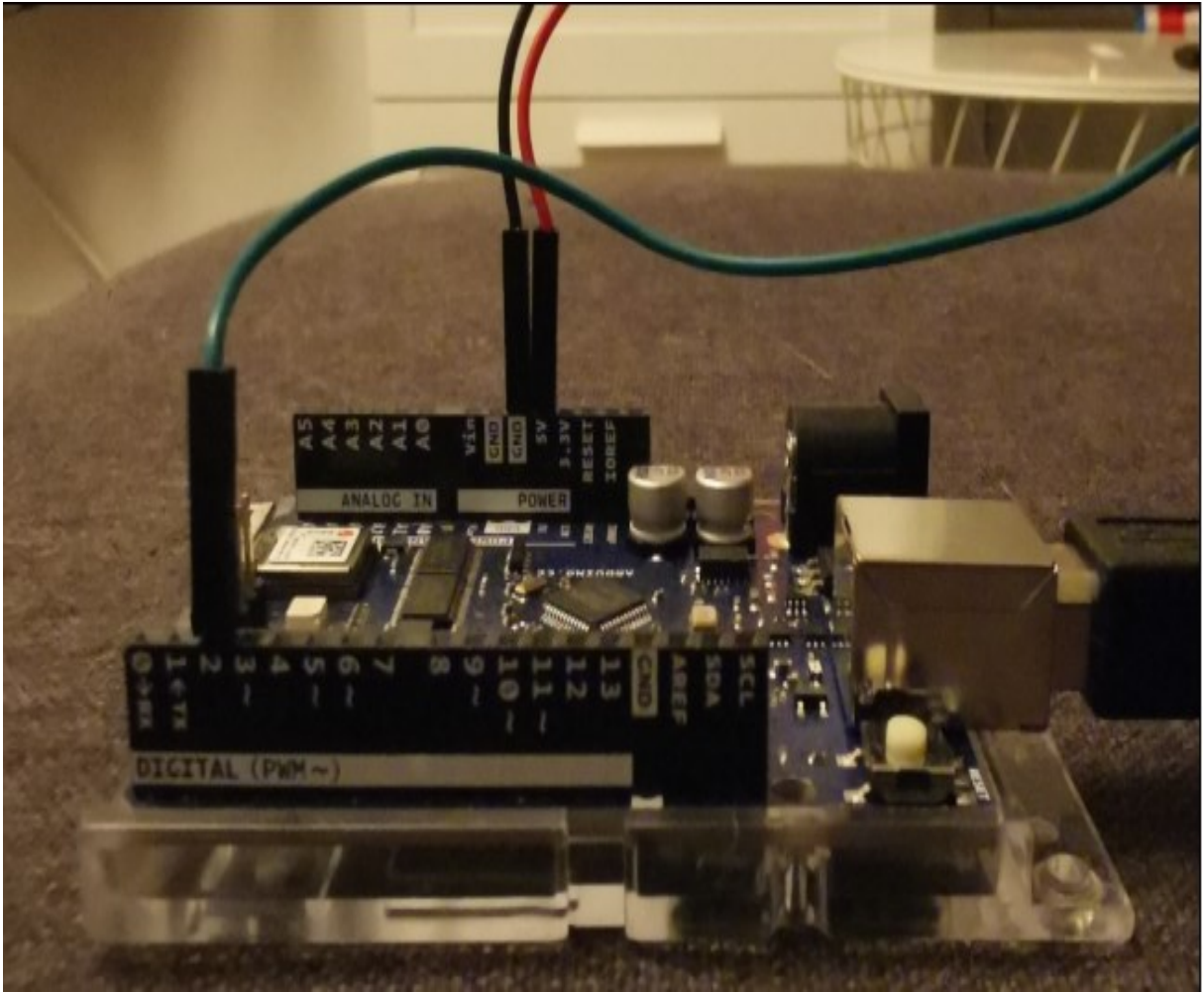
Arduino WiFi Rev2 tärkeimmät ominaisuudet ovat seuraavat:

- Atmega4809-Mikrokontrolleri
- 5V Käyttöjännite
- Suositeltu syöttöjännite: 6-20V
- 14 digitaalista I/O-Pinniä, joista 6 on PWM-ohjattuja lähtöjä
- 6 analogista tuloa
- WiFi ja Bluetooth Nina W102 ublox moduuli
- 48kb flash-muistia
- 6kb SRAM-muistia
- 256 bittiä EEPROM-muistia
- 16 MHz kellotaajuus (Arduino, UNO WiFi Rev2, n.d.)

7.2 Kytkennät

Työ toteutettiin Hämeen ammattikorkeakoulun tarjoamalla Arduino Uno WiFi Rev2 kortilla, sekä DHT11 lämpötila/kosteusanturilla. DHT11 anturi toimii testausvälineenä verkkoserverin välittämän tiedon testauksessa. Testausvaiheessa käytettiin kahta varavirtalähdettä sekä Usb-johtoa. Arduino korttiin kytkettävä DHT11 lämpötila- ja kosteusanturi kytkettiin Kuva 7 esitetyllä tavalla. Hyppykaapelilla 5V jännitepaikalle, maadoitukseen sekä digitaalipaikkaan kaksi. Virransyöttö tapahtui Arduinossa olevaan Usb-A liittimeen tietokoneen ja varavirtalähteen kautta mikrokontrollerin sijainnista riippuen.

