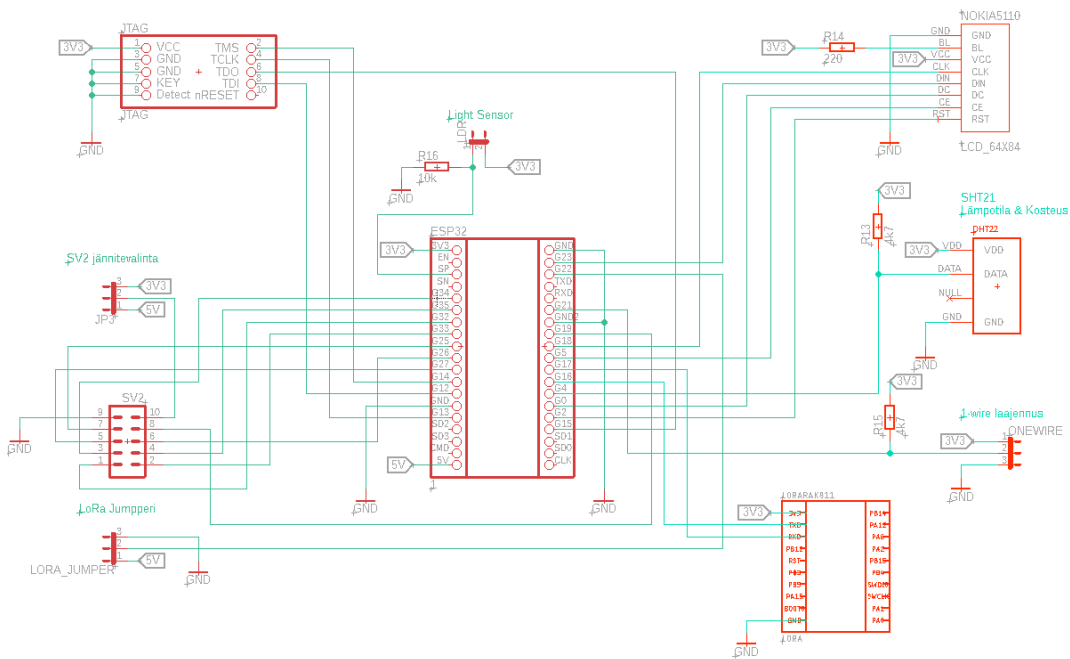


Ville Härkönen

Langattoman tiedonsiirron opetuskortti



Insinööri (AMK)
Tieto- ja viestintäteknikka
Kevät 2022



Tiivistelmä

Tekijä(t): Ville Härkönen

Työn nimi: Langattoman tiedonsiirron opetuskortti

Tutkintonimike: Insinööri (AMK), tieto- ja viestintätekniikka

Asiasanat: Sulautettu järjestelmä, tiedonsiirto, piirilevysuunnittelu, mikrokontrolleri, radio

Kajaanin ammattikorkeakoululle tarvittiin uusi mikrokontrollerin opetuskortti opiskelijoita varten. Uusi opetuskortti tuli opetuskäyttöön monelle eri kurssille, joissa käytetään mikrokontrolleritekniikkaa. Ennen opinnäytetyön valmistumista käytössä oli vanha opetusalausta, jossa oli 8-bittinen mikrokontrolleri ja joka toimii AVR-ohjelmointiympäristössä.

Työtä aloitettiin tekemään listaamalla vaatimuksia, mitä opetuskortista pitäisi löytyä. Tärkeimpiä vaatimuksia oli saada langaton LoRaWAN-radiomoduuli opetuskorttiin ja löytää siihen sopiva mikrokontrolleri. Mikrokontrollerista piti löytyä vähintään UART-tiedonsiirtotekniikka, jolla saadaan yhteys radiomoduuliin. Tämän lisäksi mikrokontrollerissa piti olla tarpeeksi muistia ja bittinopeutta, jotta se pystyisi käsittelemään radiomoduulia.

Mikrokontrolleriksi valittiin ESP32. Siitä löytyivät radiomoduuliin tarvittavat ominaisuudet ja se on helppo ohjelmoida. ESP32:ta pystyy ohjelmoimaan IDE-ohjelmointiympäristössä ja siihen löytyy internetistä paljon ohjelmointi esimerkkejä. Mikrokontrollerin ja radiomoduulin lisäksi opetuskorttiin suunniteltiin näyttö, kytkimiä, ledejä, antureita, JTAG- ja muita liittimiä.

Tässä opinnäytetyössä suunniteltiin alusta asti opetuskortti toimeksiantajan toiveiden mukaan, tutustuttiin erilaisiin komponentteihin, testattiin niiden toimivuutta ja tehtiin piirikorttisuunnitelma käyttäen EAGLE-ohjelmistoa. Testauksissa ESP32 ja komponentit kytkettiin kytkentäalustalle ja ohjelmointiin Arduino IDE-ohjelmistossa. Työn loppuvaiheessa opetuskortista jysyttiin prototyyppi ja tilattiin tehtaalta valmis opetuskortti.

Abstract

Author(s): Ville Härkönen

Title of the Publication: Wireless Communication Development Board

Degree Title: Bachelor of Engineering, ICT Engineering

Keywords: Embedded system, data transfer, board design, microcontroller, radio

Kajaani University of Applied Sciences needed a new microcontroller development board for students. The new development board was used for many different courses using microcontroller technology. Before completing this Master's thesis there was an old development board with an 8-bit microcontroller that operates in an AVR programming environment.

The work began by listing the requirements for what the development board should contain. One of the most important requirements was to get the LoRaWAN wireless radio module into the development board and find the right microcontroller for it. The microcontroller had to contain at least UART data transfer technology to connect to the radio module. In addition to this, the microcontroller had to have enough memory and bitrate to handle the radio module.

ESP32 was chosen as the microcontroller. It had the necessary features for the radio module and is easy to program. ESP32 can be programmed in an IDE programming environment and there are many programming examples on the internet. In addition to the microcontroller and radio module, a display, switches, LEDs, sensors, JTAG and other connectors were designed for the development board.

In this thesis, the development board was designed from the beginning according to the client's wishes. Different components were familiarized, their functionality was tested, and a circuit board plan using the EAGLE software was made. In the testing phase, ESP32 and components were connected to the breadboard and programmed using Arduino IDE software. At the end of the work, a prototype of the development board was milled, and a ready-made development board was ordered from the factory.

Sisällysluettelo

1	Johdanto	1
2	Sulautettu järjestelmä	2
3	Kehitysalustoja.....	7
3.1	Mikrokontrollerit.....	9
3.2	ESP32.....	12
3.3	Tiedonsiirto	15
3.3.1	Wi-Fi	20
3.3.2	Bluetooth	21
3.3.3	LoRa-modulaatio ja LoRaWAN-verkkoprotokolla	23
3.4	Työkalut.....	24
3.4.1	Arduino IDE	24
3.4.2	Piirilevysuunnittelu-ohjelmia	26
4	Opetuskortin suunnittelu ja toteutus.....	28
4.1	Vaatimukset	28
4.1.1	Komponenttien valinta	29
4.1.2	Komponenttien testaus.....	33
4.2	Piirilevyn suunnittelu	36
4.3	Piirilevyn toteutus	45
5	Pohdinta	50
6	Yhteenveto	51
	Lähteet	52
	Liitteet	58

Symboliluettelo

1-Wire	Yhden johtimen tiedonsiirtoväylä
ADC	Analog to Digital Converter, analogisen signaalin muunnos digitaaliseksi
AppEUI	LoRaWAN-radiomoduulin 64-bittinen laajennettu tunnus
AppKey	LoRaWAN-radiomoduulin salausavain
BLE	Bluetooth Low Energy, lyhyenkantaman langaton tiedonsiirtotekniikka
Bluetooth	Lyhyenkantaman langaton tiedonsiirtotekniikka
Counter	Mikrokontrollerissa laskuri, joka laskee tapahtumien määrän
CPU	Central Processing Unit, tietokoneen prosessori
DAC	Digital to Analog Converter, digitaalisen signaalin muunnos analogiseksi
DevEUI	LoRaWAN-radiomoduulin 64-bittinen henkilökohtainen tunnus
Eagle	Piirilevyn suunnitteluohjelmisto
Flash	Ohjelmamuisti
Full Duplex	Sarjaliikenteen lähetystila: lähettäjä sekä vastaanottaja voivat lähettää ja vastaanottaa dataa samaan aikaan
GPIO	General Purpose Input/Output, mikrokontrollerin dataliityntä
Half Duplex	Sarjaliikenteen lähetystila: lähettäjä sekä vastaanottaja pystyvät olemaan aktiivisia, mutteivat samaan aikaan
I2C	Kaksisuuntainen tiedonsiirtoväylä
IDE	Integrated Development Environment, ohjelmointiympäristö
Interrupt	Keskeytys, suorittimen tapahtuma, jonka avulla keskeytetään nykyinen tehtävä
IoT	Internet of Things, esineiden internet
JTAG	Joint Test Action Group, mikrokontrolleri ohjelmointiliityntä

LoRaWAN	Low Power Wide-Area Network, langaton tiedonsiirtotekniikka
RAM	Random Access Memory, muisti, johon tallennetaan väliaikaista dataa
Simplex	Sarjaliikenteen lähetystila, yksisuuntainen viestintä
SPI	Serial Peripheral Interface, synkroninen tiedonsiirtotekniikka
Timer	Ajastin, ylläpitää ohjelman ajoituksen kellon kanssa
UART	Universal Asynchronous Receiver Transmitter, kaksisuuntainen sarjaliikenne
Wi-Fi	Langaton tiedonsiirtotekniikka

1 Johdanto

Opinnäytetyön aiheena oli toteuttaa uusi langattoman tiedonsiirron opetuskortti. Aihe suunniteltiin talvella 2021 toimeksiantajien kanssa ja sitä aloitettiin toteuttamaan keväällä 2022. Uusi opetuskortti tulee Kajaanin ammattikorkeakoululle. He haluavat päivitetyn version opetuskortista, johon on lisätty LoRaWAN-radioyhteys. Työ suunniteltiin ja toteutettiin koulussa, jossa tarjottiin kaikki ohjelmat, laitteet ja komponentit opetuskortin toteuttamiseen.

Projektityö oli jaettu kahteen osaan. Yksi opiskelija suunnittelee ja toteuttaa opetuskortin ja toinen opiskelija tutustuu LoRaWAN-radioyhteyteen ja sen ohjelmointia opetuskorttiin. Molemmissa osissa oli niin paljon työskenneltävää ja tutkittavaa, että siinä olisi ollut liikaa tekemistä yhdelle opiskelijalle. Lisäksi aika ei olisi riittänyt, koska tämä piti saada tehtyä keväällä. Tässä työssä keskitytään opetuskortin toteutukseen. Valitsin tämän aiheen, koska minua kiinnosti tutustua piirilevyn suunnitteluun ja sen toteutukseen. Tässä työssä pääsi alusta asti tekemään piirilevyä, valitsemaan siihen sopivia komponentteja ja testailemaan niitä kirjoittamalla yksinkertaisen ohjelman.

Aihe oli kiinnostavaa toteuttaa koululle, koska uusi opetuskortti tuli opetuskäyttöön monelle eri kurssille ja samalla pääsi tutustumaan LoRa-tekniikkaan. Koululla ei vielä silloin ollut opetuskorttia, jossa olisi ollut LoRaWAN-radioyhteys. Opetuskortista löytyy lisäksi paljon muita hyviä ominaisuuksia, joihin opiskelijat voivat tutustua ja käyttää opetuksissa tai projektitöissä.

Tavoitteena oli oppia uutta piirilevyn suunnittelusta ja sen toteutuksesta. Siihen sisältyy komponenttien valinta, kytkentäkaavion suunnittelu, komponenttien juottaminen piirilevyyn, opetuskortin ohjelmointi ja testaukset. Sen lisäksi tutustua langattoman tiedonsiirtoon käyttäen LoRaWAN-radiotekniikkaa. Kaikkea uutta oppimista voisi hyödyntää tulevaisuudessa esimerkiksi työelämässä, jos pitää suunnitella samantyyppistä opetuskorttia tai luoda suunnitelma sen pohjalta.

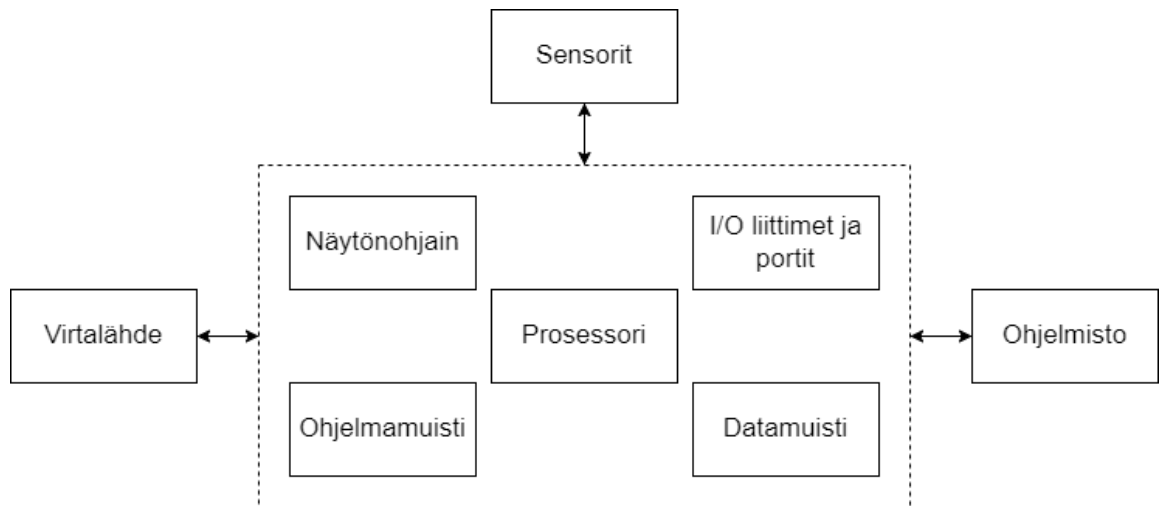
2 Sulautettu järjestelmä

Sulautettu järjestelmä on mikroprosessoripohjainen ja pienimuotoinen laitteisto, johon on lisätty ohjelmisto ja joka suorittaa sille annettuja tehtäviä. Järjestelmät voivat toimia joko itsenäisinä laitteina tai osana suurempia järjestelmiä, esimerkiksi mikroaaltouuni, auto tai MP3-soitin.

Nykyaikaisissa autoissa tai kuorma-autoissa on paljon sulautettuja järjestelmiä. Esimerkiksi yksi sulautettu järjestelmä ohjaa pyyhkijöitä, toinen näyttää kojelaudassa tietoja ja kolmas ohjaa päästöjä. Uusimmissa autoissa älypuhelimella voidaan yhdistää verkolla esimerkiksi integroituun navigaattoriin.

Useimmat tietokoneet ovat sulautettuja järjestelmiä. Ne sisältävät yleisesti: (kuva 1)

- Prosessori
- Näytönohjain
- Muisteja
- I/O-liittimet ja portit
- Sensorit
- Virtalähde
- Ohjelmisto [1]



Kuva 1. Sulautetun järjestelmän lohkokkaavio [1.]

Sulautetut järjestelmät jaetaan neljään luokkaan niiden toiminnallisuuden perusteella:

- Reaaliaikaiset (Real-time) sulautetut järjestelmät
- Itsenäiset (Standalone) sulautetut järjestelmät
- Verkotetut (Network) sulautetut järjestelmät
- Mobiili (Mobile) sulautetut järjestelmät [1]

Reaaliaikaiset sulautetut järjestelmät ovat nopeita järjestelmiä, joiden pitää näyttää tuloksia ja tuotoksia nopeasti. Tätä järjestelmää käytetään paljon eri aloilla, koska halutaan saada tärkeitä tietoja nopeasti, esimerkiksi lentokoneiden ohjauslaitteissa, automaattiohjauksessa, sensoridatan lähettämässä. Reaaliaikaiset sulautetut järjestelmät jaetaan kahteen luokkaan, koviin (hard real-time) ja pehmeisiin reaaliaikaisiin järjestelmiin (soft real-time) (taulukko 1). Kovassa reaaliaikaisessa järjestelmässä on tarkka määräaika jollekin tehtävälle, joka pitää saada tehtyä. Jos tehtävä jää tekemättä, se voi johtaa järjestelmän vikatilaa. Pehmeässä reaaliaika järjestelmässä ei olla tarkkoja, jos jokin tehtävä jää tekemättä määräajan sisällä. Sitä ei pidetä järjestelmän virheenä, vaan sen suorituskyky on heikentynyt. [1.]

Taulukko 1. Ero kovan ja pehmeän reaaliaikaisen järjestelmän välillä [2] [3]

Kova reaaliaikainen järjestelmä	Pehmeä reaaliaikainen järjestelmä
Datatiedoston koko on pieni	Datatiedoston koko on iso
Vasteaika on millisekunteina	Vasteaika on korkeampi
Huippukuormitus pitää olla ennakoitavissa	Huippukuormitus voidaan sietää
Järjestelmän turvallisuus on tärkeä	Järjestelmän turvallisuus ei ole niin tärkeä
Jos järjestelmässä tapahtuu virhe, se palautetaan ja tarkistetaan uudelleen	Jos järjestelmässä tapahtuu virhe, se palautetaan aiemmin muodostettuun tarkistuspiisteeseen
Käytetään ohjausjärjestelmissä muun muassa lentokoneissa	Käytetään multimedian suoratoistossa, peleissä, virtuaaliodellisuudessa.

Itsenäiset sulautetut järjestelmät eivät vaadi isäntätietokonetta toimiakseen, vaan voivat toimia itsenäisinä ja tehdä niille annettuja tehtäviä. Monet sulautetut järjestelmät toimivat jonkin elektronisen laitteen integroituna osana, muun muassa auton vakionopeudensäädin (kuva 2) ei toimi ilman että sen liittyy ajoneuvoon kiinni, joka toimii isompana järjestelmänä. [1.] Esimerkiksi laskimet, digikamerat, kodinkoneet ja mittausjärjestelmät ovat itsenäisiä sulautettuja järjestelmiä.



Kuva 2. Auton vakionopeudensäädin. [4]

Verkotetut sulautetut järjestelmät vievät dataa verkkoon tai web-palvelimiin joko langallisen tai langattoman verkon kautta. Kaikki sulautetut järjestelmät, jotka ovat yhteydessä verkkoon, luokitellaan verkotetuksi sulautetuiksi järjestelmiksi. [1.] Esimerkiksi turvajärjestelmissä, pankkiautomaateissa ja POS-järjestelmissä käytetään tätä järjestelmää.

Mobiili sulautetut järjestelmät ovat pieniä ja kannettavia järjestelmiä, kuten matkapuhelin, laskimet, tabletti ja läppäri. Kaikki mobiilijärjestelmät ovat käytännössä itsenäisiä järjestelmiä, mutta kaikki itsenäiset järjestelmät eivät ole mobiilijärjestelmiä [1]. Esimerkiksi pesukone on itsenäinen järjestelmä, muttei lasketa mobiilijärjestelmäksi sen koon perusteella.

Sulautetut järjestelmät voidaan jakaa kolmeen luokkaan mikrokontrollerien suorituskyvyn perusteella:

- Pienet sulautetut järjestelmät
- Keskikokoiset sulautetut järjestelmät
- Kehittyneet sulautetut järjestelmät [1]

Pienimuotoisissa järjestelmissä on 8- tai 16-bittinen mikrokontrolleri, keskikokoisessa on 16- tai 32-bittinen ja kehittyneissä on 32- tai 64-bittisiä mikrokontrollereja. Mitä enemmän bittejä löytyy

mikrokontrollerista, sitä nopeampi siinä on käsittelynopeus. [1.] Mikrokontrollerista kerrotaan lisää myöhemmässä kappaleessa.

3 Kehitysalustoja

Kehitysalusta on piirilevyllä tehty kytkentä, joka on tarkoitettu esimerkiksi opetuskäyttöön tai tuotekehitykseen. Kehitysalusta sisältää vähintään mikrokontrollerin, I/O-pinnejä ja jonkinlaisen USB-liittimen tai ulkoisen virtalähteen, jonka kautta alusta saa virtaa. Yleisimmin käytettyjä kehitysalustoja ovat Raspberry Pi ja Arduino. Niistä on tehty monenlaisia projekteja ja ovat helppokäyttöisiä. Molempiin pystyy liittämään erilaisia laitteita ja antureita, mittailemaan ja lähettämään dataa eteenpäin.

Raspberry Pi on pienikokoinen tietokone, jota käytetään monessa eri projekteissa. Sillä voi esimerkiksi tehdä sääaseman, pyörittää servereitä, toteuttaa IoT-pohjaisia sovelluksia ja mitata lämpötilaa ja kosteutta erilaisilla antureita. Sillä pystyy tekemään kaiken, mihin tavallinenkin tietokone pystyy, muun muassa selaamaan internetiä, katsomaan videoita, kirjoittamaan tekstiä ja pelaamaan videopelejä. Raspberry Pi avulla voi harjoitella ymmärtämään tietokoneen sisäistä toimintaa ja ohjelmoimaan erilaisia koodauskieliä, esimerkiksi Pythonia. [5.]

