Matti Raudaskoski

Yritysprojekti3 Matter

Loppuraportti

#### SISÄLLYS

[1 JOHDANTO 3](#_Toc149819478)

[2 Projektin taustat 4](#_Toc149819479)

[2.1 Yleistä nRF5340DK:sta 5](#_Toc149819480)

[2.2 nRF5340DK:n tukemat langattomat protokollat. 6](#_Toc149819481)

[2.2.1 Bluetooth Low Energy (LE). 6](#_Toc149819482)

[2.2.2 Mesh protokollat. 7](#_Toc149819483)

[2.2.2.1 Bluetooth mesh. 7](#_Toc149819484)

[2.2.2.2 Thread mesh. 10](#_Toc149819485)

[2.2.2.3 Zigbee mesh 19](#_Toc149819486)

[2.2.3 Matter 22](#_Toc149819487)

[2.2.3.1 Matter-verkon topologia 22](#_Toc149819488)

[2.2.3.2 Matter protokollan yleinen rakenne 24](#_Toc149819489)

[2.2.3.3 Matter-pinon yleinen rakenne 25](#_Toc149819490)

[2.2.3.4 Matter-ohjain 27](#_Toc149819491)

[2.2.3.5 Matter Bridge 28](#_Toc149819492)

[2.2.4 Near Field Communications (NFC) 31](#_Toc149819493)

[2.2.5 ANT/ANT+ protokolla 36](#_Toc149819494)

[3 Työskentely projektissa 38](#_Toc149819495)

[3.1 Thread Border Routerin rakentaminen 39](#_Toc149819496)

[3.2 Chip-tool asennus Matter over Thread 43](#_Toc149819497)

[4 Projektin tulokset 48](#_Toc149819498)

[5 Yhteenveto 49](#_Toc149819499)

[lähteet 50](#_Toc149819500)

LIITTEET 55

# JOHDANTO

Matter yritysprojekti 3 tarkoituksena oli tutustua nRF5340DK kehitysalustaan ja sen ominaisuuksiin sekä tuottaa opetuskäyttöön soveltuvaa materiaalia.

Projektia tehtiin koululle kolmen laite- ja tuotesuunnittelun suuntautumisvaihtoehdon opiskelijan yhteistyössä: Jere Manninen, Mikko Salojärvi ja Matti Raudaskoski.

Projekti toteutetaan OAMK:n tiloissa opettajan ohjaamana, OAMK toimiin projektin tilaajana.

Projektin tavoitteet:

Projektin tavoitteena oli selvittää ja toteuttaa nRF5340DK ja Thingy:53 kehitysalustoilla:

* Demo SEGGER JLink debuggerin käytöstä.
* nRF5340DK.
* Thingy:53.
* Käytettävissä olevat tietoliikenneprokollat nRF5340DK
* Bluetooth Low Energy.
* Mesh protokollat kuten
* Bluetooth mesh.
* Thread.
* Zigbee.
* Matter.
* NFC.
* 802.15.4
* ANT.
* Opetuskäyttöön soveltuva demo 802.15.4 radion avulla
  + Thread CoAP server & client
    - Sniffer.
    - Wireshark demo.
  + Matter
    - OTBR Raspberry Pi
    - nRF52840dongle RCP

Opiskelijoiden tavoitteena oli suorittaa yritysprojekti 3 (tai Jerellä 2) niminen opintojakso ja saada 10 op 270 tunnin työpanoksesta projektissa ja toisaalta toteuttaa projektin vaatimukset ja esittää projektin saavutukset tilaajalle.

# Projektin taustat

Yritysprojektissa 3 oli tavoitteena ensin tutustua kehitysalustoihin nRF5340DK (1) ja Thingy:53 (2) ja niiden ominaisuuksiin ja tuottaa käyttökelpoista materiaalia opetuskäyttöön tilaajalle.