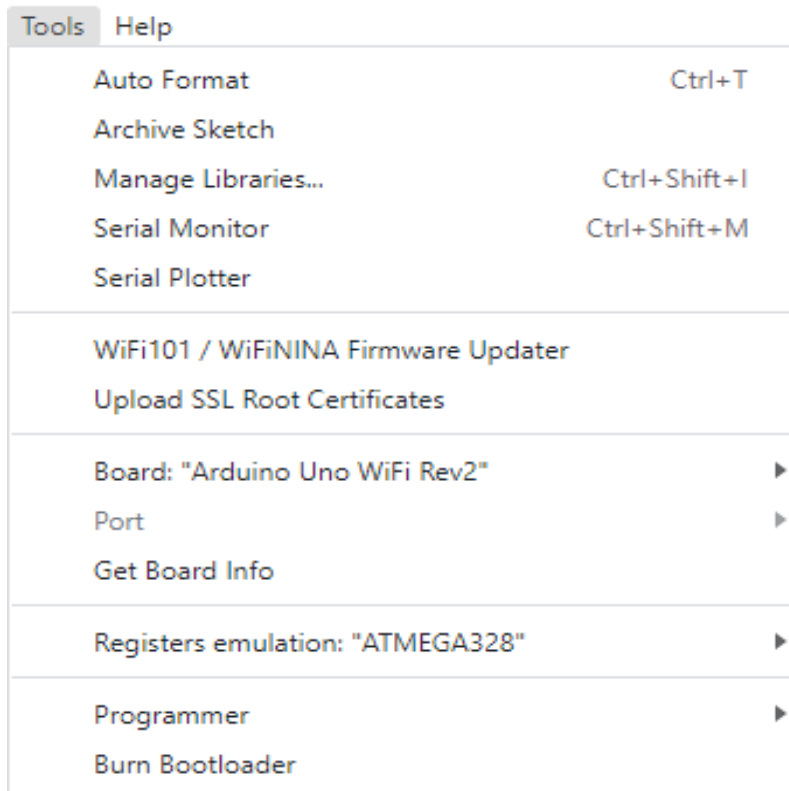
Kuva 7. DHT 11 anturin kytkentä mikrokontrolleriin (Olli-Pekka Laukkanen, 2023)



7.3 Ohjelmointi

Arduino Ide ohjelmiston käyttö aloitettiin valitsemalla tietokoneeseen kytketty mikrokontrollerin malli yläreunan "Tools" valikosta (alla Kuva 8), josta löytyy kytketty mikrokontrolleri Arduino Uno WiFi Rev2. Kirjastojen asennus tapahtuu samasta valikosta kohdasta "Manage Libraries", asennetut kirjastot ovat "WiFiNINA" ja "DHT sensor library". Viimeisenä luodaan itse ohjelmakoodi.

Kuva 8. Arduino Ide Tools alavetovalikko



7.3.1 WiFi-yhteys

Langaton WiFi-yhteys luotiin Arduinoon kirjoittamalla ohjelmakoodi, joka yhdistää Arduinon suoraan reitittimeen. Ohjelmassa määritetään ennalta haluttu WiFi-yhteys SSID:n avulla. Koodiin määritettiin WiFi-yhteyden salasana. Arduino Ide ohjelmistossa WiFi-yhteyden luomiseen tarvittava kirjasto nimeltä "Wifi.h" asennettiin kirjastovalikosta. Mobilisoitavuuden takia reitittimenä käytettiin Lenovon TB-X505L android tablettia.

7.3.2 Turvallisuus

Arduinon kirjastoon WiFi Nina on mahdollista lisätä SSL-Varmenne, joka estää verkkoyhteyden tietojen vakoilun. SSL-varmenne asennetaan ohjelmistoon Arduino IDE alustan kautta "Tools" valikosta "Firmware updater" tämän jälkeen määritetään varmenne tarjoajan osoite esimerkiksi "Google.com". Suojausta yritettiin toteuttaa WiFi Nina kirjaston mahdollistamalla SSL-varmenteella, joka luo yhteydestä turvallisen, mutta Arduino IDE

(versio 2.x.x) puutteellisen tuen takia sertifikaatin lataus ei tunnista arduino Uno Wifi Rev 2 piirilevyä. Sertifikaatin lataus sen sijaan onnistui vanhemmalla Arduino IDE ohjelmistolla (versio 1.x.x). Tietoturvan näkökulmasta onnistuttiin luomaan toinenkin suojaus. WiFi-verkko segmentoitiin kahteen eri WiFi-yhteyteen reitittimessä, joista Arduino on yhteydessä ainoana laitteena toiseen yhteyksistä, tämä estää vakoilun WiFi verkossa oleviin muihin laitteisiin. Kolmantena turvallisuutta edistävänä asiana projektissa käytettiin VPN-yhteyttä kiinteään WiFi-yhteyteen yhdistettynä, jonka suojausprotokollat ovat haastavia murtaa. Mobilisoitavana laitteena turvallinen etäyhteys luotiin TeamViewer sovelluksen kautta.