Raspberry Pi on hitaampi kuin tavallinen tietokone, mutta se tarjoaa paljon erilaisia ominaisuuksia. Siihen pystyy liittämään oheislaitteita, kuten näppäimistö, hiiri ja näyttö, se toimii pienellä virrankulutuksella ja siitä löytyy SD-korttipaikka. SD-kortin kokoa suositellaan vähintään 8 Gt. Raspberry Pi -laudasta löytyy auki kammattuna GPIO-liityntöjä, johon voi liittää laitteita, ledejä, moottoreita ja antureita ja josta voi ohjata niitä. Lisäksi laudasta löytyy SPI-, I2C-, I2S- ja UART-moduuleja. [6.]

Raspberry Pi:stä on olemassa monta erilaista versiota, jotka on lueteltu alla olevaan taulukkoon (taulukko 2):

Taulukko 2. Raspberry Pi -lautoja ja niiden ominaisuuksia [6] [7]

Ominaisuudet	Raspberry Pi 1 Model B+	Raspberry Pi 2 Model B	Raspberry Pi Zero	Raspberry Pi 3 Model B	Raspberry Pi 4 Model B
Proessori	700 MHz, ARM11, BCM2835	900 MHz, Quad Cortex A7, BCM2836	1 GHz, ARM11, BCM2835	1.2 GHz, Quad Cortex A53, BCM2837	1.5 GHz, Quad Cortex A72, BCM2711
RAM	512 MB SDRAM	1 GB SDRAM	512 MB SDRAM	1 GB SDRAM	1/2/4/8 GB SDRAM
GPU	250 MHz Videocore IV	250 MHz Videocore IV	250 MHz Videocore IV	400 MHz Videocore IV	500 MHz Videocore IV
Muistikortti	Mikro-SD	Mikro-SD	Mikro-SD	Mikro-SD	Mikro-SD
GPIO	40	40	40	40	40
Langattomat yhteydet	Ei ole	Ei ole	Ei ole	Wi-Fi ja Bluetooth	Wi-Fi ja Bluetooth
Virtalähde	5V Mikro USB	5V Mikro USB	5V Mikro USB	5V Mikro USB	5V USB tyyppi-C

Raspberry Pi:hin pystyy asentamaan erilaisia käyttöjärjestelmiä, muun muassa Archlinux, RISC OS, Windows 10 IOT Core, Ubuntu. Monet Raspberry Pi -laitteet käyttävät Raspbian käyttöjärjestelmää, joka on virallinen käyttöjärjestelmä Raspberry Pi:lle [6]. Se on optimoitu hyvin Raspberry Pi -laitteille, se on ilmainen, se sisältää erilaisia työkaluja nettiselailuun, ohjelmointiin ja tekstinkäsittelyyn. Käyttöjärjestelmät asennetaan SD-kortille, joka on helppo vaihtaa, kun tehdään erilaisia projekteja eivätkä kaikki ole samassa SD-kortissa. Tällä hetkellä nopein Raspberry Pi on 4 versio (kuva 3).

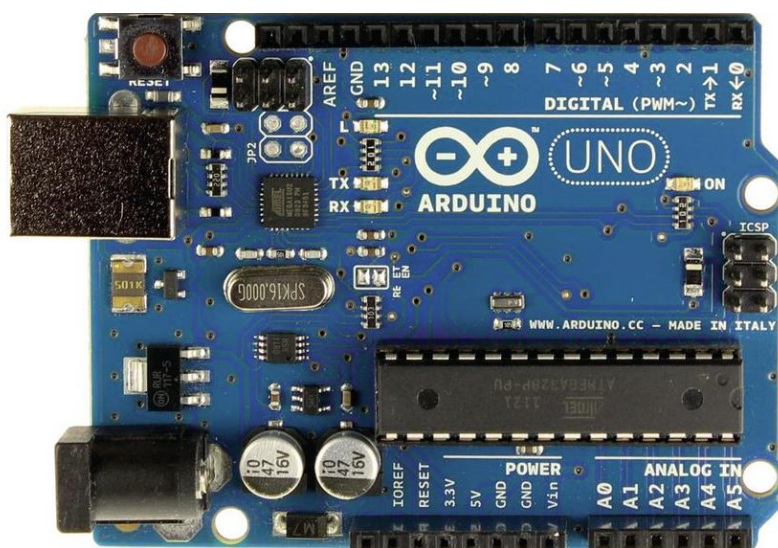


Kuva 3. Raspberry Pi 4. [8]

Arduino on pienikokoinen piirilevy, jota käytetään paljon elektroniikkaprojektien rakentamiseen. Se on helppokäyttöinen, edullinen ja avoimen lähdekoodin alusta, johon pystytään kirjoittamaan ja lataamaan koodeja, esimerkiksi sytyttämällä LED-valon, ohjaamaan moottoria tai lukemaan anturidataa [9]. Tämä alusta sopii hyvin uusille aloittelijoille, jotka ovat kiinnostuneita työskentelemään elektroniikkojen parissa.

Arduino-levyn koodin ajaminen on todella helppoa ja levy ei tarvitse mitään käyttöjärjestelmän asennusta. Koodin ajamiseen tarvitsee tietokone, johon on asennettu Arduino IDE-ohjelma ja USB-kaapeli, jonka kautta Arduinoon saadaan ajettua koodi sisään. Arduino IDE -ohjelmasta löytyy paljon esimerkkejä, joilla pystyy testaamaan Arduinoa. IDE ohjelma käyttää C++-kieliversiota.

Arduinosta on olemassa monta erilaista levyä. Yleisimmin ja suosituin käytetty Arduino-levy on Arduino Uno [10]. Se on hyvä valinta aloittelijoille ja on edullinen. Nettikaupoista löytyy Arduino Uno Starter Pack, josta löytyy Arduino Uno -levy (kuva 4), antureita, ledejä ja hyppylankoja. Unon-levyn lisäksi on olemassa LilyPad-, RedBoard-, Mega- ja Leonardo-levyjä, joilla kaikilla on erimäärä pinnipaikkoja, muistin määrä ja lukunopeus [10]. Lähes kaikista Arduino -levyistä löytyy vähintään GND, 5V, Analog, Digital, PWM- ja AREF-pinnipaikkoja.



Kuva 4. Arduino Uno. [10]

3.1 Mikrokontrollerit

Useimmat sulautetut järjestelmät käyttävät mikrokontrolleria laskentatoimintojen ja funktioiden suorittamiseen. Mikrokontrolleri on pieni laite, joka ohjaa, käsittelee ja tallentaa tietoja sisäiseen

muistiin. Muistin koko vaihtelee eri mikrokontrollerissa. Mikrokontrolleri sisältää CPU:n, kellon, muistit ja muut sisäiset oheislaitteet.

Mikrokontrollerista on olemassa monibittisiä versioita, esimerkiksi 8-, 16- ja 32-bittisiä. 8-bittinen mikrokontrolleri suorittaa logiikka- ja aritmeettisiä operaatioita. Esimerkiksi Intel 8031/8051 käyttävät 8-bittistä mikrokontrolleria. 16-bittinen suorittaa paremman tarkkuuden ja suorituskyvyn kuin 8-bittinen. Intel 8096 käyttää 16-bittistä mikrokontrolleria. 32-bittistä käytetään automaattilaitteissa, kuten toimistokoneissa ja lääketieteellisissä laitteissa. ESP32 käyttää 32-bittistä mikrokontrolleria. [11.]

CPU on mikrokontrollerissa keskeisin osa, jonka ympärillä kaikki toimii ja keskustelee. Lyhyesti sanottuna CPU on mikrokontrollerin aivot, jossa suoritetaan kaikki käyttäjän antamat käskyt. Käskyn suorittamista varten mikrokontrolleri vaatii kellotaajuuden, koska ilman sitä käskyä ei pystytä suorittamaan. Kristallioskillaattorin avulla saadaan mikrokontrolleriin lisättyä kellotaajuuden. Kun käyttäjä antaa käskyn mikrokontrollerille, CPU alkaa hakemaan käskyn mikrokontrollerin muistista (ROM), purkaa ja suorittaa sen. Tätä tekniikkaa kutsutaan nimellä pipelining. Pipelining suoritetaan data- ja osoiteväylien avulla. CPU toimii joko MegaHertz- tai GigaHertz-nopeudella. Nopeudet vaihtelevat eri CPU:ssa. [12.]

Rekisterit ovat datan tallennuselementtejä. Se tallentaa 8-bittisiä binäärisanoja. Mikrokontrollereihin on lisätty erilaisia yleiskäyttöisiä rekisterejä ja oheislaiterekisterejä. Yleiskäyttöisiä rekistereitä ovat ohjelmalaskurit ja pino-osoitin datan ja ohjeiden tallentamiseen. Oheislaiterekisterit ovat hyödyllisiä esimerkiksi laitteiston määrittämisessä mikrokontrollerissa. [12.]

I/O-portteja (tulo-/lähtö) kutsutaan yleisesti GPIO-porteiksi (General purpose input output). Näitä portteja voidaan käyttää joko tuloina tai lähtöinä ja niitä voidaan käyttää erilaisiin toimintoihin. GPIO-portteja tarvitaan ulkoisten laitteiden liittämiseen, esimerkiksi LCD-näyttö, moottori, mittarit.

Kun ohjelmoimme mikrokontrollerin suorittamaan jonkun ohjelman, se pitää ensin tallentaa mikrokontrollerin muistiin. Mikrokontrollerista löytyy kolmen tyyppisiä muisteja: Flash, RAM ja EEPROM. Flash muistiin tallennetaan pysyvää dataa, joka säilyy siellä, vaikka mikrokontrollerista katkaistaan virta. Flashia käytetään usein tiedon tallentamiseen. RAM (Random Access Memory) muistiin tallennetaan väliaikaisesti dataa, esimerkiksi silloin, kun ohjelmaa ajetaan aktiivisesti. EEPROM-muisti on kuten Flash-muisti, eli sinnekin voi tallentaa pysyvää dataa. Se on kuitenkin hitaampi ja siinä on vähemmän muistia, joten tämän takia Flashia käytetään enemmän tiedon tallentamiseen. [13.]

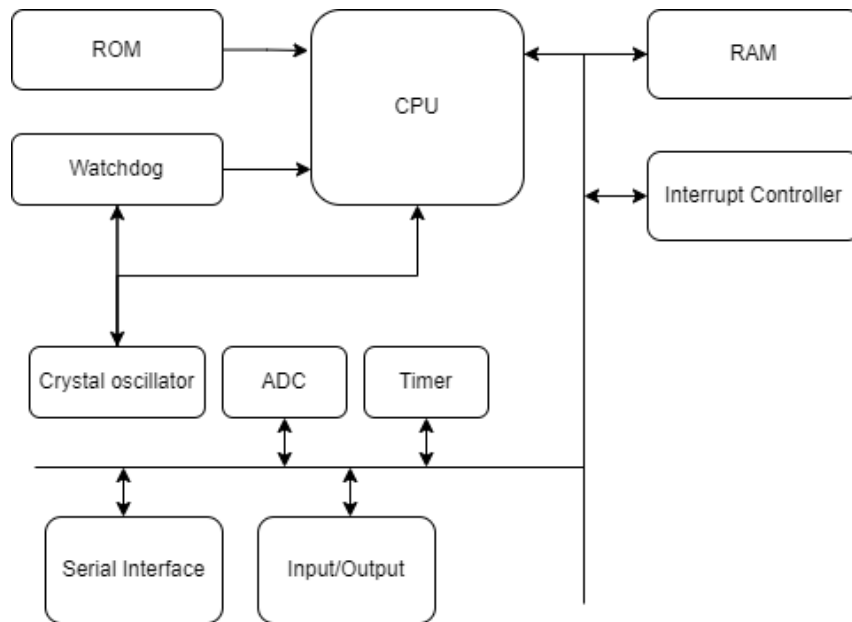
Ajastin (Timer) on tärkeä sovellus sulautetuissa järjestelmissä. Se ylläpitää ohjelman ajoituksen synkronoituna järjestelmän kellon tai ulkoisen kellon kanssa. Se pystyy toistamaan ennalta määritellyn käskyn tietyllä ajanjaksolla. Ajastimen sovelluksia ovat esimerkiksi moottorin ohjaus, GPS, kytkimen päälle/pois-toiminto. [14.]

Laskuri-sovellus (Counter) laskee tapahtumia, jotka tapahtuvat mikrokontrollerin ulkopuolella. Laskuria käytetään esimerkiksi mittauksissa, esineiden tunnistamiseen ja lämpötilan nostamiseen tai laskuun tietyssä pisteessä. [12.]

Eri laitteet keskustelevat keskenään sarjaviestinnällä. Sarjaviestintää suoritetaan erilaisilla kytkennöillä, esimerkiksi SPI, I2C, 1-Wire [15]. Sen lisäksi laitteet pystyvät keskustelemaan myös langattomasti, esimerkiksi Wi-Fi-, Bluetooth- tai LoRaWAN-radion kautta.

Kun anturi kytketään mikrokontrolleriin ja mitataan arvoja, siitä tulee fysikaalisia parametrejä, kuten lämpötila, kosteus tai painearvoja, jotka muunnetaan analogisiksi signaaleiksi. Analoginen signaali voi olla esimerkiksi joku jännitearvo. ADC (analog to digital converter) avulla voidaan muuntaa nämä analogiset signaalit digitaalisiksi tavuiksi. DAC (digital to analog converter) on ADC vastakohta, eli se muuntaa digitaalisen datan analogiseen signaalimuotoon. DAC käytetään esimerkiksi digitaalisessa potentiometrissä. [12.]

Keskeytykset (Interrupts) ovat tapahtumia, joita käytetään silloin, kun suoritetaan enemmän kuin yhtä tehtävää. Keskeytyksen avulla voidaan mikrokontrollerissa keskeyttää nykyisen tehtävän ja suorittaa muita tehtäviä. Kun muut tehtävät ovat suoritettu, palataan takaisin ensimmäiseen tehtävään ja suoritetaan se loppuun (kuva 5). [12.]



Kuva 5. Mikrokontrolleriarkkitehtuuri. [10]

3.2 ESP32

ESP32 on pieni, edullinen ja vähän virtaa kuluttava 32-bittinen mikrokontrolleri, johon on integroitu paljon erilaisia ominaisuuksia, muun muassa Wi-Fi- ja Bluetooth-yhteydet. Espressif Systems teki ensimmäisen ESP32:n vuonna 2016 [16]. Espressif ovat kehittäneet ja julkaisseet erilaisia ESP32 variaatioita vuosien aikana ja päivittänyt vanhoja moduuleja. On olemassa yksi- ja kaksiytimisiä ESP32:n laitteita, kuten yksiytimisiä ESP32-SOLO-sarjan moduuleja ja kaksiytimisiä ESP32-WROOM-32-moduuleja. Noiden lisäksi on olemassa myös WROVER-sarjan moduuleja [17].

ESP32:n prosessori on ESP32-D0WDQ6, jonka toimintaa ohjaa joko yksi- tai kaksiytiminen Xtensa 32-bittinen LX6-mikroprosessori. Se toimii maksimissaan 240MHZ taajuudella. [16.]

Tekniset tiedot ESP32:

- Ohjelmamuisti (Flash): 32MB
- Datamuistia (SRAM): 520KB
- GPIO: 34
- Debugger: JTAG

- Ajastin (Timer): 2 x 64-bit
- AD-muunnin: 18 kpl 10-bit tuloa
- Tiedonsiirto: SPI, I2C, UART
- Pääominaisuudet: Wi-Fi, Bluetooth ja BLE [17] [18]

ESP32 WROOM on todella suosittu ESP32-laite, jota käytetään monessa eri projekteissa (kuva 6). Sitä on helposti saatavilla eri kaupoista, esimerkiksi Amazonista. WROOM-laitteessa on mikro-USB-liitin, jonka kautta laite saa virtaa tietokoneelta ja voidaan syöttää koodia joko Arduino IDE tai jonkun muun IDE-ohjelman kautta. Laitteesta löytyy kaksi nappia, BOOT- ja EN-nappia. BOOT-nappi käynnistää käynnistyslaitain vaihteen ja EN-nappi toimii reset-nappina, jolla voidaan käynnistää laitteen uudelleen. WROOM-laitteessa on yhteensä 38 kappaletta pinnipaikkoja.



Kuva 6. ESP32 WROOM. [19]

WROOM-laitteesta on olemassa myös samanlainen ESP32-laite nimeltä TTGO LoRA SX1276 OLED (kuva 7). Siihen on integroitu 0,96-tuumainen OLED-näyttö kiinni ja siinä on 2 vähemmän pinnipaikkoja. Tämä laite on hyvä esimerkiksi IOT-projekteissa, jos halutaan reaaliaikaisesti seurata laudan toimintoa näytön kautta. Laudassa on samanlainen mikro-USB-liitin ja laudan pohjalta löytyy erillinen liitin, johon voi ulkoisen virtalähteen kytkeä laitteeseen kiinni. [20.]



Kuva 7. ESP32 TTGO LoRA SX1276 OLED. [20]

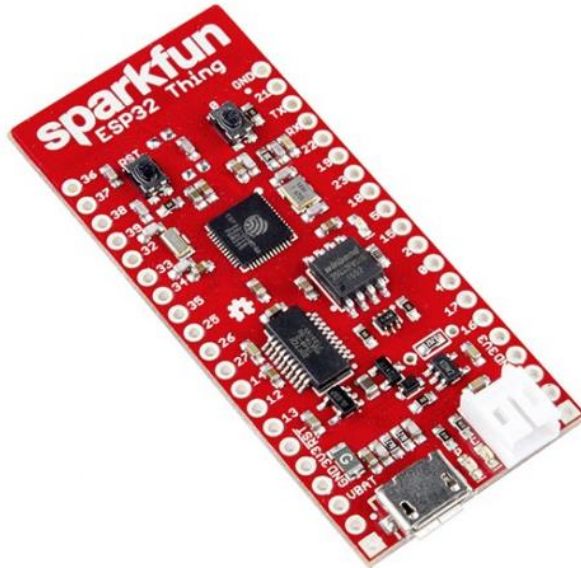
ESP32:sta on olemassa myös vielä pienempiä versioita. Esimerkiksi ESP32-CAM-laite (kuva 8), josta löytyy 16 pinnipaikkoja ja 9 IO-portteja. CAM-laitteeseen on integroitu kiinni OV2640-kamera ja TF-korttipaikka, jotka käyttävät ESP32 IO-portteja. CAM-laitteesta ei löydy samanlaista mikro-USB-liitintä kuin muissa ESP32-laitteissa, vaan siihen joutuu asentamaan erikseen FTDI-adapterin, jotta pystyy ohjelmoimaan laitetta. [21.] [22.]



Kuva 8. ESP32-CAM. [21]

Lopuksi on olemassa myös Sparkfun ESP32 Thing -laite (kuva 9). Sparkfun tuotteet ovat usein erittäin huippulaatuisia, ne ovat hyvin dokumentoitu ja näille on saatavilla hyviä tutoriaaleja. ESP32 Thing -laitteesta löytyy mikro-USB-liitin ja sen vierestä erillinen liitin, johon voi kytkeä ulkoisen

virtalähteen. Sparkfun käyttää erilaista ESP32-moduulia kuin muissa ESP32-laitteissa. Se on kooltaan pienempi kuin muut moduulit. ESP32 Thing -laitteesta löytyy 40 kappaletta pinnipaikkoja. [23.]



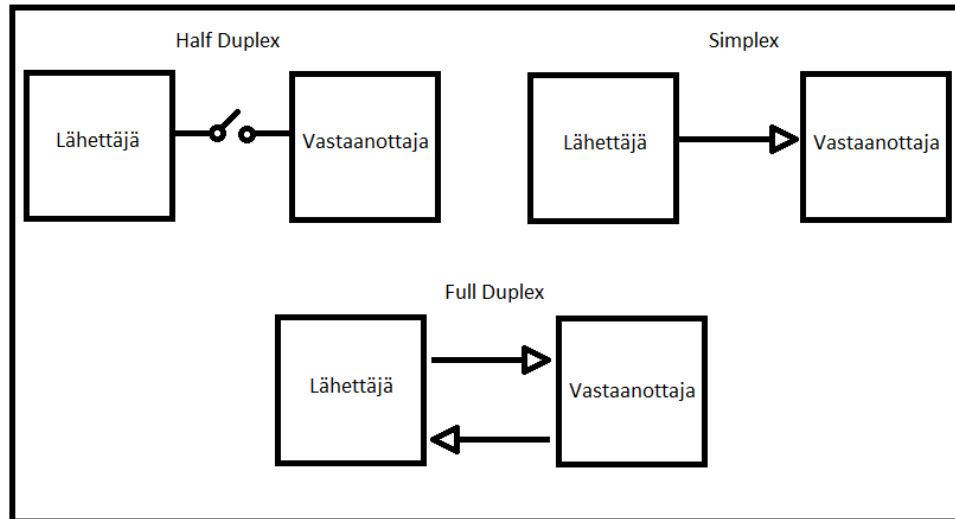
Kuva 9. Sparkfun ESP32 Thing. [23]

3.3 Tiedonsiirto

Sarjaliikenne on yleisimmin käytetty tapa siirtää tietoa laitteiden välillä. Laite voi olla esimerkiksi tietokone, puhelin tai tabletti, joiden välillä voidaan vaihtaa videoita, kuvia ja asiakirjoja. Sulaute- tussa järjestelmässä sarjaliikenteen avulla voidaan vaihtaa dataa eri menetelmien avulla, joko langallisella tai langattomalla tekniikalla. Dataa käsitellään binäärisessä muodossa. Tunnetuimpia tiedonsiirtoon käytetyt rajapintoja ovat RS-232, RS-485, 1-Wire, I2C ja SPI. [24.]