Koska projektia tehtiin kolmen oppilaan yhteistyönä, oli järkevää yhdessä selvittää ensiksi kehitysalustoiden tukemat langattomat protokollat. Tässä nimenomaan keskityttiin nRF5340DK kehitysalustaan.

Tämän jälkeen projekti jaettiin vastuu osiin:

* Jere: SEGGER JLink (3) Debugger
* nRF5340DK
* Thingy:53
* Mikko: Thread CoAP server & client
* Sniffer (4).
* Wireshark (5) demo.
* Matti: Matter
* OTBR Raspberry Pi 4.
* nRF52840dongle RCP.
* Matter controller

Yritysprojektiin on varattu vähintään 10 opintopisteen verran aikaa eli 200 tuntia työskentelyyn + 70 tuntia dokumentointiin. Ja projektipalaverissa sovittiin loppupalaverin aikatauluksi 20.10.2023 klo 10.00.

Ja minulle vastuualueeksi tuli Matter järjestelmä, Open Thread Border Router (Raspberry pi 4 ) OTBR + Radio Co-Processor (nRF52840 dongle) RPC ja Matter Controller. Tästä lisää myöhemmin.

## Yleistä nRF5340DK:sta

Koska OAMK:lle on hankittu Nordic Semiconductorin nRF5340 Multicore System on Chip piirisarjaan pohjautuvia nRF5340DK jaThingy:53 kehitysalustoja. Niin oli luontevaa valita projektiin myös nämä alustat. Kuvassa 1 (6) nRF5340DK.

nRF5340DK kehitysalusta on nRF5340 kahden prosessorin SoC.

* 128 MHz Arm® Cortex®-M33 -sovellusprosessori, 1 Mt Flash ja 512 kt RAM.
* 64 MHz Arm® Cortex®-M33 -verkkoprosessori, 256 kt Flash ja 64 kt RAM.

Se tukee laajaa valikoimaa langattomia protokollia:

* Bluetooth Low Energy, high-throughput 2 Mbps, Advertising Extensions ja Long Range.
* Mesh protokollat: Bluetooth mesh, Thread ja Zigbee.
* Matter- ja HomeKit -sovellukset.
* NFC (Near-Field communication) lähimaksu.
* ANT.
* IEEE 802.15.4 wireless networking standard developed for low-power, low-data-rate apps

NFC-antenni

SEGGER J-Link OB debugger

Arduino Uno Rev3 yhteensopivia

Neljä painiketta ja neljä LEDiä , ja ne ovat kaikki käyttäjän ohjelmoitavissa.

Kuva, joka sisältää kohteen teksti, virtapiiri, sähkökomponentti, Sähkötekniikka

Kuvaus luotu automaattisesti

*Kuva 1: nRF5340DK*

## nRF5340DK:n tukemat langattomat protokollat.

Tässä lyhyesti nRF5340DK sovellusalustan kanssa käytettävistä tietoliikenneprokollista. Bluetooth Low Energy / -mesh, Thread-mesh ja Matter – protokollat ovat tässä, ehkä tärkeimmät varsinkin, jos ajatellaan projektissa demottavan Matter-systeemiä.

### Bluetooth Low Energy (LE).

Bluetooth Low Energyn (7) tärkein etu on sen alhaiset kustannukset verrattuna muihin vähätehoisiin henkilökohtaisiin verkkoihin, mikä tekee siitä houkuttelevan sovelluksille, jotka vaativat massakäyttöä.