7.3.3 Ohjelmakoodi

Arduino ohjelmointi tehtiin Arduino IDE ohjelmointiympäristössä, jotta laite toimii toivotulla tavalla. Arduino IDE mahdollisti koodin kirjoittamisen ja sen ajamisen Arduino mikrokontrollerille.

Arduinon koodi toimi seuraavasti. Ennen setup funktiota koodiin sisällytettiin Arduinon käyttämät kirjastot, joita olivat tässä projektissa DHT11 lämpötila ja kosteus anturin kirjasto "DHT.h", Synkronisen sarjaväylän kirjasto "SPI.h" (Serial Peripheral Interface) ja WiFi yhteyden mahdollistava "WiFi.h". Tämän jälkeen annetaan muuttujat verkon tunnus "ssid" ja verkon salasana "pass" sekä määritetään anturin DHT11 tyyppi, digitaalinen pinni sekä käytettävä lämpötila-asteikko.

Setup funktio avaa sarjamuotoisen yhteyden 9600 baudin tiedonsiirtonopeudella ja käynnistää anturin arvojen lukemisen. Tämän jälkeen luodaan yhteys reitittimeen edellä mainittujen tietojen mukaisesti. Tämän jälkeen yhdistetään laite verkkoon ja annetaan yhdistämiselle 10 sekuntia aikaa yhdistää delay komennolla, kun yhteys on muodostettu, käynnistetään serveri.

Verkkoserverin koodi toteutettiin komennolla "WiFiServer server(80);" joka sijaitsee koodin määrittämisessä ennen setup komentoa, numero 80 viittaa käytettävään porttiin. Serverin

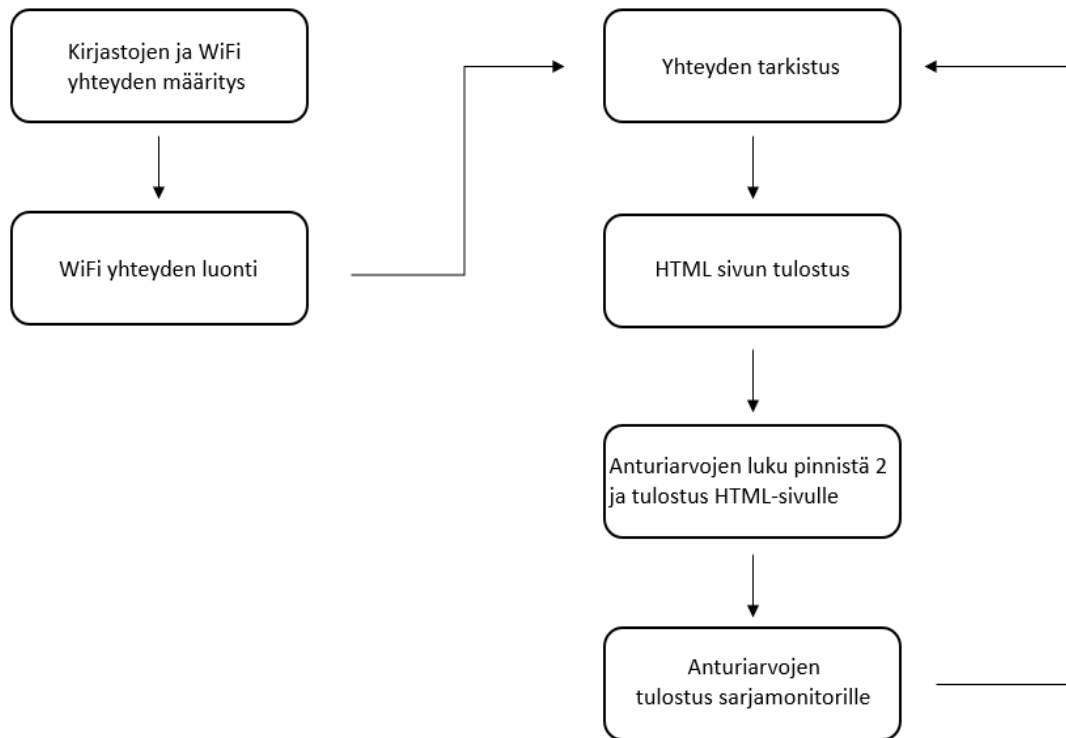
käynnistykseen käytetty komento on "server.begin();" , joka käynnistää serverin. Verkkoserverin sisältö luotiin Arduinossa "Client.println(""): " komennolla, jonka tuloksena syntyy HTML-sivun. Serveri tarkastaa paikalla olevat asiakkaat komennolla "WiFiClient client = server.available();" jolloin tietojen tulostus tapahtuu, muissa tapauksissa serveri on ns. lepotilassa. Verkkosivun sisältö luotiin soveltamalla HTML-kieltä "Client" komentojen sisällä kuten "client.println("<!DOCTYPE HTML>");".