Sarjaliikenteelle on olemassa erilaisia lähetystiloja. Ne ovat luokiteltu kolmeen luokkaan: Simplex, Half Duplex ja Full Duplex (kuva 10). Jokaisessa lähetystilassa on määritelty lähettäjä, josta data lähtee, ja datan vastaanottaja. Simplex-tilassa on vain yksisuuntainen viestintä, jossa lähettäjä lähettää dataa vastaanottajalle, joka hyväksyy sen. Vain joko lähettäjä tai vastaanottaja voivat olla aktiivisia kerrallaan. Half Duplex -tilassa sekä lähettäjä että vastaanottaja pystyvät olemaan aktiivisia, muttei samaan aikaan, jos esimerkiksi lähettäjä lähettää dataa, vastaanottaja voi hyväk-

syä sen, mutteivat pysty lähettämään dataa takaisin. Sama toisinpäin. Full Duplex -tilassa lähettäjä sekä vastaanottaja voivat lähettää ja vastaanottaa dataa samaa aikaan. Tätä tekniikkaa käytetään laajasti monessa projekteissa ja on suosituimpia. [24.]



Kuva 10. Sarjaliikenteen lähetystiloja.

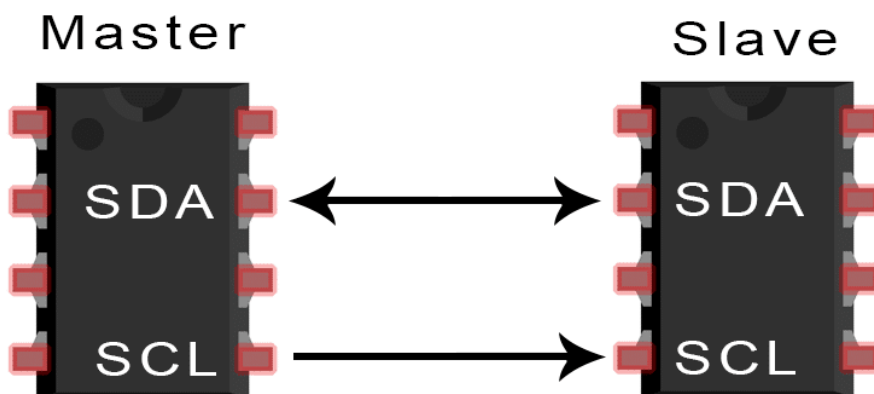
Dataa voidaan siirtää kahdella eri tavalla: sarjaliikenteellä tai rinnakkaisliikenteellä (taulukko 3). Sarjaliikenteessä dataa lähetetään aina bitti kerrallaan, ne vaativat vähemmän IO-linjoja, ovat halpoja asentaa ja pystyvät välittämään dataa pitkälle matkalle. Tätä käytetään esimerkiksi modeemeissa. Rinnakkaisliikenteessä voidaan lähettää monta bittiä kerralla, niin sanottuina datapaloina, esimerkiksi 8, 16 tai 32 bittiä. Jokainen databitti vaatii oman IO-linjan. Rinnakkaisliikenteessä on nopea viestintänopeus, mutta se käyttää enemmän IO-linjoja ja se on kallis asentaa. Rinnakkaisliikennettä käytetään esimerkiksi tietokoneissa, muun muassa CPU:n, RAM-muistin, modeemien ja muiden laitteistojen yhdistämiseen. [24.]

Taulukko 3. Sarjaliikenteen ja rinnakkaisliikenteen ero [24]

Sarjaliikenne	Rinnakkaisliikenne
Lähetää bitti kerrallaan yhdellä kellopulsilla	Lähetää monta bittiä kerrallaan
Vaatii yhden johdon datan lähettämiseen	Vaatii "x" määrän johtoja "x" määrän bitin lähettämiseen
Hidas viestintänopeus	Nopea viestintänopeus
Halpa asentaa	Kallis asentaa
Pitkän matkan viestintä	Lyhyen matkan viestintä

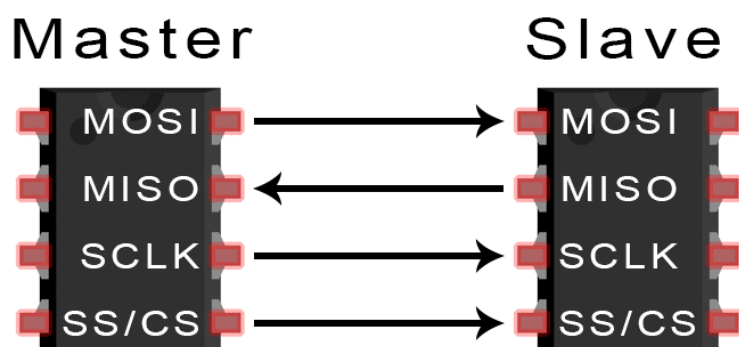
Sarjaliikenteen protokollit jaetaan kahteen eri tyyppiin niiden tiedonsiirtojen mukaan: synkroninen ja asynkroninen tiedonsiirto. Synkronisessa tiedonsiirrossa laitteet käyttävät yhtä väylää datan siirrossa ja kellosignaalina. Tämä tiedonsiirto on nopea, vaatii vähemmän IO-linjoja ja käyttää master/slave-konfigurointia. Esimerkiksi I2C ja SPI käyttävät tätä tiedonsiirtona. Asynkronisessa tiedonsiirrossa ei ole ulkoista kellosignaalia, mutta siinä pystyy säätämään tiedonsiirtonopeutta, ohjaamaan datavirtaa, lähetystä, vastaanottoa ja virheenhallintaa. [25.] Asynkroniset sopivat vakaaseen viestintää ja näitä käytetään pitkän kantaman sovelluksissa. Esimerkiksi RS-232 ja RS-485 ovat asynkronisia protokollia. [24.]

I2C-protokolla on kaksijohtiminen, kaksisuuntainen tiedonsiirto, jota käytetään datavaihtoon eri laitteiden välillä (kuva 11). Samaan väylään voidaan kytkeä monta laitetta, jopa 1024 laitetta. I2C käyttää 7- tai 10-bittistä osoitetta, joka mahdollistaa monen laitteen yhdistämistä, mutta se vaatii kellosignaalin, jotta voidaan luoda käynnistys- ja pysäytysehtoja. Tämän protokollan tiedonsiirtonopeus on 400 kbps. [24.]



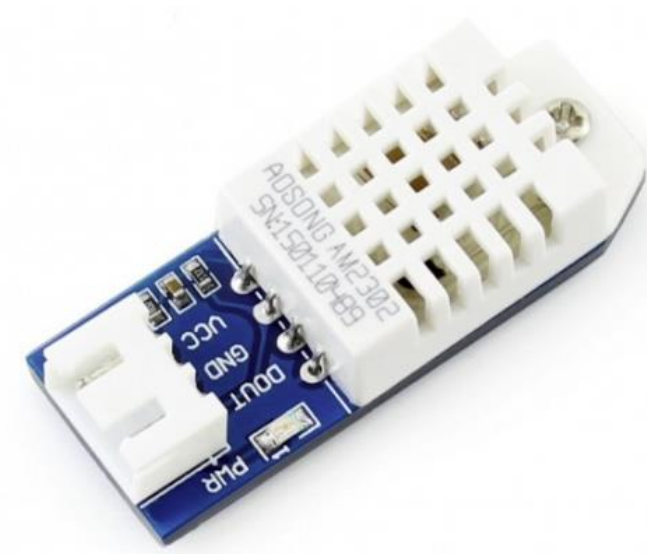
Kuva 11. I2C-protokolla. [26]

SPI-protokolla lähettää ja vastaanottaa dataa jatkuvana virtana, ilman keskeytyksiä. SPI on nelijohtiminen sarjaliitäntä, jossa on MOSI (Master out slave in), MISO (Master in slave out), kello- ja orjasignaalit (kuva 12). Tätä protokollaa suositellaan, jos tarvitaan nopeaa tiedonsiirtoa. Sen enimmäisnopeus on 10 Mbps. Tähän pystyy kytkemään lähes rajattoman määrän orjalaitteita, mutta se riippuu väylän kuormakapasitanssista. [24.]



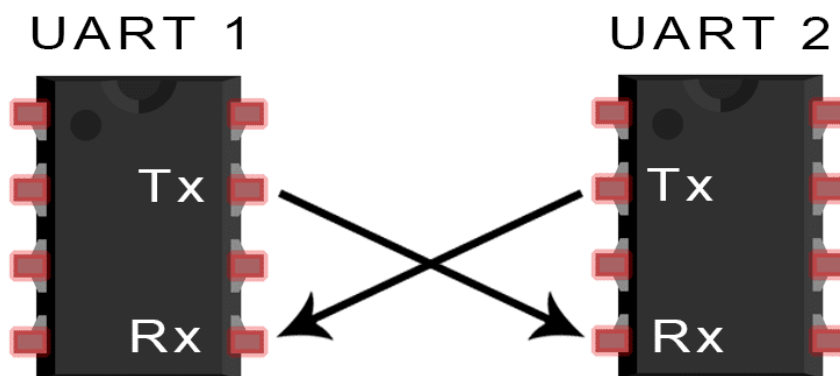
Kuva 12. SPI-protokolla. [27]

1-Wire-protokolla on samanlainen kuin I2C-protokolla, mutta erona on siinä, että 1-Wire käyttää vain yhtä datalinjaa ja maadoitusta. Se ei vaadi omaa kello-signaalia, vaan se kelloitetaan sisäisellä kideoskillaattorilla. 1-Wire käyttää 64-bittistä osoitejärjestelmää, käyttää Half Duplex -lähetystila ja tukee pitkän matkan viestintää, mutta toimii pienemmällä nopeudella. [24.] Monet anturit käyttävät 1-Wire tekniikkaa, esimerkiksi DHT22-anturi (kuva 13).



Kuva 13. DHT22 kosteus- ja lämpötila-anturi. [28]

UART (Universal Asynchronous Transmitter Receiver) (kuva 14) on yleisin kaksisuuntaisessa sarjaviestinnässä käytetty protokolla, joka löytyy mikrokontrollerin sisältä. Tämä laite on suunniteltu suorittamaan asynkronista viestintää ja muuntaa saapuvat ja lähtevät datat sarjamuotoiseksi binaärivirraksi. UART ei vaadi montaa johdinta; yhden johtimen datan lähettämiseen ja toisen johdon datan vastaanottamiseen. UART-protokollasta on olemassa myös toinen versio nimeltä USART. USART-protokolla ei ole teknisesti sama asia kuin UART, mutta niillä on sama määritelmä. Suurin ero niiden välillä on, että UART tukee vain asynkronista viestintää, kun USART tukee sekä asynkronista että synkronista viestintää. [29.]



Kuva 14. UART-protokolla. [30]

3.3.1 Wi-Fi

Yksi ESP32 tärkeimmistä ominaisuuksista, joka tekee siitä hyvän IOT-sovelluksiin, on sen sisäänrakennettu Wi-Fi ominaisuus. Listataan tähän ESP32:n Wi-Fi ominaisuudet:

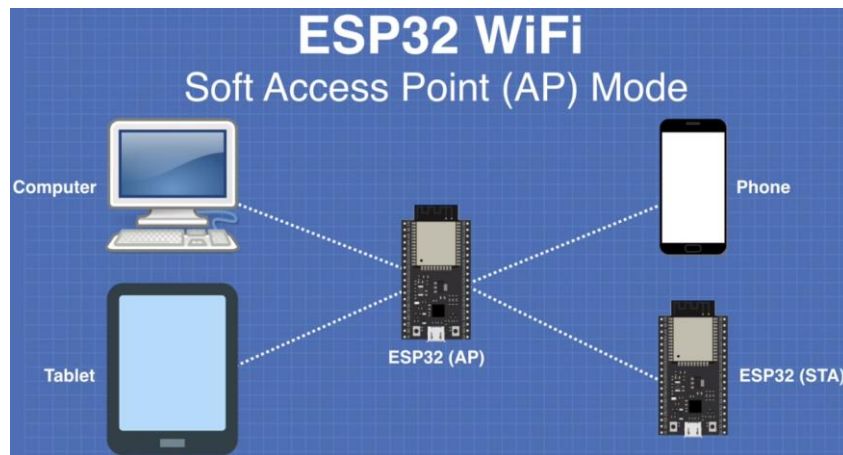
- ESP32 tukee Wi-Fi protokollia 802.11 b/g/n
- Toimii 2,4 GHz taajuudella
- Tukee datanopeutta 150 Mbit sekunnissa
- Säädettävä lähetysteho 20 dBm
- Arduino IDE käyttää Wi-Fi-kirjastoa
- Tukee ESP32:ta joko asemana (Station) tai pehmeänä tukiasemana (Soft Access Point) [16] [18]

Station-tilassa ESP32 on reitittimen asiakas, minkä kautta Wi-Fi yhteys toimii muihin asiakaslaitteisiin, kuten tietokoneeseen, tablettiin tai puhelimeen (kuva 15). Soft Access Point -tilassa ESP32 on verkon keskus, eli tukiasema, joka tarjoaa yhteydet ulkoisille laitteille, ja jopa toiselle ESP32-laitteelle, joka on Station-tilassa. Soft Access Point nimi tulee siitä, kun ESP32:lla ei ole suoraa yhteyttä internettiin, vaan sen ympärillä oleviin laitteisiin. [18.]



Kuva 15. ESP32 asematilana. [18]

Station-tilassa ESP32 saa IP-osoitteensa reitittimistä ja Access Point-tilassa ESP32 määrittää IP-osoitteensa asiakkaille. Kun käytetään Access Point-tilaa, ESP32 pystyy sallimaan yhteyden enintään 5 asiakkaalle samaan aikaan (kuva 16). Molemmissa tiloissa ESP32 voi toimia itsenäisenä web-palvelimena. Arduino IDE -ohjelmassa Wi-Fi-kirjaston sisältä löytyy monta esimerkkiä, joilla pystyy testaamaan Wi-Fi-ominaisuuksia, muun muassa WiFi Scanner, WiFi Access Point ja WiFi Simple Server. [18.]



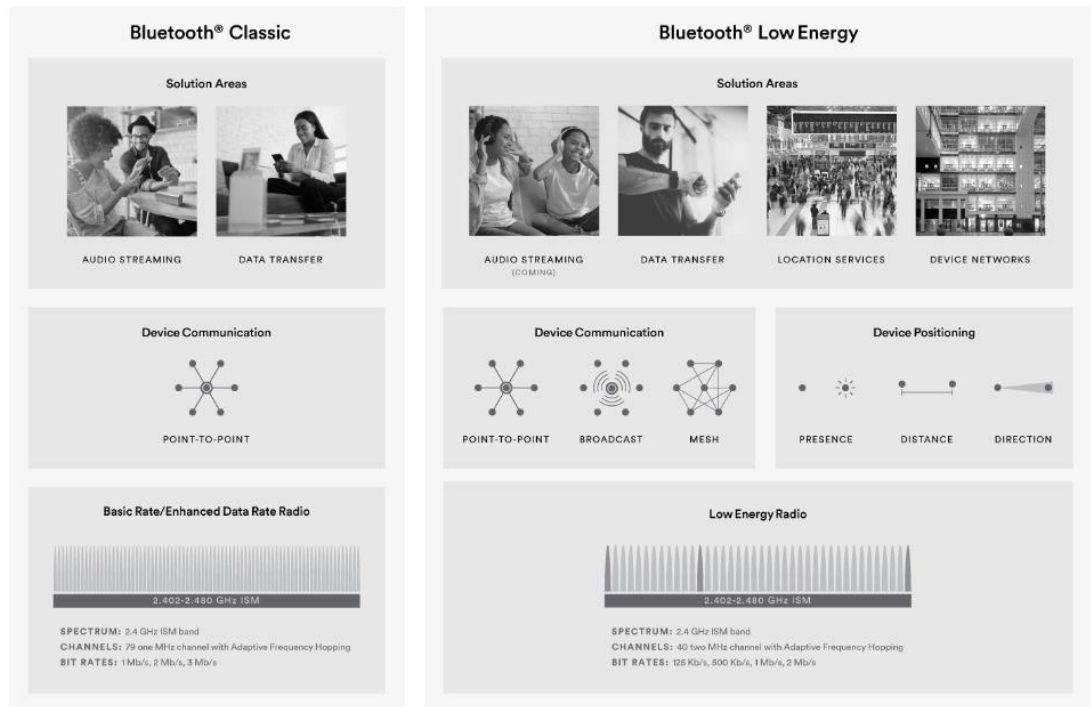
Kuva 16. ESP32 pehmeänä tukiasemana. [18]

3.3.2 Bluetooth

Bluetooth on radiotekniikkaan perustuva langaton tiedonsiirtotekniikka, joka toimii lyhyellä kantamalla laitteiden välillä. Sitä hyödynnetään monissa eri laitteissa ja nykyään kaikista älypuhelimista löytyy joku Bluetooth versio. Bluetooth on olemassa kaksi eri radiotekniikkaan, Bluetooth Classic ja Bluetooth Low Energy (BLE) (kuva 17). Molemmat on suunniteltu lähettämään ja vastaanottamaan dataa eri laitteiden välillä. [31.]



The global standard for simple, secure device communication and positioning



Kuva 17. Bluetooth Classic ja Bluetooth Low Energy. [31]

Bluetooth ensimmäinen versio 1.0 julkaistiin 1999, jolloin oheislaitteet pystyttiin yhdistämään langattomasti tietokoneeseen. Silloin tiedonsiirtonopeus oli vain noin 1 Mbps ja kantama noin 10 metriä. Silloin ei vielä pystynyt lähettämään ja vastaanottamaan suuria tiedostoja. 2.0 versiossa tiedonsiirtonopeus nousi noin 3 Mbps ja kantama noin 31 metriä. Tärkeimmät erot kaikkien versioiden välillä on tiedonsiirtonopeus, kantama ja viestien lähetyskoko. [32.]

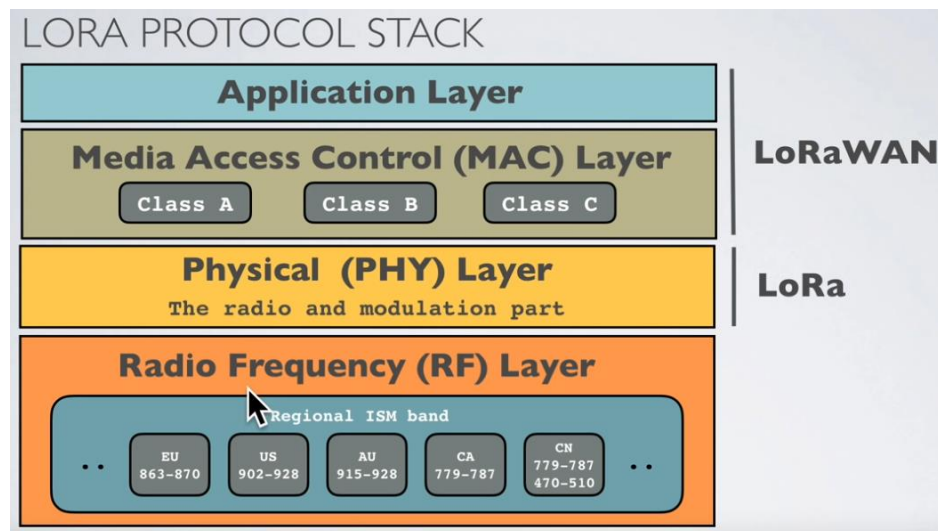
Suurin osa laitteista käyttävät tällä hetkellä Bluetooth 4.2 versiota, esimerkiksi langattomat kuulokkeet, hiiret, näppäimistöt ja äänilaitteet. Bluetooth 4.2 julkaistiin vuonna 2010, jonka mukana tuli myös Bluetooth Low Energy. Sitä käytetään vähän virtaa kuluttavissa laitteissa, esimerkiksi sykemittareissa ja esineiden internet (IoT) sensoreissa. Vuonna 2020 julkaistiin Bluetooth 5.2, joka on paljon tehokkaampi kuin 4.2 versio. [33.]

ESP32 sisältää Bluetooth Classic ja Bluetooth Low Energy. ESP32 voi toimia joko Bluetooth asiakkaana tai palvelimena. Bluetooth käytetään jatkuvaan datan suoratoistoon ja BLE käytetään lyhyisiin datan lähetyksiin. Arduino IDE-ohjelmasta löytyy oletuksena Bluetooth- ja BLE-kirjastot, kun asentaa ESP32-levyhallinnan tietokoneelle. [18.]

3.3.3 LoRa-modulaatio ja LoRaWAN-verkkoprotokolla

LoRa (Long Range) on pienitehoinen, pitkän kantaman ja langaton modulaatiotekniikka, joka on implementoitu radiopiirille. LoRa-radiopiiriin on kehittänyt Semtech yhtiö. Radiopiiri mahdollistaa laitteiden yhdistämisen pilveen ja lähettämään ja vastaanottamaan dataa. Tämä tekniikka toimii melkein samalla tavalla kuin Wi-Fi mutta vaatii vähemmän virtaa ja toimii pitemmällä kantamalla. LoRa modulaatiota on jo hyödynnetty monessa IoT-projekteissa. [34.]