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti, numero

Kuvaus luotu automaattisestiTekniikka on yleistä myös älypuhelimissa (useimmat älypuhelimet tukevat sekä Bluetooth Classicia että Bluetooth LE:tä), mikä tekee sovellusten testaamisesta ja prototyypistämisestä helppoa kaikkialla. Älypuhelimen lisäksi tarvitaan vielä yksi Bluetooth LE -laite, jotta voidaan testata kaksisuuntaista viestintää. Koska käytännössä kaikilla on älypuhelin, tämä vähentää testien suorittamisen kustannuksia ja monimutkaisuutta verrattuna muihin teknologioihin, joissa tarvitaan erityisiä laitteita. Kuvassa 2 Bluetooth LE protokollapino

*Kuva 2: Bluetooth LE protokollapino.*

Yläosassa on sovellus. Tämä on kerros, jonka kanssa käyttäjä on vuorovaikutuksessa API:iden kautta käyttääkseen Bluetooth LE -protokollaa. Sovelluskerroksen tärkeitä osia ovat profiilit, palvelut ja ominaisuudet. Seuraavat kerrokset muodostavat isännän (Host), joka olennaisesti määrittää, kuinka Bluetooth LE -laitteet tallentavat ja vaihtavat tietoja keskenään. Lopuksi ohjain muodostaa alemmat kerrokset, joista merkittävin osa on fyysinen radio, joka tuottaa radioaallot ja koodaa signaalit datalla, jonka haluat lähettää.

**Host**

Bluetooth LE -isäntä koostuu seuraavista kerroksista:

* Loogisen linkin ohjaus- ja mukautusprotokolla (L2CAP): tarjoaa tietojen kapselointipalvelut ylemmille kerroksille.
* Security Manager Protocol (SMP): määrittää ja tarjoaa menetelmät suojattua viestintää varten.
* Attribute Protocol (ATT): sallii laitteen paljastaa tiettyjä tietoja toiselle laitteelle.
* Generic Attribute Profile (GATT): määrittää tarvittavat alimenettelyt ATT-kerroksen käyttöä varten.
* Generic Access Profile (GAP): liitetään suoraan sovellukseen laitteen etsinnän ja yhteyksiin liittyvien palvelujen käsittelemiseksi.

**Ohjain**

Bluetooth LE -ohjain koostuu seuraavista kerroksista:

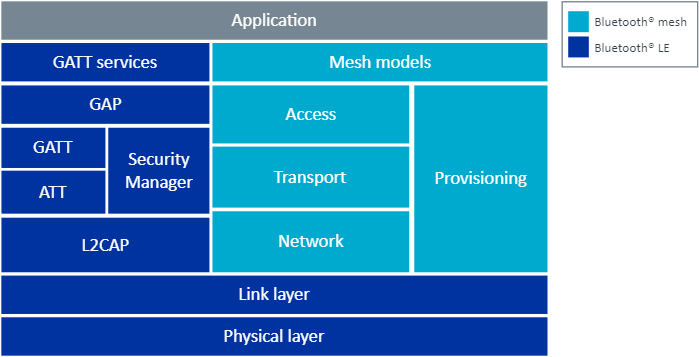
* Physical Layer (PHY): määrittää, kuinka todellinen data moduloidaan radioaalloille ja kuinka se lähetetään ja vastaanotetaan.
* Link Layer (LL): hallitsee radion tilaa, joka määritellään joksikin seuraavista: valmiustila, mainonta, skannaus, aloitus, yhteys.

### Mesh protokollat.

nRF5340DK alusta tukee useita mesh protokollia, kuten Bluetooth mesh, Thread, Zigbee

### Bluetooth mesh.

Bluetooth-verkko perustuu Bluetooth® 4.0 -määrityksen Bluetooth LE -osaan ja jakaa alimmat kerrokset tämän protokollan kanssa. Kuvassa 3 näkyy BLE :n ja Bluetooth mesh välinen suhde (8).



*Kuva 3: Bluetooth meshin ja Bluetooth LE -määritysten välinen suhde*

**Sovellusalueet**

Bluetooth mesh kohdistuu ensisijaisesti yksinkertaisiin ohjaus- ja valvontasovelluksiin, kuten valonhallintaan tai anturitietojen keräämiseen. Pakettimuoto on optimoitu pienille ohjauspaketeille, jotka antavat yksittäisiä komentoja tai raportteja, eikä sitä ole tarkoitettu datan suoratoistoon tai muihin suuren kaistanleveyden sovelluksiin. Sen käyttö kuluttaa enemmän virtaa kuin perinteisiä Bluetooth LE -sovelluksia käytettäessä. Johtuu pääasiassa tarpeesta pitää radio jatkuvasti päällä.