Toiminnallisuudet verkkosivustolle on tehty luomalla verkkosivulle painikkeita, jotka liitetään valittuihin Arduinon pinneihin. Painiketta painaessa verkkosivu päivittyy ja uusi UI (User Interface) latautuu ruutuun, jolloin painikkeiden toiminnallisuudet muuttuvat. Tällä menetelmällä Arduino toimii ohjauksessa valmiina pohjana haluttujen laitteiden ohjauksessa.

Loop funktiossa tarkistetaan, onko yhteys muodostettu verkkoon, luetaan anturin arvot digitaalipinnistä 2 ja tulostetaan HTML sivu, johon tulostetaan anturin antamat arvot koodissa määritetyllä tavalla. Lämpötilan asteikoksi on määritetty celsius asteet ja kosteus tulostuu prosentteina. Saman aikaisesti yhteyden muodostamista ja anturin antamia arvoja voi seurata Arduino IDE:n sarjamonitorissa.

Viimeisenä osana koodi tulostaa sarjamonitoriin verkon tunnuksen, IP osoitteen ja yhteyden vahvuuden, joka tulostetaan kerran yhteyden muodostamisen jälkeen. Koodin toiminta esitetty diagrammina alla Kuva 9.

Kuva 9. Ohjelman toimintadiagrammi (Olli-Pekka Laukkanen, 2023)



7.4 Etäohjaus

Tämän projektin tärkeimpänä ominaisuutena on etäohjaus. Etäohjaus mahdollistaa mikrokontrollerin käytön ja seurannan sijainnista riippumatta. Kiinteää WiFi-verkkoa käyttäessä reitittimeen lisätään VPN-yhteys, jolloin halutut laitteet voidaan yhdistää reitittimeen VPN-tunnelin avulla julkisen verkon kautta ja tätä kautta päästään seuraamaan ja käyttämään Arduinoa. Mobiiliverkkoa ja mobilisoitavaa WiFi-tukiasemaa käyttäessä VPN yhteys ei onnistu, koska laitteiden välillä tulee olla VPN-serveri. Yhteyden muodostukseen otettiin käyttöön TeamViewer ohjelmisto, joka mahdollistaa mobilisoitavan WiFi-tukiaseman (tässä projektissa Tabletti) etäkäytön. Laitteiden välille syntyy langaton yhteys julkisenverkon kautta ja ohjelmisto mahdollistaa tabletin käytön etänä toisella laitteella, jolloin päästään käsiksi Arduinon toiminnallisuuksiin. TeamViewer siis välittää ohjattavan laitteen näyttönäkymää ohjaavalle laitteelle ja kyseistä laitetta pääsee fyysisesti ohjaamaan toisen laitteen kautta.

7.4.1 VPN

VPN yhteyttä käytetään mikrokontrollerin etäohjaukseen kiinteää WiFi-verkkoa käyttäessä. Projektissa käytetty reititin oli TP-Link merkkinen 4G reititin. Reitittimen konfigurointi aloitettiin reititinkohtaisten ohjeiden mukaisesti. Reitittimen käynnistämisen jälkeen avattiin asetukset verkkoselaimessa osoitteessa "http://192.168.1.1". Ensimmäisellä kerralla reititin pyytää luomaan salasanan. Seuraavassa vaiheessa tullut Quick Setup Wizard ajettiin loppuun täyttämällä kysytyt kohdat. Eri laitteistoille löytyy laite kohtaiset ohjeet, joita seuraamalla konfigurointi onnistuu. Quick setupin valmistuttua kokeiltiin toimiko yhteys verkkoon. Onnistuneen yhteyden muodostuksen jälkeen luotiin Dynaaminen nimipalvelu. Reitittimen asetuksissa yläreunassa valikko "Advanced", Sieltä valittiin No-IP ja klikataan "Go To Register". No-IP Käyttäjän luomisen jälkeen avattiin "Dashboard", jossa on mahdollista luoda Hostname. Hostname:n luomisessa on tärkeää varmistaa, että valitsemasi Hostname on hyväksytty. Seuraavaksi kirjauduttiin reitittimen asetuksissa No-IP tunnuksilla sisään aiemmin mainitussa "Advanced" valikossa. VPN yhteyden luonti vaatii staattisen IP osoitteen. Tämä tulee varmistaa operaattorilta. Mikäli IP osoite on dynaaminen, ilmaisena palveluna löytyy operaattorin puolesta "Dynaaminen staattinen IP". Verkossa on monia palveluita, joilla varmistetaan, onko käytössä oleva IP-osoite dynaaminen vai staattinen esimerkiksi "WhatsMyIp.com". Mikäli IP-osoite on sama verkkosivulla ja reitittimen asetuksissa on osoite staattinen. Seuraavaksi valittiin reitittimen asetuksissa vasemmasta laidasta VPN valikko. VPN palvelun tarjoajana käytettiin OpenVPN:n tarjoamaa palvelua. VPN-yhteyden käyttöönotto aloitettiin "Advanced" valikossa VPN server, löytyy OpenVPN ja valitaan "Enable VPN Server". Seuraavaksi valittiin UDP protokolla, vaihtoehtoja oli UDP (User Datagram Protocol) ja TCP (Transmission Control Protocol). Lisätään palvelulle portti, jonka tulee olla 1024 ja 65535 välillä. VPN Subnet/Netmask ruutuihin laitettiin 10.8.0.0 ja 255.255.255.0. Client access kohtaan valittiin "Internet and Home Network" ja painettiin "Save" eli tallenna. Tallentamisen jälkeen generoidaan sertifikaatti aiemmin täytetyn kohdan alapuolella. Sertifikaatin generointivalikon alapuolelta tallennetaan OpenVPN konfigurointi tiedosto, jota tarvitaan yhteyksien luomiseen eri laitteilta reitittimeen.