LoRaWAN on verkkoprotokolla, joka on rakennettu LoRa-radiomodulaatiotekniikan päälle (kuva 18). Se on suunniteltu yhdistämään laitteet ja koneet langattomasti internettiin ja hallitsemaan laitteiden ja verkkoyhdyskäytävien välistä viestintää [36]. LoRaWANin käyttö kasvaa monessa eri kaupungeissa, koska se on edullinen, siinä on pitkä kantama ja kaksisuuntainen viestintäprotokolla. Lisäksi siinä on alhainen virrankulutus, joten sitä voidaan käyttää monia vuosia pienellä akulla.



Kuva 18. LoRa-protokollapino. [35]

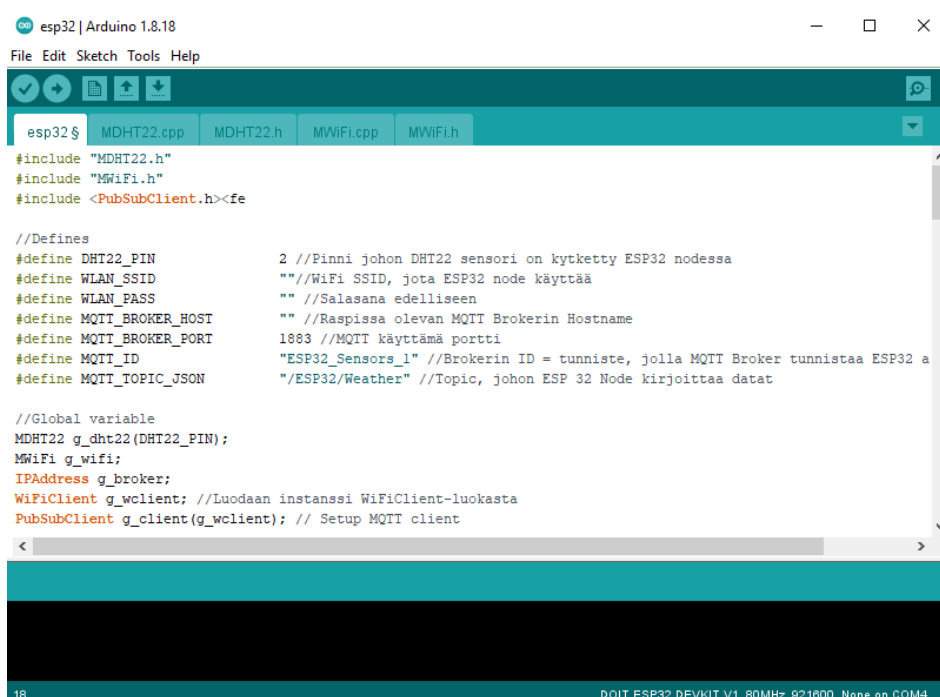
Laitteella voidaan LoRaWANin avulla ottaa yhteyttä verkkoon kahdella tavalla: OTAA-tilassa (Over the air Activation) laite määrittää verkkoavaimen ja sovelluksen istuntoavaimen muodostaakseen yhteyden verkkoon ja ABP-tilassa (Activation by Personalization) laitteeseen on määritetty jo valmiiksi verkkoavaimia, joiden avulla voi kommunikoida verkon kanssa. Tämä tila on vähemmän turvallinen, mutta helpompi käyttää kuin OTAA-tila. [37.]

3.4 Työkalut

Opetuskorttiin suunnittelussa ja toteutuksessa tarvitaan erilaisia työkaluja. Opetuskortin suunnittelussa tarvitaan piirilevysuunnittelu-ohjelma, jolla voidaan suunnitella komponenttien paikkoja piirilevyssä ja tarkistaa kytkennät oikein. Tämän lisäksi tarvitaan ohjelmaeditori, jolla voidaan ajaa ohjelmistoja opetuskortin sisään. Seuraavissa kappaleissa tutustutaan erilaisiin työkaluihin.

3.4.1 Arduino IDE

Arduino Software IDE on avoimen lähdekoodin ohjelmisto (kuva 19), jonka avulla pystyy kirjoittamaan, kääntämään ja lataamaan koodia Arduino-laitteiston sisälle. Ohjelmistoa käytetään pääsääntöisesti Arduino-moduuleihin, mutta tämä soveltuu myös muihin laitteistoihin, kun asentaa laitteistoa varten tarvittavia kirjastoja IDE-ohjelmaan. Ohjelmistoa on helppo käyttää ja sopii hyvin uusille aloittelijoille. Ohjelmisto löytyy paljon esimerkkiprojekteja, joita pystyy testaamaan Arduino-ympäristössä. Arduino IDE tukee sekä C- että C++ -kieliä. Arduino IDE on saatavilla kaikille käyttöjärjestelmille, kuten MAC, Windows ja Linux.



```

esp32 | Arduino 1.8.18
File Edit Sketch Tools Help
esp32 § MDHT22.cpp MDHT22.h MWIFI.cpp MWIFI.h
#include "MDHT22.h"
#include "MwIFI.h"
#include <PubSubClient.h><fe

//Defines
#define DHT22_PIN          2 //Pinni johon DHT22 sensori on kytketty ESP32 nodessa
#define WLAN_SSID         "" //WiFi SSID, jota ESP32 node käyttää
#define WLAN_PASS         "" //Salasana edelliseen
#define MQTT_BROKER_HOST  "" //Raspissa olevan MQTT Brokerin Hostname
#define MQTT_BROKER_PORT  1883 //MQTT käyttämä portti
#define MQTT_ID           "ESP32_Sensors_1" //Brokerin ID = tunnistee, jolla MQTT Broker tunnistaa ESP32 a
#define MQTT_TOPIC_JSON   "/ESP32/Weather" //Topic, johon ESP 32 Node kirjoittaa datat

//Global variable
MDHT22 g_dht22(DHT22_PIN);
MwIFI g_wifi;
IPAddress g_broker;
WiFiClient g_wclient; //Luodaan instanssi WiFiClient-luokasta
PubSubClient g_client(g_wclient); // Setup MQTT client

18 DOIT ESP32 DEVKIT V1, 80MHz, 921600, None on COM4

```

Kuva 19. Arduino IDE -ohjelma.

Arduino IDE-ohjelmistolla luodut ohjelmat kutsutaan luonnoksiksi (sketches). Nämä kirjoitetaan tekstieditorissa ja tallennetaan ino-tiedostomuotoon. Editorissa pystytään tarkistamaan koodia, kääntämällä ne alustalle ja tarkistamalla, onko koodissa virheitä. Koodi ei käänny alustalle, jos koodista löytyy virheitä. Kun käyttäjä kirjoittaa ja kääntää koodin, IDE luo koodille hex-tiedoston ja lähettää sen alustalle USB-kaapelilla. Alustassa mikrokontrolleri vastaanottaa hex-tiedoston ja toimii kirjoitetun koodin mukaan. [38.]

Arduino IDE koostuu monesta eri osasta. Käydään ohjelman ulkoasu läpi ylhäältä alaspäin:

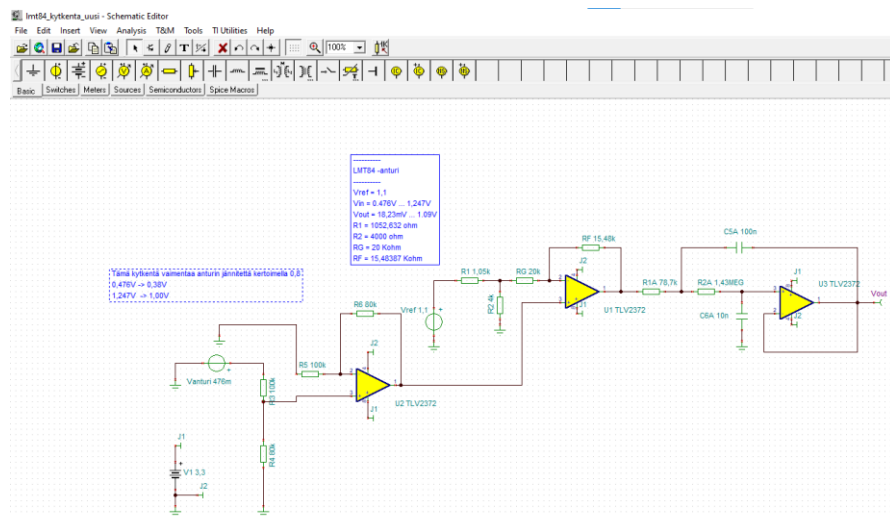
- Ikkunapalkki, sisältää tiedostonnimen ja Arduino IDE sovellusversion
- Valikkopalkki, sisältää file, edit, sketch, tools ja help. File valikosta pystyy luomaan uuden, avaamaan ja tallentamaan ino-kooditiedoston. Sieltä löytyy myös esimerkkiprojekteja monesta eri aiheista, joista voi ottaa mallia omaan projektiin. Edit valikosta löytyy erilaisia tekstityökaluja. Sketch valikosta pystytään tarkistaa ja kääntämään koodin alustalle ja lisäämään kirjastoja. Tools löytyy serial monitor, josta voidaan lukea mitä koodissa tapahtuu ja tulostuu, voidaan asettaa mitä alustaa käytetään ja mihin porttiin se on yhdistetty.
- Valikkopalkin alta löytyy kuusi nappia (Vasemmalta oikeelle).
 - Verify, tarkistaa ohjelman, jos siinä on virheitä.
 - Upload, kääntää koodin alustalle.
 - New, luo uuden ino-tiedoston.
 - Open, avaa olemassa olevan projektin.
 - Save as, tallentaa nykyisen projektin.
 - Serial Monitor, näyttää visuaalisen tiedonsiirron lähettämällä ja vastaanottamalla dataa. Tämä on todella hyvä virheenkorjaus työkalu, kun halutaan tarkastella mitä dataa koodi lähettää ja vastaanottaa ja saa paremman käsityksen ohjelman toiminnasta. Monitor sisältä löytyy tiedonsiirtonopeusvalikko, josta pitää valita oikea siirtonopeus riippuen mitä alustaa käyttää. Esimerkiksi Arduino Uno siirtonopeus on 9600.
- Tekstieditori

- Tulostuspaneeli, tulostaa tapahtumia koodista, jos koodi on käännetty onnistuneesti tai kääntäminen epäonnistui. Jos koodin kääntämisessä tapahtui virhe, tulostuspaneeli kertoo tarkasti mistä virhe johtui ja missä kohtaa koodissa virhe tapahtui. Lopuksi se myös kertoo, paljonko tilaa laudasta on viety. [39]

3.4.2 Piirilevysuunnittelu-ohjelmia

Piirilevyn suunnittelussa kannattaa hyödyntää erilaisia piirilevysuunnittelu- ja simulointiohjelmia. Niillä pystytään simuloimaan digi- ja analogikytkentöjä, samalla testaamaan nopeasti erilaisia komponentteja, mitä arvoja ne antavat ulos ja ennakoimaan mahdolliset kytkentävirheet.

Tina-TI on piirisimulointiohjelma (kuva 20). Sen ovat suunnitelleet ja julkaisseet DesignSoft ja Texas Instruments. Tina-TI on ladattavissa heidän verkkosivustollansa ilmaisena versiona, mutta sen voi päivittää maksulliseen versioon, joka sisältää enemmän ominaisuuksia ja kirjastoja. [40.]

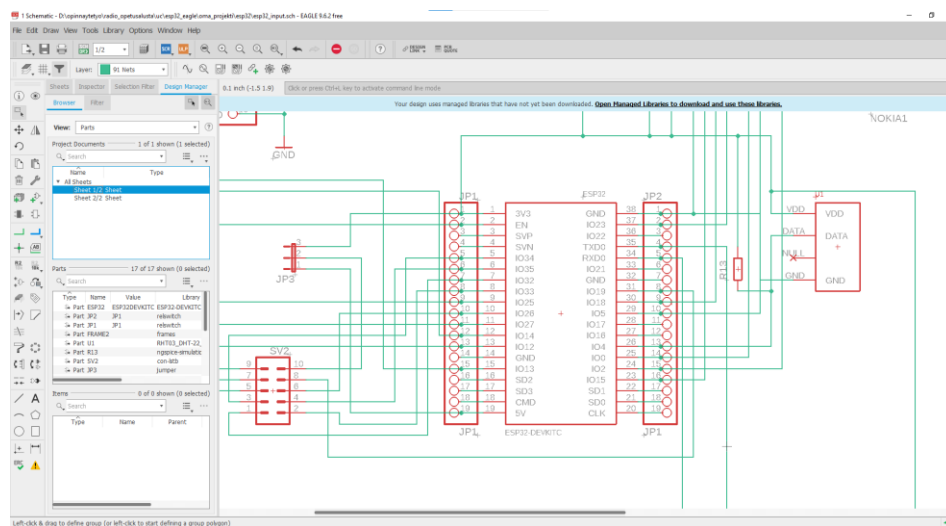


Kuva 20. Tina-TI -ohjelma.

Tina-TI tarjoaa paljon ominaisuuksia analogisen piirin suunnitteluun. Sillä pystyy simuloimaan taavirtaa, transienttia, taajuusalueita ja paljon muuta. Virtuaalisten instrumenttien avulla voi valita tuloaaltomuotoja ja mittauspiirin solmujen jännitteitä ja aaltomuotoja. Ohjelma on todella hyvä työkalu suunnittelijoille ja insinööreille, kun halutaan suunnitella, kehittää ja testata erilaisia piirejä. Tina-TI on myös helppo käyttää ja sopii hyvin uusille aloittelijoille. [41.] Internetistä löytyy myös paljon ohjeita ja esimerkkejä kytkennöistä.

Eagle on piirilevy-suunnittelu-ohjelma (kuva 21). Sen ovat julkaisseet Autodesk vuonna 2016. Eagle-ohjelman pystyy asentamaan mille tahansa käyttöjärjestelmälle; Windows, Mac ja Linux. Sen asennus ei vie paljon tilaa, sitä voi käyttää ilmaiseksi ja se tarjoaa paljon apuohjelmia piirilevyn suunnitteluun. Eaglestä on tullut yksi suosituimmista työkaluista ja monet yhtiöt käyttävät ja suosittelevat sitä. [42.]

Ohjelmasta löytyy kaksi erilaista suunnitteluikkunaa; piirikaaviosuunnittelu-, (Schematic) ja piirilevy-suunnittelu-ikkuna (Board). Piirikaaviosuunnittelussa suunnitellaan komponenttien kytkennät ja piirilevy-suunnittelussa suunnitellaan piirilevyä, johon sijoitetaan kaikki komponentit ja vedetään niiden kytkennät. Eaglestä löytyy oma komponenttikirjasto, josta saa haettua laajasti erilaisia komponentteja. On myös mahdollista suunnitella ja piirtää uusi komponentti, jos haluttua komponenttia ei löydy kirjastosta.



Kuva 21. Eagle-ohjelma.

4 Opetuskortin suunnittelu ja toteutus

Projektina oli suunnitella ja toteuttaa Kajaanin ammattikorkeakoululle opetuskortti opetuskäyttöön uusille opiskelijoille. Tätä aloitettiin toteuttamaan tammikuussa 2022 ohjaajan, asiakkaiden ja toisen ammattikorkean opiskelijan kanssa. Projektityö jaettiin kahteen osaan opiskelijan kanssa; Yksi opiskelija toteuttaa opetuskortin, johon tulee mikrokontrolleri ja radiomoduuli ja toinen opiskelija tutustuu ja ohjelmoi radiomoduulin toiminnallisuuden opetuskorttiin.

Kaikki suunnitelmat, koodit ja kaaviot jaettiin Gitlab-sivustolla. Se on tuttu ja helppokäyttöinen pilvipalvelu, joka tarjoaa Git-versiohallintaa, tehtävienhallintaa ja wikin käyttöä. Sivustolta pystyy ottamaan kopion työprojektista omalle tietokoneelle ja päivittämään Gitlabiin uutta materiaali muutamalla komennolla. Projektin käyttöoikeudet annettiin pelkästään ohjaajalle, toimeksiantajille ja opiskelijoille, joten ulkopuoliset eivät pääse siihen käsiksi.

Työn alussa piti listata vaatimuksia, mitä opetuskorttiin halutaan ja mitä erilaisia komponentteja voidaan käyttää. Sen jälkeen suunnitella kytkennät eri komponenttien välillä ja testata niiden toimivuutta. Testaukset suoritetaan kytkentäalustalla, johon kytketään mikrokontrolleri ja komponentit. Tämän jälkeen suunnitellaan opetuskortin piirikaavio Eagle-ohjelmalla ja jyrkitään koulun koneella piirilevy.

4.1 Vaatimukset

Vaatimuksena on, että opetuskortilla pystyisi ottamaan yhteyttä radioyhteydellä. Opetuskorttiin tulee käyttöön LoRaWAN-radiomoduuli, jonka kautta lähetetään ja vastaanotetaan dataa verkosta. LoRaWAN liitetään opetuskorttiin liittimellä kiinni, eli sitä ei juoteta opetuskorttiin kiinni. Syy tähän on, että radiomoduulin pystyisi myöhemmin vaihtamaan uuteen, jos vanha moduuli hajoaa.

Opetuskorttiin tulee mikrokontrolleri, jonka kautta kaikki logiikka toimii ja se on tärkein osa koko projektia. Suunnittelussa pitää tarkkaan valita oikea mikrokontrolleri ja ottaa selvää, mitä eri mikrokontrollerit sisältävät. Alkuvaatimuksena oli, että otetaan käyttöön 8-bittinen mikrokontrolleri, mutta myöhemmin päätettiin ottaa käyttöön 32-bittinen mikrokontrolleri, jotta LoRa-radiomoduulin kanssa ei tulisi ongelmia toteutusvaiheessa.

Asiakas halua opetuskorttiin ledejä ja kytkimiä, jotka olisivat kiinteänä kortilla, eli niitä ei tulla suoraan liittämään mikrokontrollerin portteihin. Mikrokontrollerista otetaan erikseen GPIO-liityntöjä auki, jonka kautta voidaan esimerkiksi laattakaapelilla yhdistää ledit ja kytkimet. Vaatimuksena olisi, että olisi vähintään 1 liitin, 8 bitin kokonaisuudessa ja käyttöjännite olisi 5 V tai 3,3 V, jonka pystyisi valitsemaan jumpperilla.

Opetuskorttiin halutaan ohjelmoinnin rajapinnan, eli debuggaus-mahdollisuuden, joko JTAG- tai UPDI-liitynnällä. Ohjelmointiliityntä riippuu valitusta mikrokontrollerista, joten sekin pitää ottaa huomioon, kun valitaan mikrokontrolleri.

Virta opetuskorttiin tulisi ensisijaisesti ESP32:n USB-johdon kautta. Sen kautta ajetaan myös ohjelmia opetuskortin sisään. Suunnittelussa on otettu huomioon myös, että opetuskorttiin saisi virran myös ulkoisen virtalähteen kautta, mutta se ei ole tässä projektissa niin vaadittu osa.

Lopuksi opetuskorttiin halutaan LCD-näyttö ja kiinteä anturi, jonka kautta voidaan lukea joitakin arvoja ja tulostaa ne LCD-näyttöön. Näyttö on tärkeä osa opetuskortissa, josta voidaan lukea reaaliaikaisesti dataa, joko anturista tai verkosta tulevaa dataa.

4.1.1 Komponenttien valinta

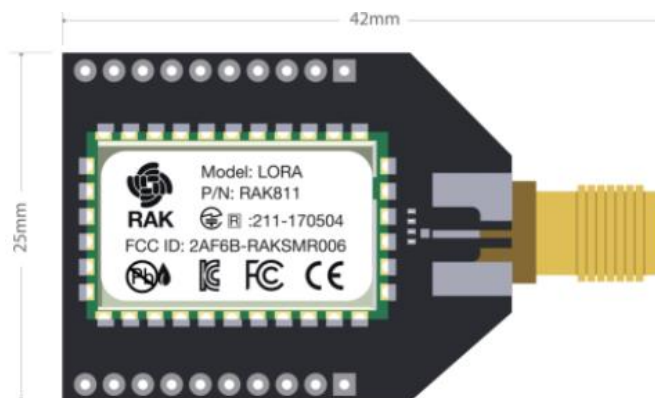
Ensimmäisenä komponenteista piti valita oikea mikrokontrolleri. Mikrokontrolleria valittaessa pitää ottaa huomioon, että siinä on tarpeeksi Flash- ja SRAM-muistia, IO-portteja, oikea debuggeri ja erilaisia tiedonsiirtoja. Tärkeintä olisi, että se olisi yhteensopiva LoRaWAN-radiomoduulin kanssa.

Aloin Microchip-sivustolta tutkimaan 8-bittisiä mikrokontrollereja ja listaamaan, mitä eri mikrokontrollereista löytyy. Sivustolta löytyi AVR 8-Bit pdf -tiedosto, josta löytyi lista eri AVR-mikrokontrollereista (kuva 22).

ESP32 WROOM 32D -moduuli (kuva 7). Se tulee opetuskortin päälle liittimellä, samalla idealla kuin LoRaWAN-radiomoduuli.

Loin uuden Excel-taulukon ESP32-moduulille, johon listasin kaikki portit ja ominaisuudet, mitä moduulista löytyy. Samaan taulukkoon listasin kaikki muut komponentit, jotka tulevat opetuskorttiin.