**Verkkotopologia ja välitys (relaying)**

Bluetooth mesh on lähetyspohjainen verkkoprotokolla, jossa jokainen verkon laite lähettää ja vastaanottaa kaikki viestit kaikille radioalueella oleville laitteille ja niiltä.

Mesh-verkossa ei ole käsitettä yhteyksistä. Mikä tahansa verkon laite voi välittää viestejä mistä tahansa muusta laitteesta. Tämän ominaisuuden avulla mesh-laite voi lähettää viestejä radioalueen ulkopuolella olevaan laitteeseen siten, että yksi tai useampi laite välittää viestit määränpäähän. Tämän ominaisuuden avulla laitteet voivat myös liikkua ja liittyä verkkoon ja sieltä pois milloin tahansa.

**Mesh kuljetus (transport)**

Bluetooth mesh hyödyntää Bluetooth LE -mainostajan ja skannerin rooleja, jotka kommunikoivat Bluetooth LE -mainospakettien kautta. Läheiset mesh-laitteet poimivat mainospaketit ja niitä käsitellään kuten muita Bluetooth LE -mainospaketteja. Mesh-paketit esitetään ainutlaatuisella AD-tyypillä (advertisement data) ja lisätään mainospakettien hyötykuormaan.

Bluetooth LE -laitteet lähettävät mainospaketteja säännöllisin mainosvälein, eivätkä verkkopaketit ole poikkeus. Kuitenkin toisin kuin perinteiset mainostajat, mesh-laitteet muuttavat mainosten hyötykuormaa jokaisessa lähetyksessä ja lähettävät uusia mesh-paketteja, kun ne ovat jonossa mesh-pinossa. Jos mesh-verkossa tai mesh-pinossa ei ole liikennettä tai sovellus ei tuota viestejä, laitteet pysyvät hiljaa, kunnes on lähetettävää.

**Releet**

Bluetooth mesh laajentaa verkon kantamaa välittämällä viestejä. Mikä tahansa mesh-laite voidaan määrittää toimimaan releenä, eikä verkon rakentamiseen tarvita erillisiä relelaitteita. Jokainen releenä toimiva laite vähentää vastaanotettujen viestien TTL (time-to-live) -arvoa ja välittää ne eteenpäin, jos TTL on kaksi tai suurempi. Tätä ohjaamatonta välitystä kutsutaan *viestitulvaksi* ja se varmistaa viestin toimitukseen suurella todennäköisyydellä ilman, että se tarvitsee tietoja verkon topologiasta. Bluetooth mesh -profiilimäärittelyssä ei ole reititysmekanismeja, ja kaikkia viestejä välittävät kaikki releet, kunnes TTL-arvo saavuttaa nollan. Kaikki mesh-laitteet ylläpitävät viestivälimuistia, jotta viestit eivät välittyisi toistuvasti samojen releiden toimesta. Sitä käytetään suodattamaan pois paketit, jotka laite on jo käsitellyt.

**Tehon kulutus**

Lähetyspohjaisen viestinnän mahdollistamiseksi laitteiden on pidettävä radionsa jatkuvasti kuuntelutilassa. Tämän vuoksi virran kulutus on huomattavasti suurempaa, kuin tyypillisessä Bluetooth LE -laitteessa.