Halutuille laitteille asennetaan OpenVPN ohjelma ja lisätään aiemmin luotu konfigurointi tiedosto ohjelmaan. Tästä eteenpäin PPTP VPN yhteyden käyttö on Plug and play tyylinen, joka mahdollistaa helpon yhteyden luomisen.

7.4.2 TeamViewer

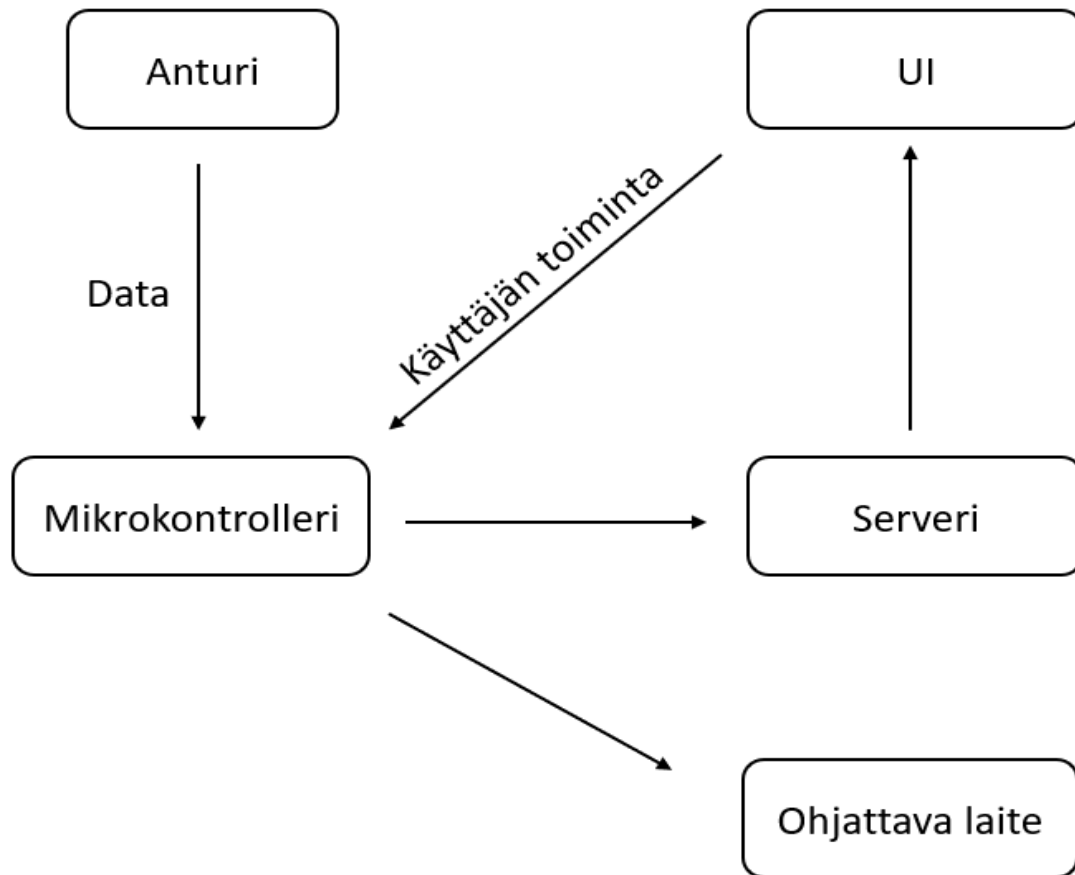
TeamViewerin käyttö toimii etäyhteyden luomisessa moitteettomasti mobiililaitteiden välillä. Käyttöönotto aloitettiin luomalla käyttäjä TeamViewerin verkkosivuilla.