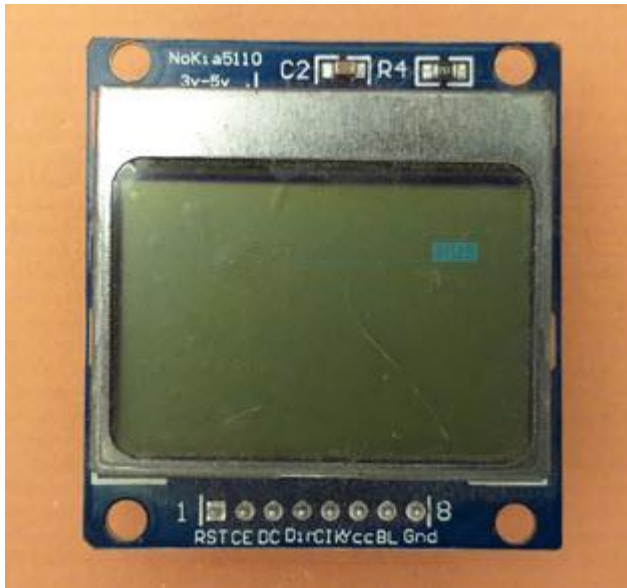
Seuraavaksi valittiin LoRaWAN-radiomoduuli. Pitkän tutkimuksen jälkeen päätettiin ottaa käyttöön RAK811 LoRaWAN -radiomoduulin (kuva 23) [44]. Se on pieni, toimii UART-tiedonsiirrolla ja siihen löytyy netistä hyviä esimerkkejä ESP32-moduulille, joita voi hyödyntää ohjelmoinnissa. Jotta radiomoduulia voi käyttää yhteydenotossa, radio pitää erikseen rekisteröidä radion omilla osoitteilla. Osoitteita on yhteensä 3 kappaletta; DevEUI, AppEUI ja AppKey. DevEUI on radiomoduulin oma henkilökohtainen 64-bittinen tunnus. Radiomoduulin oma tunnus löytyy moduulin päältä. AppEUI on laajennettu 64-bittinen tunnus, jota käytetään palvelimen tunnistamiseen aktivoinnin aikana. AppKey on laitteen salausavain, jota käytetään viesteissä jokaisen aktivoinnin aikana. Salausavain rakentuu DevEUI- ja AppEUI-tunnuksista [45].



Kuva 23. RAK811 LoRaWAN -radiomoduuli. [44]

Opetuskorttiin halutaan anturi, jolla voi mitata erilaisia arvoja huoneista. Päätin valita DHT22-anturin (kuva 14), jolla voi mitata lämpötilaa ja kosteutta. Se toimii 1-Wire tiedonsiirrolla ja vaatii 3,3 V. Se on helppo asentaa ESP32:een ja siihen löytyy valmiiksi kirjastoja netistä, jotka pitää asentaa ohjelmistoon.

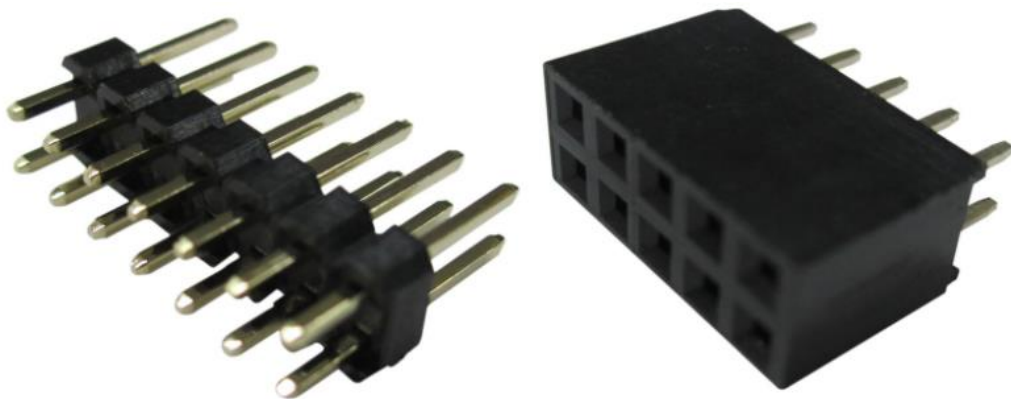
Seuraavaksi piti valita näyttö opetuskorttiin. Päätettiin, että otetaan käyttöön Nokia 5110 LCD-näyttö (kuva 24). Se on 84x48-pikselinen näyttö, johon löytyy myös netistä valmiit kirjastot ja kytkennät. Nokia näyttö käyttää SPI-tiedonsiirtoa, joten se tarvitsee vähän enemmän IO-portteja ESP32:lta [46].



Kuva 24. Nokia 5110 LCD-näyttö. [46]

Lopuksi opetuskorttiin laitetaan 4 lediä ja 4 kytkintä, joita voi laattakaapelilla yhdistää ESP32:een ja ohjelmoida sen kautta. Ledit ja kytkimet tulee pintaliitoksena opetuskorttiin. Suunnitelman aikana huomattiin, että ESP32 jää ylimääräisiä GPIO-portteja, joten ne kammataan auki opetuskorttiin.

ESP32:lle, DHT22, LoRa-radiomodulille ja Nokia-näytölle suunniteltiin piirilevyille pinniliittimet, joihin komponentit voidaan liittää, eli niitä ei juoteta suoraan piirilevyyn kiinni. Komponentteja voi myöhemmin vaihtaa uuteen, jos komponentti hajoaa. Kaikille komponenteille, paitsi radiolle, suunniteltiin 2,54 mm naaraspinniliittimet (kuva 25). Radiolle tulee 2,00 mm naaraspinniliitin, koska siinä on pienemmät pinnit kuin muissa komponenteissa. JTAG, jumbpereille ja GPIO-liittimille suunniteltiin urospinniliittimet (kuva 25).



Kuva 25. Uros- ja naaraspinniliitin [47] [48]

4.1.2 Komponenttien testaus

Valittuja komponentteja piti testata kuinka hyvin ne toimivat ESP32 kanssa. Alkuvaiheessa on hyvä testata kaikkia komponentteja, että ne varmasti toimivat ja että niiden kanssa ei tule jatkossa ongelmia piirilevyn toteuttamisen kanssa. Samalla pystyy kartoittamaan kuinka komponentit voisi sijoittaa piirilevyn, että siitä tulee järkevän näköinen.

Kaikkia komponentit kytkettiin kytkentäalustalle ja hyppylangoilla. Testauksessa käytettiin erilaisia ledejä ja kytkimiä, koska pintaliitoksisia komponentteja ei saa suoraan yhdistettyä kytkentäalustalle. Ledeinä käytettiin 8-bittistä Led Bar-lautaa (kuva 26). Laudasta löytyy 8 lediä, mutta koska opetusalustaan tulee käyttöön 4 lediä, laudasta otetaan testiin 4 lediä. Kytkiminä käytettiin yksinkertaisia nappeja, jotka sai helposti kytkentäalustaan kytkettyä (kuva 26). Napit ja ledit tarvitsi 8 IO-portteja, jotka varataan heille ESP32:sta. Valmiissa opetuskortissa niille tehdään oma liitin, jonka voi liittää ESP32:een.

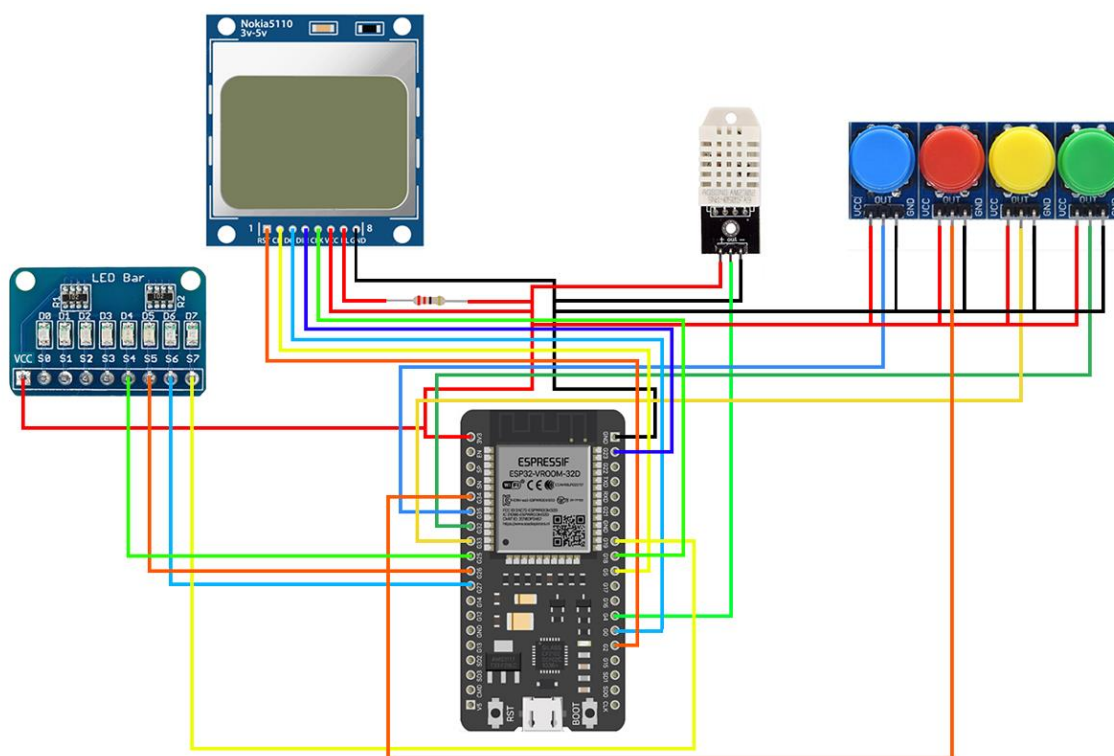


Kuva 26. Led Bar-lauta ja kytkimiä. [49] [50]

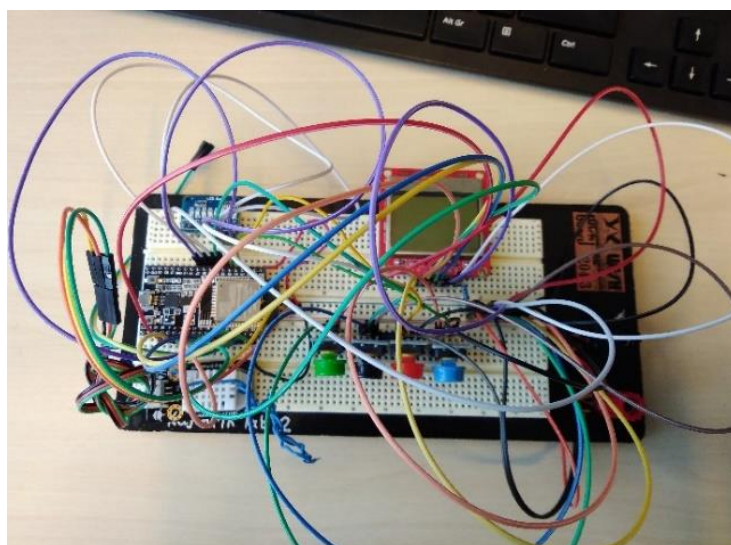
DHT22-anturi käyttää 1-Wire tiedonsiirtoa, eli se tarvitsee pelkästään maan, 3,3 V:n jännitteen ja yhden dataportin. Sille varataan ESP32:sta IO4-portti.

Nokia 5110 LCD-näyttö käyttää SPI-tiedonsiirtoa, joten se sille varataan monta porttia ESP32:lta. Se tarvitsee resetin, sirun, datan, serial datan, kellon, 3,3 V:n käyttöjännitteen, taustavalon ja maadoituksen. ESP32:sta annetaan portit IO2, -0, -5, -18 ja -19. Näytön taustavalo kytketään 220 ohmin vastuksella 3,3 V:n jännitteeseen.

Valmiista opetuskortista löytyy lisäksi JTAG ja GPIO liittimiä, mutta niitä ei tässä testausvaiheessa oteta huomioon. Tässä testauksessa on tarkoitus testata komponentteja ja niiden toimivuutta yhdessä. Kuva 27 näyttää kytkentäkuva, joka toteutettiin kytkentäalustalle (kuva 28).



Kuva 27. Kytentäkuva. [49] [50] [51] [52] [53]

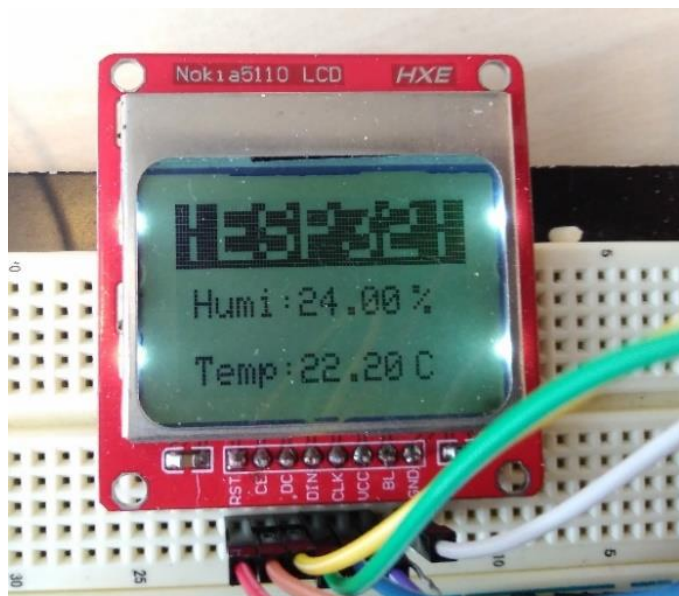


Kuva 28. Kytentäalusta.

Kytkemisen jälkeen ESP32:een piti ohjelmoida yksinkertainen ohjelma, jolla pystyisi testaamaan kaikkia komponentteja yhdessä. Suunnittelin ESP32:een ohjelman, joka tulostaa Nokian näytölle DHT22-arvoja (kuva 29), jotka päivittyvät aina sen mukaan, kun painaa nappia. Samalla napeilla pystyisi sytyttämään ledin, kun pitää nappia pohjassa.

Ohjelman alussa luetaan tarvittavia kirjastoja ja luokitellaan DHT22 -anturille, Nokian näytölle, ledeille ja napeille pinniportit. Kirjastot saa ladattua joko internetistä tai Arduino IDE-ohjelman Sketch -valikosta. Kaikki komponentit alustetaan, asetetaan tarvittavia oletusarvoja ja käynnistetään näyttö ja DHT22-anturi. Nappien tila asetetaan nollassi ja sisääntuloksi. Ledit asetetaan ulostuloksi, jotka palavat sen mukaan, kun nappien tila on painettuna alas.

Nappeja on yhteensä 4 ja ledejäkin on myös 4, joten kaikille on oma parinsa. Jokaisella parilla on oma toimintansa. Kaikille napeille on samanlainen toimintatapa, mutta ensimmäiseen nappiin on lisätty enemmän toimintoja. Kun painaa ensimmäistä nappia, ensimmäinen ledi syttyy ja samalla DHT22-anturi lukee uuden arvon, joka tulostetaan Nokian näytölle. DHT22-anturille ja näytölle on kirjoitettu oma funktio, jota kutsutaan, kun painetaan nappia. Samaa funktioon on kirjoitettu myös tarkistus: jos DHT22-anturilta ei saada luettua arvoa, tulostetaan virheviesti näytölle. Jokaisen DHT22-arvon lukemisen jälkeen on lisätty 2 sekunnin viive.



Kuva 29. Nokia 5110 LCD-näyttö.

LoRaWAN-radiomodulille tehtiin myöhemmin erikseen omat testaukset. Sitä ei saatu kytkettyä suoraan kytkentäalustaan, koska siinä on pienemmät pinnijalat. Modulille juotettiin hyppylangat kiinni, joiden avulla se saatiin kytkettyä ESP32:een. Radiomoduuli käyttää UART tiedonsiirtoa, joten sille varattiin ESP32:sta pinnit IO16 ja IO17. IO16-pinni vedettiin radion TX-pinniin ja IO17 vedettiin RX-pinniin. Moduuli tarvitsee lisäksi 3,3 V:n käyttöjännitteen ja maadoituksen. Testauksessa toteutettiin yksinkertainen testaus, jossa otettiin yhteyttä verkkoasemaan ja palautettiin sieltä OK viesti, että radiomoduuli on saanut yhteyden verkkoasemaan. Radiomodulista annettiin DevEUI-avain (kuva 30) verkkoasemalle, jotta se osasi tunnistaa oikean radiomodulin.



Kuva 30. LoRaWAN RAK811 -moduuli.

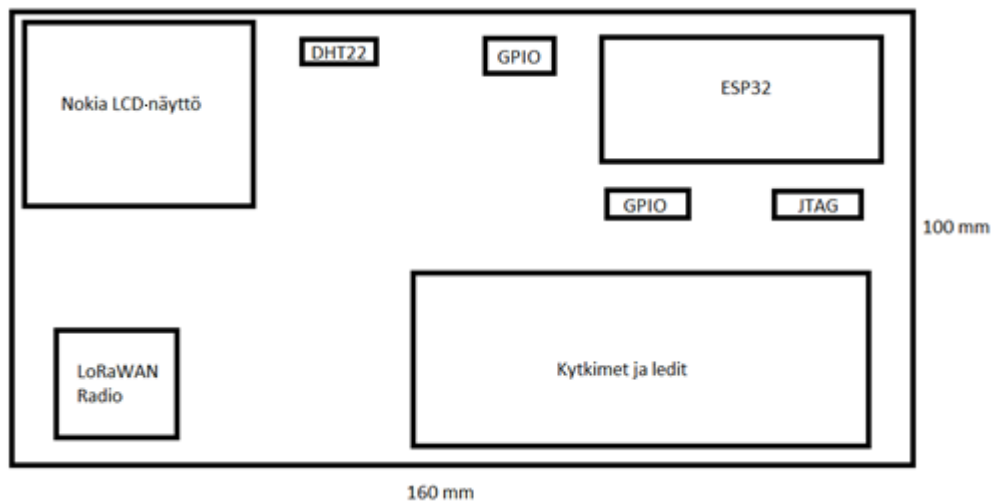
Ohjelman testaamisen aikana kaikki komponentit toimivat hyvin yhdessä ja ei tullut mitään suurempia ongelmia. Samaa ohjelmaa joutuu myöhemmin testaamaan uudestaan, kun saadaan valmis opetuskortti tehtyä.

4.2 Piirilevyn suunnittelu

Opetuskortille piti suunnitella piirilevy, johon tulevat kaikki komponentit ja niiden kytkennät. Piirilevyn suunnittelu on tarkkaa hommaa ja voi vaatia paljon aivotyöskentelyä. Kytkenät joutuvat vetämään erittäin tarkasti, ettei tule mitään oikosulkuja sekä komponenttien paikkoja pitää harkita tarkkaan, kuinka ne laitetaan. Samalla pitää ottaa huomioon, minkä kokoisen piirilevyn halutaan tehdä. Liian pientä piirilevyä ei kannata tehdä, koska komponentit tarvitsevat tilaa, kun niille vetää kytkennät.

Piirilevyn suunnittelun alussa mietittiin, minkä kokoisen piirilevyn halutaan tehdä. Aluksi harkittiin, että tehdään 100x100 mm kokoinen piirilevy, mutta se todettiin olevan liian pieni kaikille komponenteille. Lopulta päädyttiin tekemään 160x100 mm kokoinen piirilevyn. Sen jälkeen mietittiin komponenttien paikkoja piirilevyssä. Nokia-näyttö suunniteltiin, että se laitetaan ylävasemmalle nurkkaan, ESP32 laitetaan yläoikealle nurkkaan, ledit ja kytkimet ESP32:en alle ja LoRaWAN-

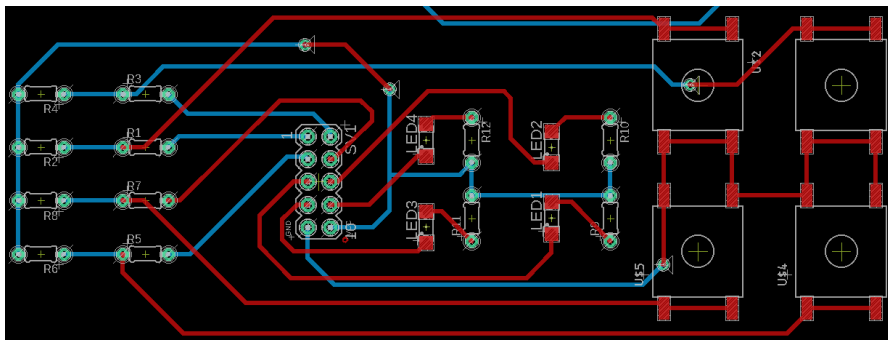
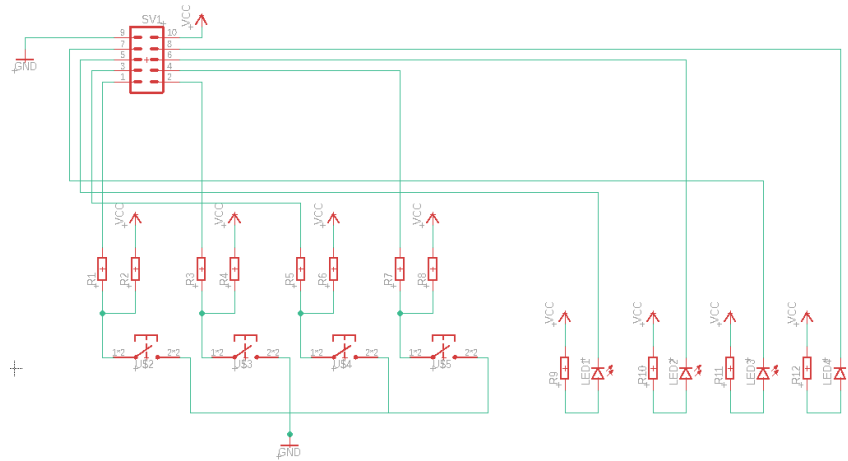
radiomoduuli laitetaan Nokia näytön alapuolelle. ESP32:en viereen tulee myös JTAG-, GPIO-liittimiä ja DHT22-anturi laitetaan piirilevyn yläpuolelle (kuva 31).