Jotta vähän virtaa käyttävät laitteet voivat liittyä mesh-verkkoon, Bluetooth mesh sisältää ystävyysominaisuuden (*a* friendship*feature*). Tämä protokolla mahdollistaa pienitehoisille laitteille luoda yhteyden tavalliseen mesh-laitteeseen, joka sitten tallentaa viestit välimuistiin ja välittää ne säännöllisin väliajoin pienitehoiselle laitteelle. Tämä auttaa vähän virtaa kuluttavaa laitetta, jonka ei tarvitse olla jatkuvasti päällä kuunnellakseen saapuvia viestejä.

**GATT-välityspalvelin**

Mahdollistaakseen tuen vanhoille Bluetooth LE -laitteille, jotka eivät tue mesh-pakettien vastaanottamista, Bluetooth mesh määrittää erillisen protokollan mesh-viestien tunneloimiseksi Bluetooth LE GATT -protokollan kautta. Tätä tarkoitusta varten Bluetooth mesh -profiilimäärittelyssä määritellään GATT-siirtotie ja vastaava GATT-välityspalvelinprotokolla. Tämän protokollan avulla vanhat Bluetooth LE -laitteet voivat liittyä mesh-verkkoon muodostamalla GATT-yhteyden mesh-laitteeseen, jossa välityspalvelinominaisuus on käytössä.

Vanhalle laitteelle annetaan osoite ja tarvittavat avaimet, jotta siitä tulee verkon täysivaltainen jäsen. Laite vastaanottaa suojaustiedot tavanomaisen hallintamenettelyn tai jonkin kaistan ulkopuolisen mekanismin kautta.

**Osoittaminen**

Bluetooth mesh -osoitusmalli eroaa Bluetooth LE -osoitusmenetelmästä. Siinä on kolmenlaisia osoitteita:

* **Unicast-osoitteet** - Ainutlaatuiset jokaiselle laitteelle.
* **Ryhmäosoitteet -** Laiteryhmän muodostamiseen ja niiden kaikkien osoittamiseen kerralla.
* **Virtuaaliosoitteet** - Seuraamattomat UUID-pohjaiset osoitteet, joissa on suuri osoiteavaruus.

Kun laite lisätään verkkoon, sille osoitetaan joukko yksittäislähetysosoitteita, jotka edustavat sitä. Laitteen unicast-osoitteita ei voi muuttaa, ja ne ovat aina peräkkäisiä. Unicast-osoiteavaruus tukee 32767 unicast-osoitteen käyttöä yhdessä mesh-verkossa. Mikä tahansa sovellus voi käyttää Unicast-osoitteita viestin lähettämiseen suoraan laitteeseen.

Ryhmäosoitteet allokoidaan ja määritetään osana verkon konfigurointimenettelyä. Ryhmäosoite voi edustaa mitä tahansa määrää laitteita, ja laite voi olla osa mitä tahansa määrään ryhmiä. Mesh-verkossa voi olla enintään 16127 yleiskäyttöistä ryhmäosoitetta.

Virtuaaliosoitteita voidaan pitää ryhmäosoitteiden erityisenä muotona, ja niitä voidaan käyttää edustamaan mitä tahansa laitteita. Jokainen virtuaalinen osoite on 128-bittinen UUID (Universal Unique Identifier), joka on luotu tekstitunnisteesta. Virtuaaliosoitteita ei tarvitse seurata verkon konfigurointilaitteen avulla, ja tällä tavalla käyttäjät voivat luoda virtuaalisia osoitteita ennen käyttöönottoa tai osoitteita voidaan luoda ad-hoc verkon laitteiden välillä.

**Mallit ja elementit**

Eri valmistajien laitteiden välisen viestinnän standardoimiseksi Bluetooth mesh -profiilimääritykset määrittelevät pääsykerroksen, joka reitittää mesh-viestit laitteen eri mallien välillä. Malli edustaa tiettyä käyttäytymistä tai palvelua ja määrittää joukon tiloja ja viestejä, jotka vaikuttavat näihin tiloihin. Bluetooth-verkkoprofiilin eritelmät ja mallispesifikaatiot määrittelevät joukon malleja, jotka kattavat tyypilliset käyttöskenaariot, kuten laitteen kokoonpanon, anturin lukemat ja valonsäädön. Näiden lisäksi toimittajat voivat vapaasti määritellä omat mallinsa mukana olevilla viesteillä ja tiloilla.