Käyttäjätunnus ei ole pakollinen, mutta mahdollistaa ”osoitekirjan” ylläpitämisen, joka helpottaa yhteyden luomista tulevaisuudessa. WiFi-yhteyttä jakavalle mobiililaitteelle asennetaan TeamViewer Quick Support sovellus, josta löytyy yhteyden luomiseen vaadittava ID numerokoodi. Käyttölaitteelle, jolla Arduinon ylläpitämää serveriä halutaan käyttää, asennetaan TeamViewer Remote Control sovellus. Sovelluksen asentamisen jälkeen kirjaututaan sisään tämän kappaleen alussa mainitulla käyttäjätunnuksella, mikäli se on luotu. Ilman käyttäjätunnusta käytetään kolmannen osapuolen käyttäjätiliä, jotka ovat Google, Microsoft tai Apple käyttäjätunnukset. Sisään kirjautumisen jälkeen avautuu yhdistä ruutu, joka kysyy ”Kumppanin tunnusta”, tämä tunnus löytyy WiFi-yhteyttä jakavan laitteen TeamViewer Quick Support sovelluksesta. Etäohjauksen toimiminen vaatii myös lisäosan, jonka asennuttamista Quick Support ehdottaa sovelluksen käynnistyessä. Lisäosa mahdollistaa laitteen etäkäytön edellä mainitulla ID:llä. Suojauksena TeamViewerin yhteydessä toimii RSA, joka on julkisen avaimen salausalgoritmi, sekä 256 bittinen AES (Advanced Encryption Standard).

7.5 Toiminnallisuus

Serveri toimii alla kuvassa 10 esitetyn lohkoketjun mukaisesti. Anturi mittaa dataa, joka lähetetään mikrokontrollerille. Mikrokontrolleri lähettää tiedon serverille, joka näkyy käyttäjälle verkkosivustolla. Serveri lähettää tiedon paikalla olevasta asiakkaasta sekä asiakkaan toiminnan esimerkiksi sivustolla olevan painikkeen käytön takaisin mikrokontrollerille. Painikkeiden käyttö vaikuttaa muuttujien tilaan mikrokontrollerilla, joka ohjaa mahdollisesti liitetyjä laitteita Arduinossa. Tässä projektissa ohjattava laite oli Arduinon sisäänrakennettu Led-valo.

Kuva 10 Serverin toiminta (Olli-Pekka Laukkanen, 2023).



Sivujen toiminnallisuudet voidaan liittää minkä tahansa laitteen ohjaukseen käyttämällä sille annettua pinni paikkaa. Painiketta painaessa pinni paikka aktivoituu tai deaktivoituu, tämä vaikuttaa siihen, mikä sivu on auki ja aktiivinen sivu vaikuttaa siihen mitä painikkeesta tapahtuu. Alla kuvassa 11 toiminnallinen painike on aktivoimatta, kun taas Kuva 12 toiminnallinen painike on aktivoitu. Toiminnallinen painike soveltuu myös pienellä

muutoksella koodissa eri huoneiden mittausarvojen tarkkailuun ja muiden huonekohtaisten toiminnallisuuden lisäämiseen.

Kuva 11 Toiminnallinen painike aktivoimatta (Olli-Pekka Laukkanen, 2023).



Kuva 12 Toiminnallinen painike aktivoitu (Olli-Pekka Laukkanen, 2023).



7.6 Testaus

Arduino projektia kokeiltiin WiFi kantaman alueella tietokoneeseen kytkettynä, jolloin serialmonitoriin tulostetut arvot täsmäsivät erilliseen digitaaliseen mittariin verrattuihin tuloksiin 0,7 celsiusasteen erolla ja 1 kosteusprosentin erolla. Ulkoiseen virtalähteeseen kytkettynä arvot täsmäsivät tietokoneeseen kytkettynä saatuihin arvoihin. Serverin tulostamaa dataa tarkkailtiin Arduino IDE ohjelmiston Serial Monitorissa, joka todisti Arduino serverin lepotilan, kun serverille ei ollut yhteydessä yhtään asiakasta. Ulkoisena virtalähteenä toimi Energizerin 20000 mAh/74Wh virtalähde, jonka ulostulo jännite oli 5V ja virta 2,1A. Ulkoisen virtalähteeseen kytkettynä Arduinon suorittama mittaustaika oli seitsemän vuorokautta täydestä virrasta Arduinon sammumiseen asti. VPN-yhteyden luominen oli aluksi haastavaa, mutta onnistui toisenlaista reititintä käyttäessä moitteettomasti. Mobiililaitteiden välille ei ole mahdollista luoda VPN-yhteyttä ilman erillistä VPN-serveriä laitteiden välissä. Ongelma ratkesi käyttämällä etäohjaukseen soveltuvaa TeamViewer ohjelmistoa, joka mahdollistaa mobilisoidun laitteen käytön toisella laitteella.