Kuva 31. Piirilevyn suunnittelupiirros.

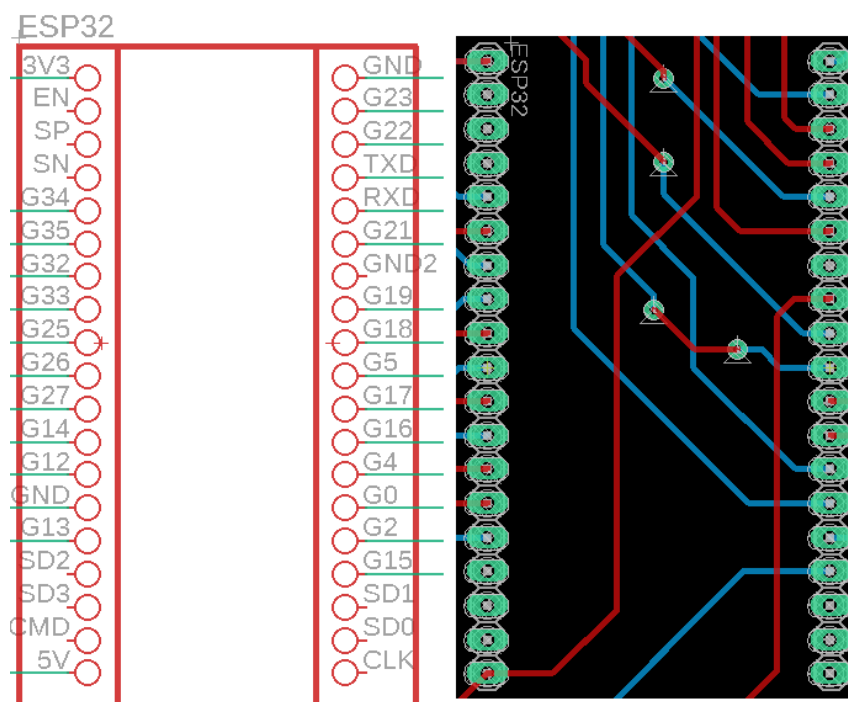
Tämän jälkeen, kun piirilevyn koko ja komponenttien paikat on päätetty, piti tämä suunnitelma toteuttaa piirilevysuunnittelu-ohjelmalla. Tässä työssä käytettiin EAGLE-ohjelmaa. Sillä saa hyvin suunniteltua piirilevyn, siitä löytyy paljon hyödyllisiä työkaluja ja sisältää paljon komponenttikirjastoja, joita voi käyttää työssä. Piirilevyn suunnittelun aikana jouduttiin käyttämään pohjana komponenttikirjastoja, joita muokattiin mieluisiksi piirilevylle.

EAGLE-ohjelmassa luotiin uusi kansio projektille, johon tulee kaikki käytetyt kirjastot, kaaviot ja lauta. Ensimmäiseksi suunniteltiin kytkimien ja ledien kytkennät piirilevylle (kuva 32). Piirilevyssä kytkimet ja ledit tulevat pintaliitoksina, joten ne joudutaan ottaa samalla huomioon piirilevysuunnittelussa. Kytkimet ja ledit liitetään omaan liittimeen, eli niitä ei liitetä suoraan ESP32:seen. Ne saadaan käyttöön liittämällä laattakaapelilla ESP32:en vapaaseen liittimeen. Kytkimet ja ledit ovat nolla-aktiivisia, eli kun kytkintä painetaan, ne asettuvat nollatilaan ja ledi palaa, kun sen lähtö on nollatilassa.



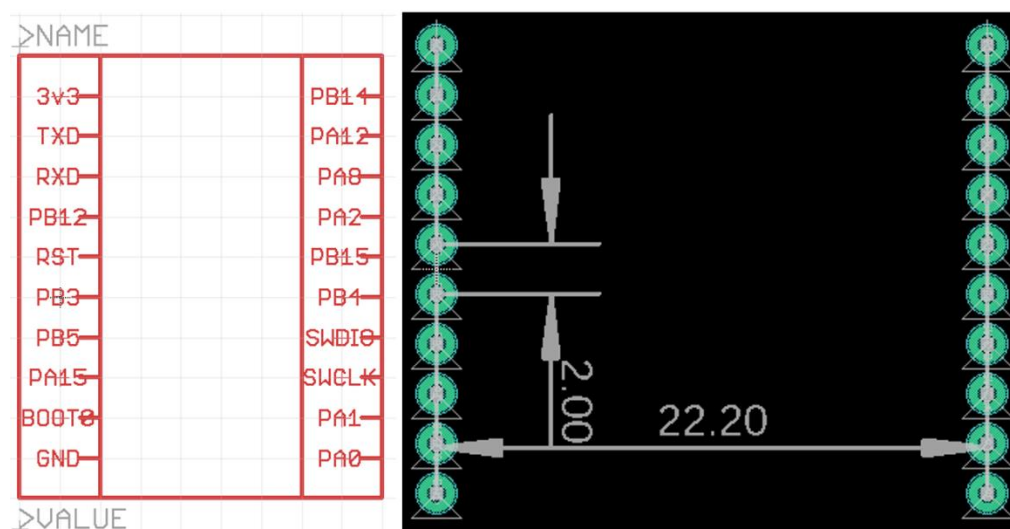
Kuva 32. Kytkimien ja ledien kytkennät.

ESP32:sta löytyi netistä monta erilaista komponenttikirjastoa, mutta jokaisessa kirjastossa ESP32:en pinnipaikat olivat väriä tai oli väärä moduuliversio. Tässä työssä käytetään ESP32 Wroom 32D -moduulia. Päätettiin piirtää uusi ESP32:en symboli ja nimeämään pinnipaikat oikeille paikoille (kuva 33). ESP32:sta ei käytetä kaikkia pinnejä; EN, SP, SN, SD0-3, CMD ja CLK jäävät kytkemättä. Piirilevy suunnitelmassa pitää ottaa huomioon pinnireikien koko ja niiden väli. Yleisesti pinnireikien väli on 2,54 mm.



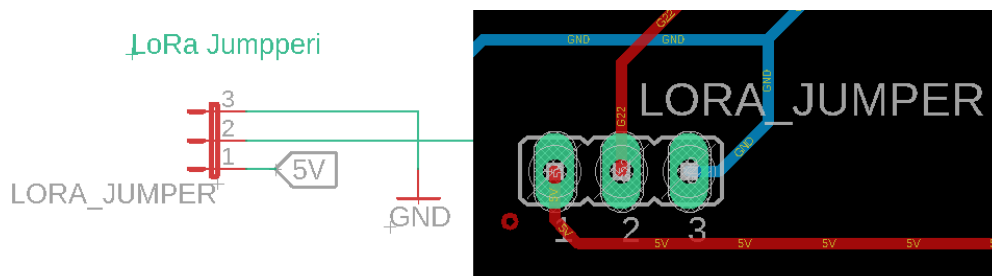
Kuva 33. ESP32.

LoRaWAN-radiomoduulista ei löytynyt komponenttikirjastoa, joten siitä jouduttiin luomaan uusi kirjasto. Radiomoduuliin ei mene tavallinen 2,54 mm liitin, vaan siihen menee 2,00 mm:n liitin. Tämä piti ottaa huomioon symbolin piirtämisessä. Radiomoduulista tehtiin yksinkertainen piirros ja siihen mitattiin tarkalleen kaikki pinnipaikat (kuva 34). Piirilevyssä pinnien porauspaikoille juotetaan liittimet, joiden päälle radio liitetään. ESP32:sta vedettiin IO16, -17, maadoitus ja 3,3 V:n kytkennät radioon.



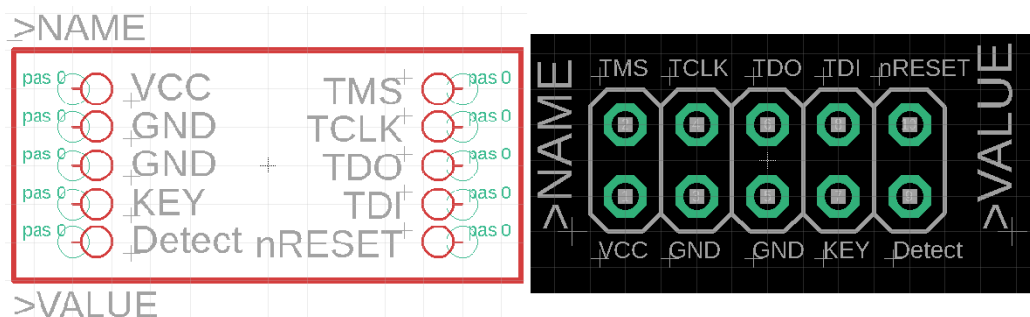
Kuva 34. LoRaWAN-radiomoduuli.

LoRaWAN-radiomoduulille suunniteltiin oma jumpperi, josta pystyy valitsemaan radiomoduulille 1 tai 0 tilan (kuva 35). Jumpperin ideana on, että radiomoduulin pystyy tarvittaessa laittamaan lepotilaan, kun sitä ei haluta käyttää. Radiomoduulin tilan pystyy ohjelmoimaan käyttäen IO22-porttia. Jumpperiin on lisäksi vedetty 5V:n jännite ja maadoitus.



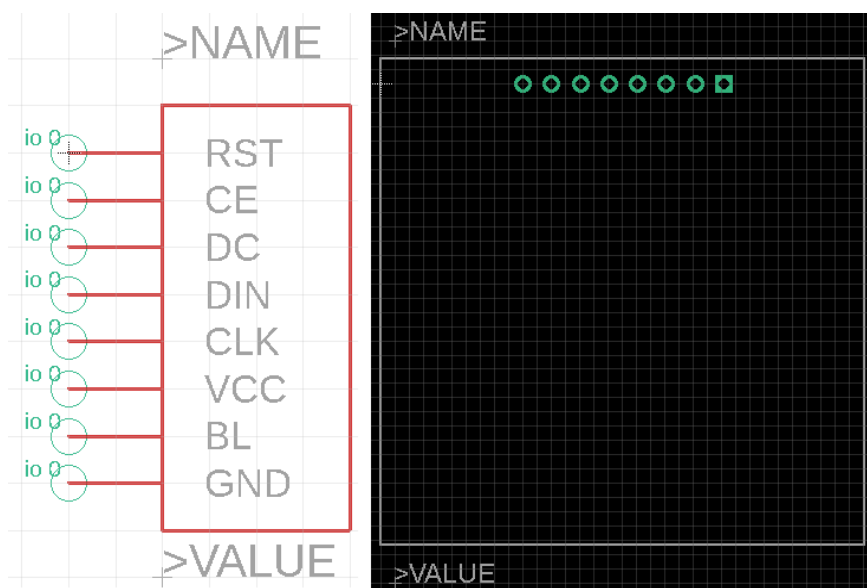
Kuva 35. LoRaWAN -jumpperi.

JTAG-liittimeen löytyi valmis komponenttikirjasto, jota pystyi käyttämään piirilevyssä (kuva 36). Liittimestä piti tarkistaa pinnien paikat, että ne ovat oikeissa paikoissa. JTAG:in VCC kytketään 3,3 V:n jännitteeseen, maadoitukset, KEY ja Detect vedetään maahan. TMS, TCLK, TDO ja TDI kytketään ESP32:n GPIO-pinneihin, joita käytetään ohjelmoinnin rajapinnassa. Reset-pinniä jätettiin kytkemättä.



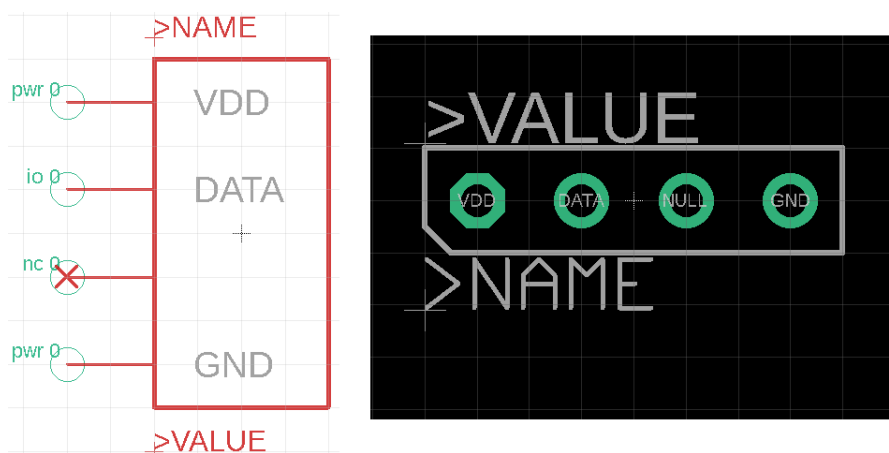
Kuva 36. JTAG-liitin.

Nokia 5110 LCD näytölle ei löytynyt komponenttikirjastoa, joten sillekin jouduttiin luomaan uusi kirjasto (kuva 37). Näytöstä piirrettiin piirilevy-suunnitelmaan näytön pituus ja leveys, paljon se tulee viemään piirilevystä. Tämä helpottaa piirilevyn toteutusta ja välttää virheitä, että näytön alle menisi vahingossa joku komponentti tai osa. Näytöstä kytkettiin kaikki pinnit ESP32:een ja BL-pinnille lisättiin vastus.



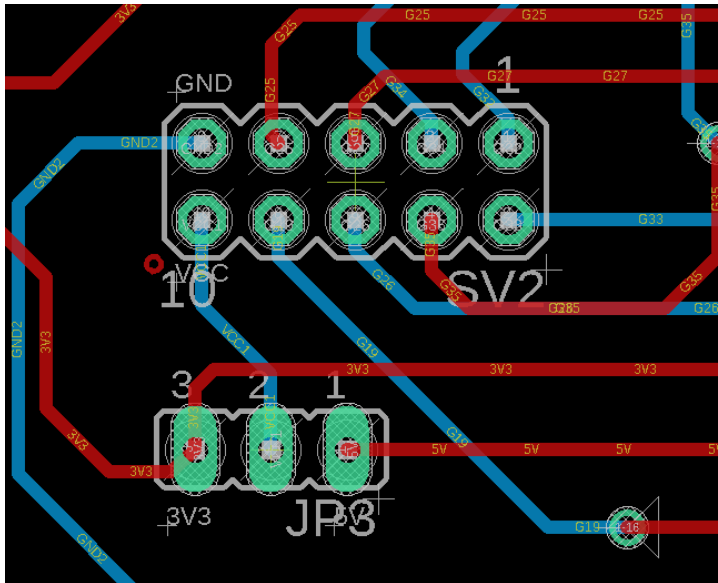
Kuva 37. Nokia 5110 LCD-näyttö.

DHT22-anturille löytyi oma komponenttikirjasto (kuva 38), joten siitä ei tarvinnut erikseen piirtää symbolia. Anturin VDD kytketään 3,3 V:n jännitteeseen, GND maahan ja Datapinni ESP32:en vaapaaseen GPIO-pinniin. Anturille lisättiin ylösvetovastus Data ja VDD väliin, jotta anturilta saada oikeanlaisia data-arvoja, kun mitataan huoneen lämpötilaa ja kosteutta.



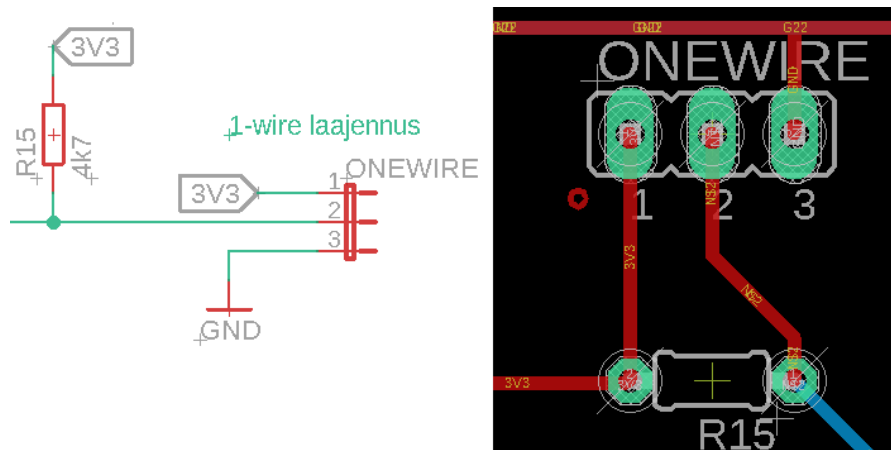
Kuva 38. DHT22-anturi.

ESP32:sta kammattiin auki 8 GPIO-porttia. Piirilevyyn lisättiin 5x2-liitin, johon tulee 8 GPIO, käyttöjännite ja maadoitus (kuva 39). Liittimelle lisättiin jumpperi, josta voi vaihtaa 3,3-5 V:n käyttöjännitteen. Liitintä on tarkoitusta käyttää kytkimien ja ledien ohjaukseen, mutta sitä voi käyttää myös muihin kytkentöihin ja testauksiin.

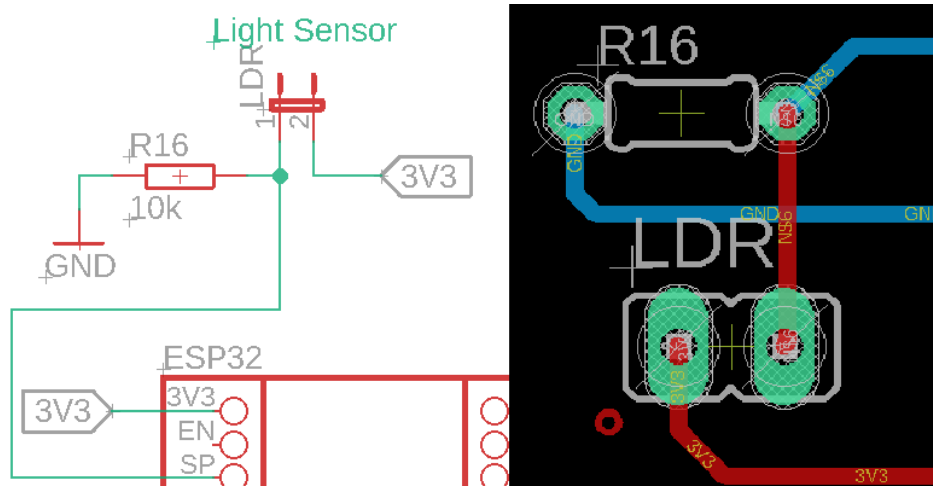


Kuva 39. GPIO-liitin ja jumpperi.

Piirilevylle suunniteltiin alussa toinenkin GPIO-liitin, mutta myöhemmässä vaiheessa päädyttiin toimeksiantajan kanssa lisätä piirilevyyn ylimääräisen 1-Wire laajennuksen ja analogisen LDR-vastuksen. 1-Wire laajennukselle tehtiin 3-pinninen liittymä, johon tulee GPIO, 3,3 V:n jännite ja maadoitus. GPIO kytkentään lisättiin myös 4,7 kilon ylös vetovastus (kuva 40). LDR-vastus tarvitsee ESP32:lta analogisen portin, joten sille varataan SP-portti. Siihen kytketään myös alasvetovastuksen ja 3,3 V:n jännite (kuva 41).

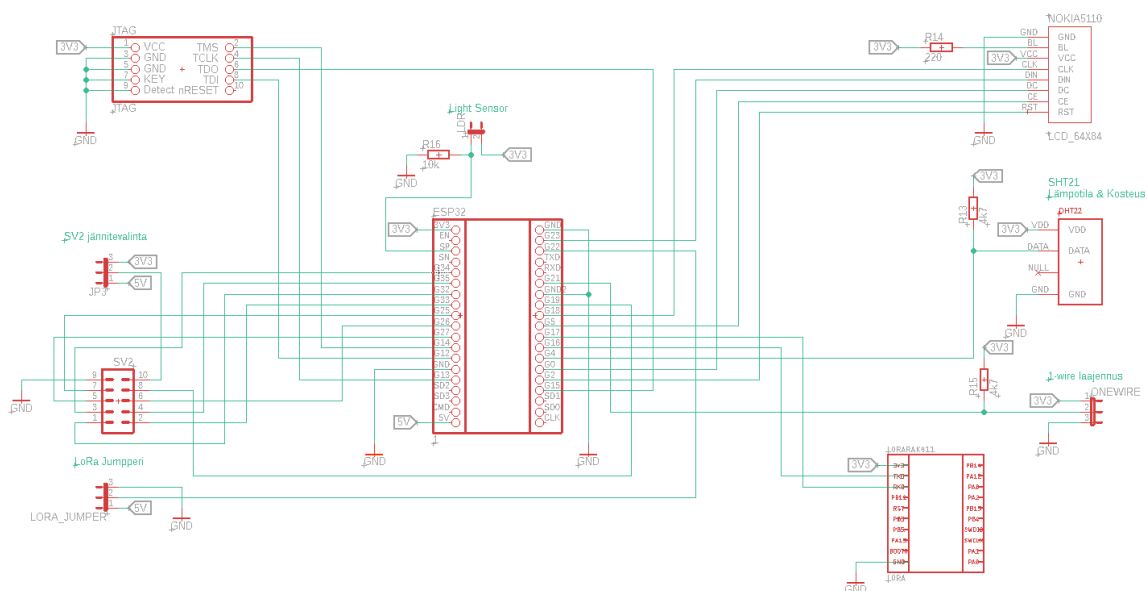


Kuva 40. 1-Wire laajennus.