### Thread mesh.

**Thread**

Thread (9) on pienitehoinen mesh-verkkotekniikka, joka on suunniteltu erityisesti kotiautomaatiosovelluksia varten. Se on IPv6-pohjainen standardi, joka käyttää 6LoWPAN-tekniikkaa IEEE 802.15.4 -protokollan yli. Voit yhdistää Thread mesh -verkon Internetiin Thread Border Routerilla.

**OpenThreadin yleiskatsaus**

Open Thread on kannettava ja joustava avoimen lähdekoodin toteutus Thread-verkkoprotokollasta. Google on luonut sen aktiivisessa yhteistyössä Nordic Semiconductorin kanssa nopeuttaakseen yhdistettyyn kotiin tarkoitettujen tuotteiden kehitystä.

Thread-toteutus nRF Connect SDK:ssa perustuu OpenThread-pinoon, joka on integroitu Zephyriin. Löydät OpenThread-pinon osoitteesta ncs/modules/lib/openthread.

OpenThreadilla on muun muassa seuraavat tärkeimmät edut:

* Kapea alustan abstraktiokerros, joka tekee OpenThreadista alustan agnostisen
* Pieni muistitila
* Tuki [system-on-chip (SoC)](https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/protocols/thread/overview/architectures.html#ug-thread-architectures-designs-soc-designs) sekä [verkon rinnakkaisprosessori (NCP) ja radio co-prosessori (RCP)](https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/protocols/thread/overview/architectures.html#thread-architectures-designs-cp) suunnittelu
* Virallinen ketjusertifikaatti

Löydät lisätietoja OpenThreadista osoitteessa [OpenThread.io](https://openthread.io/) .

**Tuetut thread ominaisuudet**

Thread-protokollan OpenThread-toteutus tukee kaikkia Thread 1.3.0 -määrityksessä määriteltyjä ominaisuuksia, joita Thread 1.3 -sertifiointiohjelma edellyttää:

* Kaikki thread verkkotasot:
* IPv6
* 6 LowPAN
* IEEE 802.15.4 MAC-suojauksella
* Mesh Linkin perustaminen
* Mesh-reititys
* Kaikki laitteen roolit
* Border Router tuki
* Thread 1.2:ssa esitellyt ominaisuudet:
* Coordinated Sampled Listening (CSL)
* Link Metrics Probing
* Multicast Thread-verkkojen kautta
* Thread Domain Unicast-osoitus
* Thread 1.3.0:ssa esitellyt ominaisuudet:
* Service Registration Protocol (SRP) -asiakas
* Transport Control Protocol (TCP)

nRF Connect SDK:ssa voit valita, mitä Thread-protokollan versiota käytetään sovelluksessasi. Oletusarvoisesti nRF Connect SDK tukee Thread 1.3:a, joka sisältää tuen threadille 1.2. Voit ottaa käyttöön ja määrittää minkä tahansa Thread-version käyttämällä [erityisiä asetuksia](https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/protocols/thread/configuring.html#thread-ug-thread-specification-options).

**Threadin 1.2 ominaisuudet**

nRF Connect SDK toteuttaa kaikki Thread 1.2 -määrityksen pakolliset ominaisuudet.

Thread 1.2 -määrityksen mukana tulleet ominaisuudet ovat täysin taaksepäin yhteensopivia Thread 1.1:tä (tarkemmin sanottuna Thread 1.1.1 -määrityksiä) käyttävien toteutusten kanssa. Thread 1.2 parantaa verkon skaalautuvuutta, reagointikykyä, tiheyttä ja virrankulutusta. Lisätietoja tästä säieversiosta on [kaupallisen valkoisen kirjan virallisessa Thread 1.2 -julkaisussa](https://www.threadgroup.org/Portals/0/documents/support/ThreadInCommercialWhitePaper_2542_1.pdf) ja [Thread 1.2 Base Features](https://www.threadgroup.org/Portals/0/documents/support/Thread%201.2%20Base%20Features.pdf)-asiakirjassa.