8 Yhteenveto

Opinnäytetyön tavoitteena oli luoda verkkoserveri, jota ylläpidetään Arduino mikrokontrollerilla. Työn tuloksena syntynyt verkkoserveri toimii mobilisoidavana mittausyksikkönä, jossa käytetään DHT11 lämpötila- ja kosteusanturia olosuhteiden mittaukseen ja seurantaan. Mittausyksikkö kulkee ajoneuvossa, jossa on koirille rakennettu matkustus/oleskelu tila. Tällaisenaan laitteisto ei kuitenkaan sovellu ajoneuvossa säilytettäväksi vaan vaatii koteloinnin tai muun fyysisen suojauksen. Työn tekeminen aloitettiin tutustumalla Arduinon historiaan, ohjelmoinnin soveltamiseen projektissa sekä koodauskieliin ja luotiin näistä teoreettinen pohja projektille. Lämpötilan tarkkailun lisäksi projekti laajentui Arduinon etäohjaukseen soveltuvaan ratkaisuun, jolla voidaan ohjata kaikkia laitteita, jotka ovat kytkettävissä Arduinon. Etäohjattavuus VPN-yhteyden, sekä TeamViewerin kautta mahdollistaa vielä laajemman käytön laitteistolle. Laitteistona voi olla esimerkiksi taloautomaattioratkaisut kuten valo-ohjaus, ilmanvaihto ohjaus ja monet muut. Lopputuloksena syntynyt laitteisto siirtyy käyttöön ajoneuvon olosuhteiden tarkkailuun suunnitellusti.

Lähdeluettelo

Adnanaqeel. (18.9.2018-a). *Introduction to ATmega128*.

<https://www.theengineeringprojects.com/2018/09/introduction-to-atmega128.html>

Adnanaqeel. (12.9.2018-b). *Introduction to ATmega8*.

<https://www.theengineeringprojects.com/2018/09/introduction-to-atmega8.html>

Arduino. (21.7.2023). Arduino Libraries.

<https://docs.arduino.cc/hacking/software/Libraries>

Arduino-mikrokontrollerin ja koeytkentälevyn perusteet. (n.d.).

<https://www.kasityokoulurobotti.fi/2017/10/arduino-ohjelmointi-ympariston-perusteet-ja-asennus/>

Armenta, A. (8.9.2022). *Introduction to Arduino: History, Hardware, and Software*.

<https://control.com/technical-articles/introduction-to-arduino-history-hardware-and-software/>

Atmel Corporation. (n.d.). Atmel Corporation - Company Profile, Information, Business Description, History, Background Information on Atmel Corporation:

<https://www.referenceforbusiness.com/history2/2/Atmel-Corporation.html>

Barragán, H. (n.d.). *The Untold History of Arduino*.

<https://arduinhistory.github.io/>

Brain, M. & Homer, T. (17.8.2021). *How WiFi Works*.

<https://computer.howstuffworks.com/wireless-network.htm>

C++ Introduction. (n.d.).

https://www.w3schools.com/cpp/cpp_intro.asp

CircuitSchools Staff. (9.5.2023). *What is Arduino, how it works and what you can do with arduino*.

<https://www.circuitschools.com/what-is-arduino-how-it-works-and-what-you-can-do-with-arduino/>

DIGITAL TEMPERATURE AND HUMIDITY SENSOR (DHT11). (2021).

http://robomaa.fi/index.php?route=product/product&product_id=1675

Getting started. (5.2.2018).

<https://www.arduino.cc/en/Guide/Introduction>

Gratas, B. (7.2022). *What is TeamViewer? Remote Work Made Easy.*

<https://blog.invgate.com/what-is-teamviewer>

Javapoint. (n.d.). *Arduino Ide.*

<https://www.javatpoint.com/arduino-ide>

Korpela, J. K. (4.1.2023). *Entiteetit HTML:ssä.*

<https://jokorpela.fi/merkit/entiteetit.html>

Menon, K. (8.2.2023). *Top 10 Reasons To Learn C++ Language In 2023.*

<https://www.simplilearn.com/tutorials/cpp-tutorial/learn-cpp>

Microchip. (n.d.-a). *ATmega128.* [kuva]

<https://www.microchip.com/en-us/product/atmega128>

Microchip. (n.d.-b). *ATmega8.* [kuva]

<https://www.microchip.com/en-us/product/atmega8>

NIIT Editorial (17.02.2021). *7 Reasons Why C is the Best Programming Language for Beginners.*

<https://www.niit.com/india/knowledge-centre/reasons-to-learn-C-programming-language>

Raatikainen, S. (2016). *OpenVPN-Etäyhteys.* [opinnäytetyö, Vaasan Ammattikorkeakoulu].

<https://urn.fi/URN:NBN:fi:amk-2016060812410>

Sarosa, A. (2.5.2023). *What Is HTML? Hypertext Markup Language Basics Explained.*

<https://www.hostinger.com/tutorials/what-is-html>

SSL-sertifikaatti mikä on ja miksi. (n.d). <https://www.aitoa.fi/ssl-sertifikaatti-mika-on-ja-miksi>

STK500. (n.d.).