Kuva 41. LDR-vastus.

Piirikaavio-ikkunassa (Kuva 42) tehtiin kaikki kytkennät. Kytkennöille tehtiin kaksi ikkunatasoa: kytkimet ja ledit kytkennät luotiin omalle ikkunatasolle ja ESP32:sen kytkennät omalle ikkunatasolle. Ne eriteltiin toisistaan, koska kytkimiä ja ledejä ei tulla kytkemään suoraan ESP32:een, vaan ne tulevat piirilevyn lisäosana ja ESP32:en ikkunaan tulee kaikki ESP32:en omat kytkennät. Tämän jälkeen, kun kaikki kytkennät oli tehty piirikaaviosuunnittelussa, siirryttiin seuraavaksi suunnittelemaan piirilevyä.



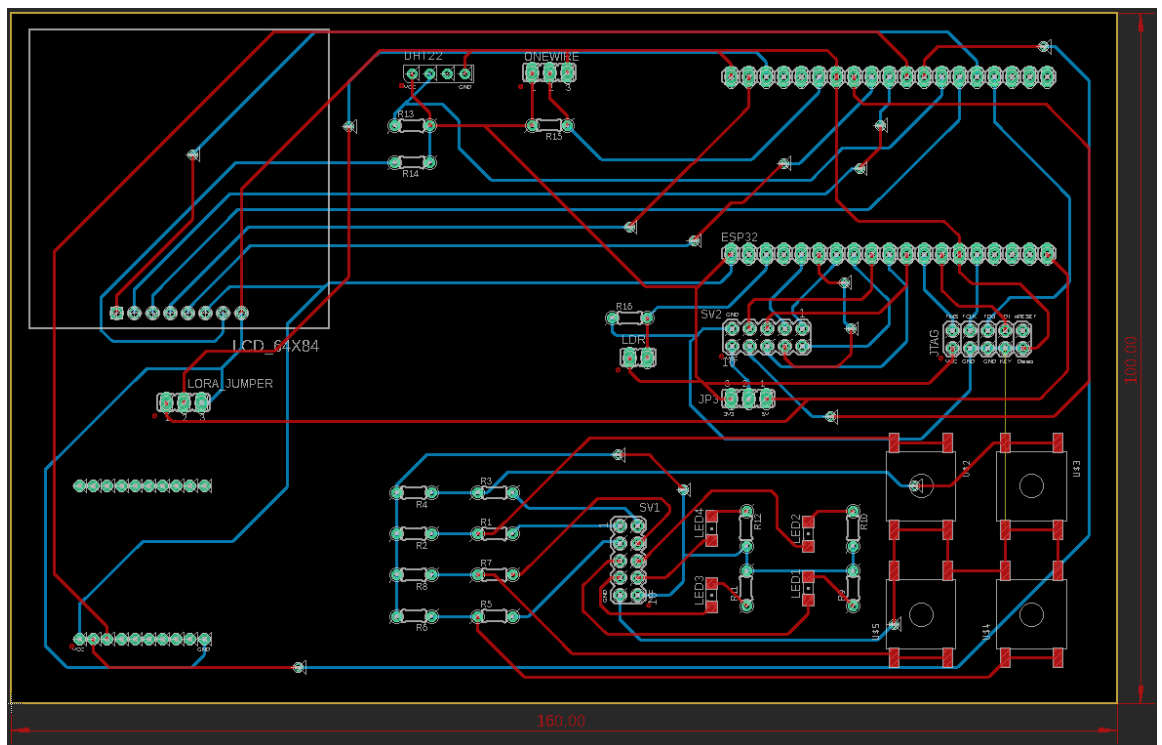
Kuva 42. ESP32:sen piirikaaviosuunnitelma.

Piirilevyn suunnittelu ja komponenttien kytkeminen on tarkkaa hommaa. Tässä pitää ottaa huomioon, kuinka paksuilla vedoilla halutaan tehdä kytkennät ja kuinka paksulla terällä halutaan tehdä piirilevyn tarvittavat poraukset. Liian ohuilla vedoilla ei kannata vetää kytkentöjä, joten

vedot tehdään 0,4 mm paksuudella ja poran teräksi valitaan 0,8 mm. Langan mutkaksi valitaan 45 astetta.

Piirilevy jysyttiin kahdelta eri puolelta, levyn yläpuolelta ja alapuolelta (kuva 43). Levyn yläpuolelle tulee kaikki komponentit. Eaglessä on kaksi eri tasoa, joissa voi vetää kytkennät: Top ja Bottom. Top-tasossa kaikki kytkennät ovat punaisia ja Bottom-tasossa sinisiä. Punaiset ja siniset kytkennät voivat koskettaa toisiaan ja mennä toisistaan läpi, koska piirilevyssä ne eivät tule aiheuttamaan oikosulkua. Kytkennässä joutui tekemään läpiporauksia, jossa yläosan kytkentä menee alapuolelle ja toisinpäin.

Eaglestä löytyy paljon työkaluja, joita voi hyödyntää komponenttien kytkemisessä. Voi joko manuaalisesti vetää kaikki kytkennät tai käyttää automaattista kytkentätyökalua nimeltä Autoroute-työkalua. Yleensä automaattinen työkalu ei tee niin hyvää jälkeä kuin saattaisi odottaa, mutta sitä käytettiin alustavasti suunnittelussa. Kytkennät tehtiin lopuksi manuaalisesti. Kytkennän suunnittelussa kannattaa ottaa huomioon, etteivät kytkennät mene liian läheltä piirilevyn reunaa tai samassa tasossa olevat kytkennät mene liian läheltä toisiaan.



Kuva 43. Piirilevysuunnitelma.

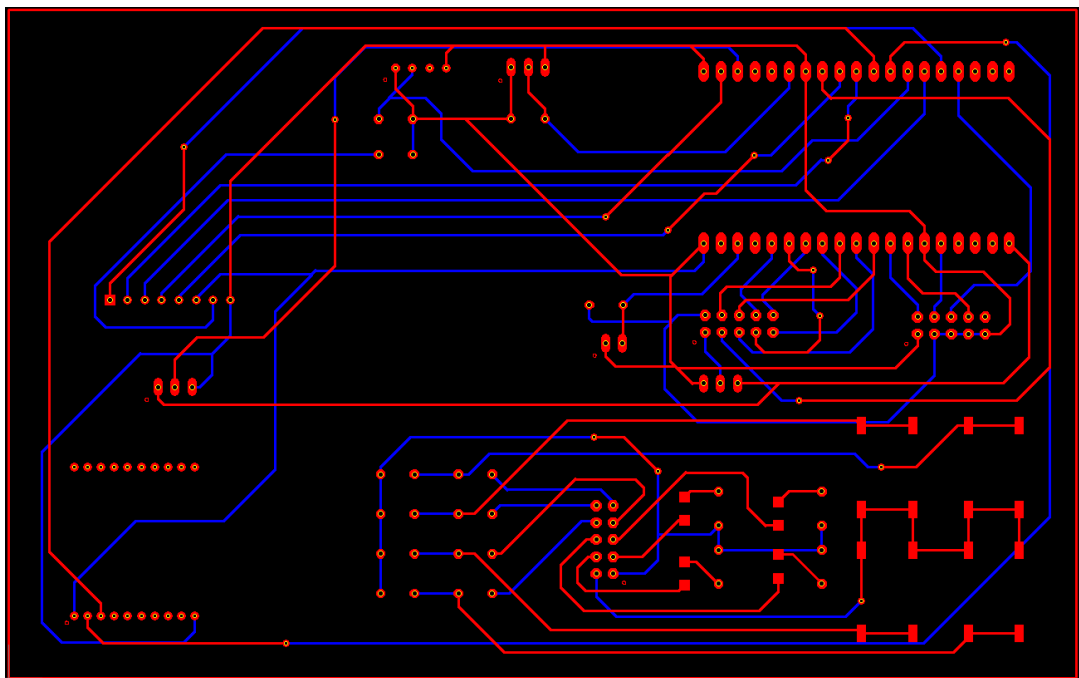
Kun piirilevyn kytkennät saatiin tehtyä, top ja bottom-taso muunnettiin CAM-tiedostomuotoon Eagle:n omalla CAM-prosessorityökalulla. CAM-tiedostoja tarvitaan piirilevyn poraukseen, josta

luetaan piirilevyn kytkennät, poraukset ja niiden paksuudet. Prosessorityökalu luo monta erilaista CAM-tiedostoa, mutta me tarvitsimme tässä työssä top- ja bottom copper-tiedostoja. Työkalu loi samalla poraustiedoston, jonka avulla saatiin piirilevyn porattua reiät. CAM-tiedostot tallennettiin muistitikulle ja vietiin tiedostot toiselle tietokoneella, jossa suoritettiin piirilevyn jyrsimisen.

4.3 Piirilevyn toteutus

Piirilevyn toteutus tehtiin koulun omilla laitteilla. Koulusta löytyy lisensoitu ISOcam-ohjelma, johon vietään tehdyt CAM-tiedostot. Siellä luotiin meille uudet jyrsimistiedostot, jotka sitten vietiin piirilevyn jyrsimislaitteelle.

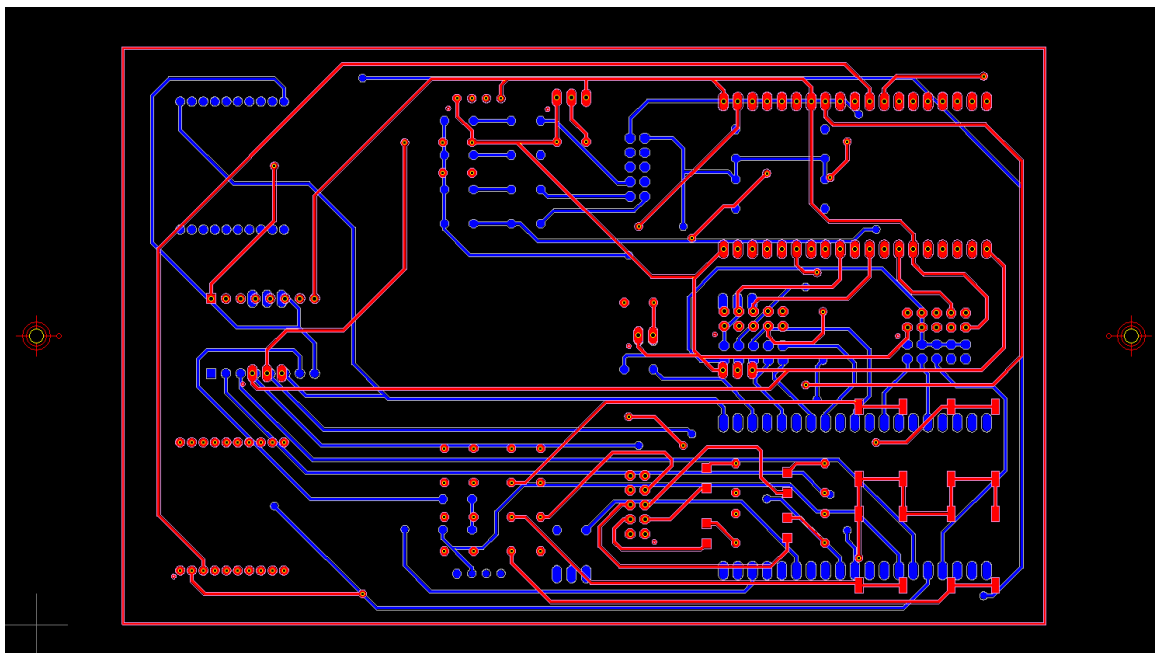
Ensimmäiseksi CAM-tiedostot piti käsitellä ja tallentaa uusiin tiedostomuotoihin, joita jyrsimislaite pystyisi lukemaan. Tiedostot vietiin ISOcam-ohjelmistoon. Jokaiset tiedostot vietiin omalle tasolle, levyn yläpuoli 0-tasolle, alapuoli 1-tasolle ja porausreiät 2-tasolle. Ohjelma kysyi, missä muodossa porausreiät luetaan, johon kuitattiin millimetreillä. Ohjelma tulosti näytölle piirilevykerrokset ja porausreiät (kuva 44).



Kuva 44. Piirilevyn kerrokset ja porausreiät.

Piirilevy pitää jyrsiä molemmilta puolilta, joten seuraavaksi piti tehdä piirilevyn kohdistusreiät. Niiden avulla pystytään tekemään jyrsimiset oikeisiin paikkoihin, kun piirilevy käännetään toisinpäin. Kohdistusreiät tehtiin piirilevyn ulkopuolelle, vasemmalle ja oikealle puolelle. Samalla peilattiin levyn alapuoli, jotta laite jyrsii piirilevyn oikein (kuva 45). Porausreiät eivät ole kohdakkain alapuolen kanssa, mutta se ei haitannut, koska reiät porattiin levyn yläpuolelta.

Seuraavaksi luotiin levyn ylä- ja alapuolesta jyrshintätiedostot. Ohjelma piirsi levyn vetojen ympärille valkoiset viivat, joihin laite jyrsii kytkennät. Ohjelma teki tarvittavat tarkistuspyynnöt, jotka kuitattiin Ok:lla. Tarkistuksien jälkeen jyrshintä- ja poraustiedostot tallennettiin ja vietiin jyrsimislaitteelle.



Kuva 45. Kohdistusreiät ja peilattu alapuoli.

Jyrsimislaite käyttää RoutePro-ohjelmaa tiedostojen lukemiseen (kuva 46). Ohjelma oli asennettu koulun vanhalle tietokoneelle, jossa on Windows 2000 käyttöjärjestelmä (kuva 47). Ohjelmasta on hyvä tarkistaa kaikki asetukset oikein muun muassa piirilevyn ja pohjalevyn paksuus. Piirilevyn paksuudeksi valittiin 1.5 mm ja pohjalevyksi 16 mm. Offset-asetuksissa laitetaan piirilevyn sijainti pohjalevyssä.



Kuva 46. RoutePro-ohjelmisto.



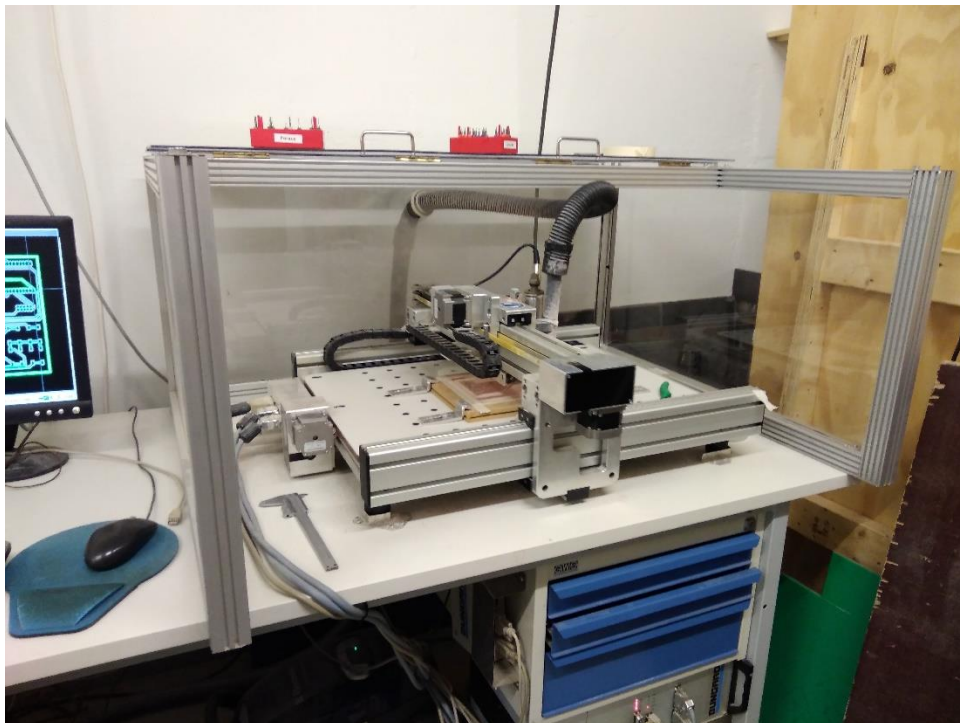
Kuva 47. Tietokone, johon oli asennettu RoutePro-ohjelmisto.

Kun ohjelmasta kaikki asetukset laitettiin kuntoon, lisättiin levy porauslaitteeseen kiinni (kuva 48). Levyn pitää olla pituudeltaan enemmän kuin 160 mm, koska piirilevyn ulkopuolelle tehdään kohdistusreiät. Pöytäleikkurilla leikattiin sopivan kokoinen levy. Levy asetetaan puualustan päälle, johon sitten porataan kohdistusreiät. Levy teipattiin tiukasti kiinni puualustaan, ettei se pääse ollemaan liikkumaan porauksen aikana.

Ensimmäiseksi porattiin levyyn kohdistusreiät. Kohdistusreiän porattiin 2,5 mm paksuisella terällä. Tämän jälkeen jyrsittiin levyn yläosa. Aluksi jyrsittiin pienemmällä terällä ja sen jälkeen isommalla terällä. Jyrsimisen aikana levystä lähtee paljon pölyä, minkä takia laitteeseen on asennettu imuri, joka imuroi suurimman osan pölyistä. Siveltimellä pystyi myös puhdistamaan levyn pölyistä.

Jyrsimisen jälkeen porattiin piirilevyn reiät. Kaikki reiät porattiin 0,8 mm:n terällä. Ohjelmassa osa rei'istä oli eri paksuisia, jonka takia ohjelma käski välillä vaihtaa poran terä. Nämä kuitattiin ja annettiin laitteen tehdä reiät samalla terällä.

Porauksen lopuksi levy käännettiin toisinpäin ja jyrsittiin levyn alapuoli. Levy asetettiin samaan kohtaan, mihin kohdistusreiät tehtiin. Tällä tavoin jyrsimiset tulee levyille oikeisiin paikkoihin.



Kuva 48. Piirilevyn jyrsimislaite.

Jyrsimisen jälkeen piirilevylle piti tehdä jälkikäsitteily hiomapaperilla. Porauskohdista jää karkea kuparipinta piirilevylle, joten ne piti hioa sileämmäksi. Piirilevyn pitäisi kestää hiomapaperin käsitteilyä.

Piirilevyn jyrsimisen lopputulos ei ollut hyvä tai odotettavissa. Jyrsimisen aikana piirilevyn pinnalta oli lähtenyt liikaa kuparia, joka haittaa suuresti. Esimerkiksi GPIO-liittimen kohdalta oli lähtenyt pinnalta lähes kaikki kupari. Samalla huomattiin, että laite oli jyrsinyt piirilevyä liian syvältä joistakin kohtaa. Tämä voi johtua siitä, että terät olivat huonossa kunnossa, asetukset eivät olleet oikein ohjelmassa tai muusta satunnaisesta syystä. Piirilevyä koitettiin muutaman kerran toteuttaa uudelleen, mutta lopputulos ei ollut siltikään hyvä. Toimeksiantajan kanssa päätettiin lopulta, että tilataan ihan uusi piirilevy tehtaalta. Tehtaalta löytyy tarkat koneet, jotka tekevät hyvää jälkeä piirilevystä.

Työn loppupuolella tehtiin loppuviimeistely piirilevysuunnitelmasta ja tarkistettiin kaikki kytkennät toimeksiantajan kanssa. Kaikki työ palautettiin Gitlab-pilvipalveluun, josta voi myöhemmin hakea tarvittavia piirilevy-tiedostoja ja itse koodattuja ohjelmia.

5 Pohdinta

Uusi opetuskortti tulee käyttöön uusille opiskelijoille, joten oli hyvin tärkeää, että opetuskortista laadittiin tarkka suunnitelma sen kytkennöistä ja toiminnasta. Opiskelijan on hyvä ymmärtää kuinka opetuskorttia olisi tarkoitus käyttää ja mitä erilaisia ominaisuuksia opetuskortista löytyy. Opetuskortin olisi tarkoitus tulla opetuskäyttöön syksyllä 2022.

Opetuskortin kaikki suunnitelmat ja ohjelmatiedostot annettiin työn lopuksi toimeksiantajalle. Toimeksiantaja pystyy myöhemmin tekemään tarvittavia muutoksia opetuskorttiin, jos pitää vaihtaa vanha komponentti uuteen tai vetää kytkennät toisella tavalla. Esimerkiksi vastuksien kytkennät voisi suunnitella ja toteuttaa pintaliitoksina.

LoRaWAN-radiomoduuleja, joita käytettiin opetuskortissa, saatiin tilattua yhteensä 4 kappaletta. RAK811-lautamoduulien tuotto oli lopetettu, joten niitä ei ole enää saatavilla. Radiomoduulin kytkentää joutuu opetuskorttiin suunnittelemaan uudelleen, jos myöhemmin otetaan käyttöön toisenlainen LoRaWAN-radiomoduuli. Kytkentää on helppo muokata EAGLEssä, koska ESP32:sta on valmiiksi vedetty RX- ja TX-portit radiomoduulin paikalle.