**Coordinated Sample Listening (CSL)**

IEEE 802.15.4–2015:ssä määritelty koordinoitu näytekuuntelu otetaan käyttöön säikeen 1.2 spesifikaatiossa, joka tarjoaa matalan viiveen viestinnän [Sleepy End Devices (SED)](https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/protocols/thread/overview/commissioning.html#thread-ot-device-types) -laitteille . Thread 1.2 -reitittimiä tarvitaan tukemaan synkronoituja CSL-lähetyksiä lapsille, jotka tarvitsevat niitä. Nämä lapset tunnetaan nimellä [Synchronized Sleepy End Devices (SSED)](https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/protocols/thread/overview/commissioning.html#thread-ot-device-types).

SSED:t voivat lähettää kehyksiä normaalisti milloin tahansa, mutta reitittimien tulee käyttää CSL-lähetysmekanismia niin kauan kuin synkronointi säilyy. Tämän ansiosta SSED voi pysyä uneliaassa tilassa yli 99 % ajasta ja käynnistää radionsa vain ajoittain muutaman sadan mikrosekunnin ajan vastaanottaakseen kehyksiä vanhemmalta.

SSED-laitteiden yleisin käyttötapa on kehittää akkukäyttöisiä toimilaitteita, joilla on pieni latenssi, kuten ikkunaverhot (sälekaihtimet).

**Link Metrics Prodin Protocol**

Thread 1.2 Specification esittelee Link Metrics Probing Protocol -protokollan mekanismina kerätä tietoa linkin laadusta naapureiden kanssa.

Käytettävissä olevat mittarit ovat:

* Vastaanotettujen PDU:iden määrä.
* LQI:n eksponentiaalinen liukuva keskiarvo.
* Linkin marginaalin eksponentiaalinen liikkuva keskiarvo (dB).
* RSSI:n eksponentiaalinen liikkuva keskiarvo (dBm).

Tämän ominaisuuden tärkein mahdollinen käyttötarkoitus on sallia akkukäyttöisen laitteen alentaa lähetystehoaan, kun linkin laatu sen emolaitteen kanssa on riittävän hyvä. Näin laite voi säästää energiaa jokaisessa lähetetyssä kehyksessä. Laitetta, joka on kiinnostunut vastaanottamaan linkin laatutietoja, kutsutaan Link Metrics Initiatoriksi, kun taas laitetta, joka toimittaa nämä tiedot, kutsutaan Link Metrics Subjectiksi.

Kehittäjät voivat käyttää OpenThread-sovellusliittymää saadakseen tietoa linkin laadusta kolmella tavalla:

* Single probe: yhden laukauksen [Mesh Link Establishment (MLE)](https://openthread.io/guides/thread-primer/network-discovery#mesh-link-establishment) -komento, joka palauttaa pyydetyt tiedot vastauksessa.
* Forward Tracking Series: Aloittaja määrittää aiheen aloittamaan linkin laatutietojen seurannan jokaiselle vastaanotetulle kehykselle. Aloittaja hakee milloin tahansa kyseisen sarjan keskiarvot MLE-tietopyynnön avulla.
* Enhanced-ACK-tutkinta: Aloittaja määrittää kohteen sisällyttämään linkkimetriikkatiedot tietoelementteinä Enh-ACK-kutsuihin, jotka luodaan vastauksena aloittajan lähettämiin IEEE 802.15.4–2015 -kehyksiin.