<https://www.microchip.com/en-us/development-tool/atstk500>

Thompson, B. (8.7.2023). *What is C Programming Language? Basics, Introduction, History.*

<https://www.guru99.com/c-programming-language.html>

Arduino. (n.d.). *UNO WiFi Rev2.*

<https://docs.arduino.cc/hardware/uno-wifi-rev2>

Liite 1. Arduino koodi

```
1  /*Made by Juho Puonti 19.12.2018
2  Modified By Olli-Pekka Laukkanen 25.9.2023
3  */
4
5  #include <DHT.h>
6  #include <SPI.h>
7  #include <WiFi.h>
8  //Define librarys
9  #define DHTTYPE DHT11          // Define sensor type
10 #define DHT11Pin 2            // DHT11 Sensor pin
11 #define TEMPTYPE 0            // Temperature type
12
13 char ssid[] = "SSID";         // Your network SSID (name) change target WiFi SSID to "SSID" spot
14 char pass[] = "Password";     // Your network password change target WiFi key to "password" spot
15 int keyIndex = 0;             // Your network key
16 int status = WL_IDLE_STATUS;  // The Wifi radio's status
17
18 float fTemp, fHum;
19 int ledState=0;
20 int headerSent=0;
21 int incoming;                 //Variables
22
23 WiFiServer server(80);
24 DHT dht(DHT11Pin, DHTTYPE);
25
26 void setup() {                 //Initialize serial and wait for port to open:
27     dht.begin();
28     Serial.begin(9600);
29     while (!Serial) {
30     }
31
32     while (status != WL_CONNECTED) { //WiFi Connection
33         Serial.print("Attempting to connect to SSID: ");
34         Serial.println(ssid);
35         status = WiFi.begin(ssid, pass);
36         Serial.println(status);
37         delay(10000);
38     }
39     server.begin();
40     pinMode(ledState, OUTPUT);
41
42 }
43 void loop(){
44     WiFiClient client = server.available();
```

```

45 WiFiClient client = server.available();
46 headerSent=0;
47 if (client) {
48     Serial.println("New Client");
49     bool currentLineIsBlank = true;
50     while (client.connected()) {
51
52         if (client.available()) {
53             char page = client.read();
54             Serial.write(page);
55             if (incoming && page==' ')
56             {
57                 incoming=0;
58             }
59             if (page=='$')
60             {
61                 incoming=1;
62             }
63             if (incoming==1)
64             {
65                 headerSent=1;
66
67                 client.println("<!DOCTYPE HTML>");
68                 client.println("<body style=background-color:#ff6347;>");
69                 if(page=='1')
70                 {
71                     digitalWrite(LED_BUILTIN, LOW); //Page button function
72                     ledState = 0;
73                     headerSent=1;
74                     client.println("<meta http-equiv='refresh' content='0;URL=/'>\n");
75                 }
76                 if(page=='2')
77                 {
78                     digitalWrite(LED_BUILTIN, HIGH);
79                     ledState = 1;
80                     client.println("<meta http-equiv='refresh' content='0;URL=/'>\n");
81                 }
82             }
83         if (page == '\n' && currentLineIsBlank) {
84             if(headerSent == 0) {
85                 client.println("HTTP/1.1 200 OK");
86                 client.println("Content-Type: text/html");
87                 client.println(); //Base for HTML
88             }

```

```

90     client.println("<title>Arduino paikallisena verkkoserverinä</title>\n"); //Page title
91     client.println("<style>");
92     client.println("body { background-color:#ff6347; color : #FFF; font-size:12px; font-family: Calibri, Arial, Verdana; text-align:center; margin : 10px; } \n");
93     client.println("<htmlbutton { display:block; margin:0 auto; min-width:40px; max-width : 35%; background-color:#e28000; border-radius:5px; padding:5px;");
94     client.println("color:#FFF; font-weight:bold; color:#FFF; text-decoration:none; margin-bottom:10px; }>");
95     client.println("<htmlbutton:hover { border:2px solid white; }>"); //Button for html
96     client.println("</style>");
97     client.println("</head>\n<body>\n");
98     fTemp = dht.readTemperature(TEMP1TYPE);
99     fHum = dht.readHumidity(); //Variable for sensor data
100    Serial.println(fTemp);
101    Serial.println(fHum); //Print sensor data to serial monitor
102    client.println("<br />");
103    client.print("<span style=font-size:5em> Temperature: " );
104    client.print(fTemp);
105    client.print(" &degC");
106    client.println("<br />");
107    client.print(" Humidity: ");
108    client.print(fHum);
109    client.print(" %");
110    client.println("<br />");
111    if(ledState == 0) {
112        client.println("<h2>Page One</h2>\n");
113        client.println("<a class='htmlbutton' href='/$2'>Button One</a><br>");
114    }
115    else {
116        client.println("<h2>Page Two</h2>\n");
117        client.println("<a class='htmlbutton' href='/$1'>Button Two</a><br>");
118    }
119    client.println("</body>\n</html>");
120    break;
121 }
122 if (page == '\n') {
123     currentLineIsBlank = true;
124 }
125 else if (page != '\n') {
126     currentLineIsBlank = false;
127 }
128 }
129 }
130 delay(30);
131 client.stop();
132 Serial.println("Client disconnected");
133 }
134 }

```