Tässä opinnäytetyössä ei saatu valmista opetuskorttia, koska aika loppui kesken. LoRaWAN-radiomoduulin testaus ESP32:n kanssa vei suurimman osan ajasta. Aluksi oli epäselvää, miten radio ja ESP32 pitäisi kytkeä yhteen, koska joitain RX- ja TX-pinnejä ei voinut käyttää, koska ne oli varattu muihin toimintoihin. Lisäksi oli epäselvää, tarvitsiko radiomoduuli UART-tiedonsiirron lisäksi muita pinnejä toimiakseen. Testauksissa radiomoduulilla ei välillä saanut yhteyttä asemaan, koska joko aseman päässä oli yhteysongelmia tai ESP32 ei saanut yhteyttä radiomoduuliin. Testauksia tehtiin yhdessä toisen opiskelijan ja toimeksiantajien kanssa. Lopulta saatiin yhteys toimimaan radion kanssa ja tilattua tehtaalta uudet opetuskortit, kun kytkennät oli selvitetty ja testattu.

Jatkokehityksessä opetuskorttiin voisi tehdä tarkemmat ja paremmat merkinnät kaikista komponenteista piirilevysuunnitelmaan. Komponenttien paikkoja voisi sijoittaa järkevimpiin paikkoihin ja suunnitella pystyisikö kytkennät tekemään paremmin ESP32:sta. Lopuksi suunnitella, pystyisikö kytkennät tekemään paksuimmilla vedoilla, kun jyrsitään koulun jyrsimislaitteella.

6 Yhteenveto

Opinnäytetyön aiheena oli suunnitella ja toteuttaa Kajaanin ammattikorkeakoululle uusi opetuskortti, joka tulisi käyttöön uusille opiskelijoille opetuskäyttöön. Opetuskorttia käytettäisiin mikrokontrolleriopetuksissa. Tämän lisäksi opetuskorttiin lisättiin LoRaWAN-radiomoduuli, johon opiskelijat pääsisivät myös tutustumaan. Radiomoduuliksi valittiin RAK811 LoRaWAN.

Työn alussa vertailtiin erilaisia mikrokontrollereja ja tehtiin taulukko kaikista komponenteista. Mikrokontrollerista piti vähintään löytyä UART-tiedonsiirtotekniikka ja tarpeeksi bittinopeutta, jotta se pystyisi kommunikoimaan LoRaWAN-radiomoduulin kanssa. Samalla piti ottaa huomioon, että siihen sai kaikki muut komponentit kytkettyä.

Mikrokontrolleriksi valittiin ESP32, jota käytettiin paljon opetuksissa ja sitä oli hyvin saatavilla. Siitä löytyy valmiiksi paljon ominaisuuksia, joita pystyi hyödyntämään opetuskortin suunnittelussa. Opetuskorttiin suunniteltiin Nokia 5110 LCD-näyttö, johon pystyy tulostamaan erilaisia arvoja reaaliaikaisesti. ESP32:sta vedettiin opetuskorttiin JTAG-ohjelmointiliitin, jonka kautta pystyy testaamaan ESP32:n ohjelmointia. Toimeksiantajan toiveena oli, että opetuskorttiin tulisi myös kytkimiä ja ledejä, jotka olisivat kiinteänä opetuskortissa. Suunnittelun loppuvaiheessa opetuskorttiin lisättiin kaksi 1-Wire liittimiä ja yksi analoginen LDR-sensori.

ESP32:seen sai LoRaWAN-radiomoduulin kytkettyä ja testattua, joka oli tämän työn tärkein tavoite. LoRaWAN-verkkotekniikkaa halutaan enemmän käyttää mikrokontrolleri kursseissa ja oli hyvin tärkeää, että sen sai toimimaan opetuskortissa. Testauksien jälkeen opetuskorttiin lisättiin radiomoduulille jumpperi, jonka avulla voidaan käskä radiomoduulin lepotilaan, kun ei haluta käyttää sitä.

Opetuskorttia ei saatu ajoissa toteutettua loppua, koska LoRaWAN testaukset veivät suurimman osan ajasta. Mutta opetuskortti saatiin hyvin suunniteltua ja dokumentoitua, että siitä pystyy joku jatkamaan ja kehittämään työtä eteenpäin. Työn lopussa opetuskortin piirilevysuunnitelma lähetettiin tehtaalte, jossa ne tekevät hyvänlaatuisen piirilevyn. Tavoitteena oli, että opetuskorttia olisi päässyt juottamaan ja testaamaan, mutta se jäi tekemättä.

Lähteet

1. Brett Daniel. What Are Embedded Systems? [Internet]. Trenton Systems; 2021 [Viitattu 17.02.2022]. Saatavilla: <https://www.trentonsystems.com/blog/what-are-embedded-systems>
2. Lithmee. Difference Between Hard and Soft Real Time System [Internet]. Difference Between; 2018 [Viitattu 17.02.2022]. Saatavilla: <https://www.differencebetween.com/difference-between-hard-and-vs-soft-real-time-system/>
3. GreeksforGeeks. Difference between Hard real time and Soft real time system [Internet]. 2020 [Viitattu 19.02.2022]. Saatavilla: <https://www.geeksforgeeks.org/difference-between-hard-real-time-and-soft-real-time-system/>
4. Banggood. Car Cruise Control Stalk Switch with Harness 8463234011 8463234017 for Toyota Lexus Scion [Internet]. [Viitattu 18.3.2022]. Saatavilla: <https://www.banggood.com/Car-Cruise-Control-Stalk-Switch-with-Harness-8463234011-8463234017-for-Toyota-Lexus-Scion-p-1542281.html>
5. Raspberry Pi Foundation. What is a Raspberry Pi? [Internet]. [Viitattu 19.02.2022]. Saatavilla: <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>
6. ElectronicWings. Raspberry Pi Introduction [Internet]. [Viitattu 20.02.2022]. Saatavilla: <https://www.electronicwings.com/raspberry-pi/raspberry-pi-introduction>
7. Nick Heath. What is the Raspberry Pi 4? Everything you need to know about the tiny, low-cost computer [Internet]. ZDNet; 2019 [Viitattu 20.02.2022]. Saatavilla: <https://www.zdnet.com/article/what-is-the-raspberry-pi-4-everything-you-need-to-know-about-the-tiny-low-cost-computer/>
8. Elfa Distrelec. RPI4 CASE RED/WHITE BULK - Raspberry Pi 4 -kotelo, vadelmanpunainen/valkoinen [Internet]. [Viitattu 18.3.2022]. Saatavilla: <https://www.elfadist-relec.fi/fi/raspberry-pi-kotelo-vadelmanpunainen-valkoinen-raspberry-pi-rpi4-case-red-white-bulk/p/30152783>
9. Arduino. What is Arduino? [Internet]. 2018 [Viitattu 20.02.2022]. Saatavilla: <https://www.arduino.cc/en/Guide/Introduction>

10. Sparkfun. What is an Arduino? [Internet]. [Viitattu 20.02.2022]. Saatavilla: <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all>
11. Electronics Hub. Microcontroller Types and Applications [Internet]. 2015 [Viitattu 23.02.2022]. Saatavilla: <https://www.electronicshub.org/microcontrollers/>
12. Kumar. Microcontroller Basics [Internet]. Codrey Electronics; 2018 [Viitattu 23.02.2022]. Saatavilla: <https://www.codrey.com/microcontroller/microcontroller-basics/>
13. Learning about Electronics. What is a Flash Programmer? [Internet]. [Viitattu 23.02.2022]. Saatavilla: <http://www.learningaboutelectronics.com/Articles/Flash-programmer.php>
14. OpenLab. Timers in 8051 microcontroller [Internet]. [Viitattu 23.02.2022]. Saatavilla: <https://openlabpro.com/guide/timers-8051/>
15. Swaroop. What is Serial Communication and How it works? [Internet]. Codrey Electronics; 2018 [Viitattu 23.02.2022]. Saatavilla: <https://www.codrey.com/embedded-systems/serial-communication-basics/>
16. Espressif Systems. ESP32 Series Datasheet [Internet]. 2021 [Viitattu 24.02.2022]. Saatavilla: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
17. Espressif Systems. ESP32-S Series [Internet]. [Viitattu 24.02.2022]. Saatavilla: <https://www.espressif.com/en/products/modules>
18. DroneBot Workshop. Introduction to ESP32 - Getting Started [Video]. 2020 [Viitattu 24.02.2022]. Saatavilla: https://www.youtube.com/watch?v=xPIN_Tk3VLQ
19. Robomaa. NODEMCU-32S ESP32 WIFI+BLUETOOTH DEVELOPMENT BOARD [Internet]. [Viitattu 18.3.2022]. Saatavilla: http://robomaa.fi/index.php?route=product/product&product_id=2230
20. Microcontrollerslab. TTGO LoRa32 SX1276 OLED Board with Arduino IDE [Internet]. 2022 [Viitattu 24.02.2022]. Saatavilla: <https://microcontrollerslab.com/ttgo-lora32-sx1276-oled-board-pinout-getting-started-with-arduino-ide/>

21. Robomaa. ESP32-CAM, CAMERA MODULE BASED ON ESP32 [Internet]. [Viitattu 24.02.2022]. Saatavilla: http://robomaa.fi/index.php?route=product/product&product_id=2232
22. DFRobot. ESP32-CAM Development Board [Internet]. [Viitattu 24.02.2022]. Saatavilla: https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/DFR0602_Web.pdf
23. Robomaa. SPARKFUN ESP32 THING [Internet]. [Viitattu 24.02.2022]. Saatavilla: http://robomaa.fi/index.php?route=product/product&product_id=1904
24. Swaroop. What is Serial Communication and How it works? [Internet]. Codrey Electronics; 2018 [Viitattu 24.02.2022]. Saatavilla: <https://www.codrey.com/embedded-systems/serial-communication-basics/>
25. Jimblom. Serial Communication [Internet]. Sparkfun [Viitattu 24.02.2022]. Saatavilla: <https://learn.sparkfun.com/tutorials/serial-communication/all>
26. Scott Campbell. BASICS OF THE I2C COMMUNICATION PROTOCOL [Internet]. Circuit Basics [Viitattu 18.3.2022]. Saatavilla: <https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/>
27. Scott Campbell. BASICS OF THE SPI COMMUNICATION PROTOCOL [Internet]. Circuit Basics [Viitattu 18.3.2022]. Saatavilla: <https://www.circuitbasics.com/basics-of-the-spi-communication-protocol/>
28. Triopak. DHT-22 digitaalinen kosteus- ja lämpötila-anturi 3–5,5 V [Internet]. [Viitattu 18.3.2022]. Saatavilla: <https://www.triopak.fi/fi/tuote/DHT22>
29. Kumar. UART Communication Protocol – How it works? [Internet]. Codrey Electronics; 2022 [Viitattu 25.02.2022]. Saatavilla: <https://www.codrey.com/embedded-systems/uart-serial-communication-rs232/>
30. Scott Campbell. BASICS OF UART COMMUNICATION [Internet]. Circuit Basics [Viitattu 18.3.2022]. Saatavilla: <https://www.circuitbasics.com/basics-uart-communication/>
31. Bluetooth. Bluetooth Wireless Technology [Internet]. 2022 [Viitattu: 27.4.2022]. Saatavilla: <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>

32. Jarett Ziemer. Bluetooth Versions Compared: What's the Difference? [Internet]. Soundgearlab; 2022 [Viitattu 27.4.2022]. Saatavilla: <https://soundgearlab.com/bluetooth-version-compared/>
33. Andy G. Different Bluetooth Versions: What You Need to Know [Internet]. Headphonesty; 2021 [Viitattu 25.02.2022]. Saatavilla: <https://www.headphonesty.com/2021/01/bluetooth-versions/>
34. Semtech. What Is LoRa? [Internet]. [Viitattu 25.02.2022]. Saatavilla: <https://www.semtech.com/lora/what-is-lora>
35. Eric B. LoRa [Internet]. LoRa; 2018 [Viitattu 27.4.2022]. Saatavilla: <https://lora.readthedocs.io/en/latest/#lora-networks>
36. Lora Alliance. About the LoRaWAN Standards [Internet]. [Viitattu 25.02.2022]. Saatavilla: <https://lora-alliance.org/lorawan-for-developers/>
37. Trendmicro. LoRaWAN [Internet]. [Viitattu 25.02.2022]. Saatavilla: <https://www.trendmicro.com/vinfo/us/security/definition/lorawan>
38. Adnan Aqeel. Introduction to Arduino IDE [Internet]. The Engineering Projects; 2018 [Viitattu 25.02.2022]. Saatavilla: <https://www.theengineeringprojects.com/2018/10/introduction-to-arduino-ide.html>
39. Botsolvers. What is Arduino IDE? And its different functions [Internet]. [Viitattu 25.02.2022]. Saatavilla: <https://botsolvers.com/what-is-arduino-ide-and-its-different-functions/>
40. Texas Instruments. TINA-TI [Internet]. [Viitattu 25.02.2022]. Saatavilla: <https://www.ti.com/tool/TINA-TI>
41. Texas Instruments. Getting Started with TINA-TI [Internet]. 2008 [Viitattu 25.02.2022]. Saatavilla: <https://www.ti.com/lit/ug/sbou052a/sbou052a.pdf>
42. Jimblom. How to Install and Setup EAGLE [Internet]. Sparkfun [Viitattu 25.02.2022]. Saatavilla: <https://learn.sparkfun.com/tutorials/how-to-install-and-setup-eagle/all>

43. Microchip. 8-bit PIC and AVR Microcontrollers [Internet]. Microship [Viitattu 14.3.2022]. Saatavilla: <https://www.microchip.com/en-us/products/microcontrollers-and-microprocessors/8-bit-mcus>
44. Rak. RAK811 Breakout Board Datasheet [Internet]. Rak [Viitattu 14.3.2022]. Saatavilla: <https://docs.rakwireless.com/Product-Categories/WisDuo/RAK811-Breakout-Board/Datasheet/>
45. LoRa Alliance. LoRaWAN 1.0.3 Specification [Internet]. LoRa Alliance [Viitattu 14.4.2022]. Saatavilla: <https://lora-alliance.org/wp-content/uploads/2020/11/lorawan1.0.3.pdf>
46. Ravi Teja. ESP32 Nokia 5110 LCD Display Tutorial | Display Text, Adjust Contrast [Internet]. Electronics Hub [Viitattu 14.3.2022]. Saatavilla: <https://www.electronicshub.org/esp32-nokia-5110-lcd/>
47. Elfa Distrelec. RND 205-00638 - Suora Uros Piirikorttirima, Läpivientiaukko, 2 Rivit, 14 Kontaktit, 2.54mm Rasteri, RND Connect [Internet]. Elfa Distrelec [Viitattu 18.4.2022]. Saatavilla: <https://www.elfadistrelec.fi/fi/suora-uros-piirikorttirima-laepivientiaukko-rivit-14-kontaktit-54mm-rasteri-rnd-connect-rnd-205-00638/p/30093658>
48. Elfa Distrelec. RND 205-00654 - Suora Naaras, Piirikorttivastake, Läpivientiaukko, 2 Rivit, 10 Kontaktit, 2.54mm Rasteri, RND Connect [Internet]. Elfa Distrelec [Viitattu 18.4.2022]. Saatavilla: <https://www.elfadistrelec.fi/fi/suora-naaras-piirikorttivastake-laepivientiaukko-rivit-10-kontaktit-54mm-rasteri-rnd-connect-rnd-205-00654/p/30093674>
49. AliExpress. OPEN-SMART Common Cathode 8 Bit 8 LED Bar Marquee LED Display Module w/ 4 Kinds of Color for Arduino [Internet]. AliExpress [Viitattu 5.4.2022]. Saatavilla: <https://www.aliexpress.com/item/32909606852.html>
50. AliExpress. 5pcs/set Electronic Building Blocks Big Key Button Module large Button Module Five Key model for Arduino [Internet]. AliExpress [Viitattu 5.4.2022]. Saatavilla: <https://www.aliexpress.com/item/32967500311.html>
51. Voltaat. DHT22 Temperature & Humidity Sensor [Internet]. Voltaat [Viitattu 14.4.2022]. Saatavilla: <https://www.voltaat.com/products/dht22-temperature-humidity-sensor>
52. Studio Pieters. ESP32 – PinOut [Internet]. Studio Pieters [Viitattu 14.4.2022]. Saatavilla: <https://www.studiopieters.nl/esp32-pinout/>

53. Last Minute Engineers. Interface Nokia 5110 Graphic LCD Display with Arduino [Internet]. Last Minute Engineers [Viitattu 14.4.2022]. Saatavilla: <https://lastminuteengineers.com/nokia-5110-lcd-arduino-tutorial/>

Liitteet

Liite 1. Komponenttien testauksessa käytetty Arduino IDE -ohjelma

```
#include <DHT.h>

#include <SPI.h>

#include <Adafruit_GFX.h>

#include <Adafruit_PCD8544.h>

#define DHT_SENSOR_PIN 4 // ESP32 pin GIOP4 connected to DHT22 sensor

#define DHT_SENSOR_TYPE DHT22

#define BUTTON1 32

#define BUTTON2 33

#define BUTTON3 34

#define BUTTON4 35

#define LED1 25

#define LED2 26

#define LED3 27

#define LED4 19

int BUTTONstate1 = 0; // To read the button1 state

int BUTTONstate2 = 0; // To read the button2 state

int BUTTONstate3 = 0; // To read the button3 state

int BUTTONstate4 = 0; // To read the button4 state

DHT dht_sensor(DHT_SENSOR_PIN, DHT_SENSOR_TYPE);

/* Declare LCD object for SPI

Adafruit_PCD8544(CLK,DIN,D/C,CE,RST); */

Adafruit_PCD8544 display = Adafruit_PCD8544(18, 23, 0, 5, 2);
```



```
int contrastValue = 60; // Default Contrast Value

const int adcPin = 34;

int adcValue = 0;

void setup() {

  Serial.begin(9600);

  dht_sensor.begin(); // initialize the DHT sensor

  /* Initialize the Display*/

  display.begin();

  /* Change the contrast using the following API*/

  display.setContrast(contrastValue);

  /* Clear the buffer */

  display.clearDisplay();

  display.display();

  delay(1000);

  pinMode(BUTTON1, INPUT); // Define Button1 as input pin

  pinMode(BUTTON2, INPUT); // Define Button2 as input pin

  pinMode(BUTTON3, INPUT); // Define Button3 as input pin

  pinMode(BUTTON4, INPUT); // Define Button4 as input pin

  pinMode(LED1, OUTPUT); // Define LED1 as output pin

  pinMode(LED2, OUTPUT); // Define LED2 as output pin

  pinMode(LED3, OUTPUT); // Define LED3 as output pin

  pinMode(LED4, OUTPUT); // Define LED4 as output pin

}

void loop() {

  BUTTONstate1 = digitalRead(BUTTON1); // Read button1 state

  if (BUTTONstate1 == LOW)
```

```
{  
    digitalWrite(LED1, HIGH); // Turn LED1 OFF  
}  
else{  
    digitalWrite(LED1, LOW); // Turn LED1 ON  
    displayValue(); // Read temperature and humidity values  
}  
  
BUTTONstate2 = digitalRead(BUTTON2); // Read button2 state  
if (BUTTONstate2 == HIGH)  
{  
    digitalWrite(LED2, LOW); // Turn LED2 OFF  
}  
else{  
    digitalWrite(LED2, HIGH); // Turn LED2 ON  
}  
  
BUTTONstate3 = digitalRead(BUTTON3); // Read button3 state  
if (BUTTONstate3 == HIGH)  
{  
    digitalWrite(LED3, LOW); // Turn LED3 OFF  
}  
else{  
    digitalWrite(LED3, HIGH); // Turn LED3 ON  
}  
  
BUTTONstate4 = digitalRead(BUTTON4); // Read button4 state  
if (BUTTONstate4 == HIGH)  
{
```

```
    digitalWrite(LED4, LOW);// Turn LED4 OFF
}
else{
    digitalWrite(LED4, HIGH);// Turn LED4 ON
}
}

void displayValue() {
    // read humidity
    float humi = dht_sensor.readHumidity();
    // read temperature in Celsius
    float tempC = dht_sensor.readTemperature();
    // read temperature in Fahrenheit
    float tempF = dht_sensor.readTemperature(true);

    // check whether the reading is successful or not
    if ( isnan(tempC) || isnan(tempF) || isnan(humi) ) {
        //Serial.println("Failed to read from DHT sensor!");
        display.clearDisplay();
        display.setTextColor(WHITE, BLACK);
        display.setCursor(0,1);
        display.setTextSize(2);
        display.print(" |ESP32| ");
        display.setTextSize(1);
        display.setTextColor(BLACK);
        display.setCursor(8,22);
        display.print("Failed to read from DHT sensor!");
        display.display();
    } else {
```

```
display.clearDisplay();

display.setTextColor(WHITE, BLACK);

display.setCursor(0,1);

display.setTextSize(2);

display.print("| ESP32 |");

display.setTextSize(1);

display.setTextColor(BLACK);

display.setCursor(5,22);

display.print("Humi: ");

display.setCursor(35,22);

display.print(humi);

display.setCursor(68,22);

display.print("%");

display.setCursor(5, 38);

display.print("Temp: ");

display.setCursor(35,38);

display.print(tempC);

display.setCursor(68,38);

display.print("C");

display.display();

}

// wait a 2 seconds between readings

delay(2000);

}
```