Enhanced-ACK-lukeminen on tehokkain tapa hakea linkkimittareita, koska vaaditaan hyvin vähän viestikuluja. nRF Connect SDK tarjoaa täyden Link Metrics -tuen jopa [Radio co-processor (RCP)](https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/protocols/thread/overview/architectures.html#thread-architectures-designs-cp-rcp) -arkkitehtuurille, joka on teknisesti haastavin, koska radio-ohjaimen on käsiteltävä tietoelementtien injektio ajoissa, jotta se vastaa kuittauksen ajoitusvaatimuksia. Päätös linkkimetriikkatietojen tulkitsemisesta lähetystehon säätämiseksi jätetään sovellukselle itselleen.

**Multicast Thread-verkkojen kautta**

Säikeen 1.1 rajareitittimillä on rajoitus olla välittämättä ryhmälähetysliikennettä laajuudella, joka on suurempi kuin realm-local. Tietyissä sovelluksissa voi olla hyödyllistä pystyä ohjaamaan monilähetysryhmiä Thread-verkon ulkopuolisesta isännästä (host). Tämä saavutetaan Thread 1.2:ssa sallimalla Threadin rajareitittimien välittää monilähetysliikennettä laajuudella, joka on suurempi kuin realm-local kahdella tavalla:

* Thread-verkosta ulkoverkkoon: konfigurointivaihtoehtona rajareitittimessä jokaiselle monilähetysryhmälle.
* Ulkoverkosta Thread-verkkoon: Primary Backbone Router (PBBR) välittää vain monilähetysliikennettä, jonka kohde vastaa yhtä sen verkkoon Thread-laitteiden rekisteröimistä monilähetysryhmistä.

Toisessa tapauksessa Thread [Commissioneria](https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/protocols/thread/overview/commissioning.html#thread-ot-commissioning-roles) voidaan käyttää myös sallittujen monilähetysryhmien rekisteröimiseen laitteiden puolesta.

OpenThread-pino käsittelee automaattisesti monilähetysryhmien rekisteröinnin asianmukaisella PBBR:llä aina, kun ne on määritetty laitteeseen.

**Thread Domain Unicast-osoitus**

Thread 1.2 Specification esittelee säikeen verkkotunnusten käsitteen.

Thread Domain on joukko säikeen laitteita, jotka vastaanottavat ja käyttävät yhteistä säiealueen toimintakonfiguraatiota. Thread Domainin toimintakonfiguraatio mahdollistaa Thread Devices -laitteiden liittymisen ja osallistumisen suurempiin toisiinsa yhdistettyihin [alueisiin](https://openthread.io/guides/thread-primer/ipv6-addressing#scopes), vaikka ne ulottuvat yhden säikeen verkon rajojen ulkopuolelle. Käyttäjä tai verkon ylläpitäjä voi käyttää joko säikeen käyttöönotto- tai säikeen rajareitittimiä määrittääkseen säielaitteisiin yhteisen säiealueen toimintakonfiguraation. Thread-laitteet voivat kuulua eri säieverkkoihin tai [osioihin](https://openthread.io/guides/thread-primer/node-roles-and-types#partitions), joilla on mahdollisesti erilaiset verkkokohtaiset valtuustiedot.

**Thread 1.2 -tuen rajoitukset**

Thread 1.2 Specification -tuella on seuraavat rajoitukset:

* Koodin kokorajoituksen vuoksi Thread 1.2 -ominaisuuksien täydellisen sarjan yhdistäminen Bluetooth® LE -moniprotokollatuen kanssa ei ole mahdollista nRF52833 DK:issa.

**Thread 1.3 ominaisuudet**

Lisätietoja tästä säieversiosta on virallisessa [Thread 1.3.0 -ominaisuuksien valkoisessa kirjassa](https://www.threadgroup.org/Portals/0/documents/support/Thread1.3.0WhitePaper_07192022_3990_1.pdf).

**DNS-pohjainen palvelun etsintä**

Thread 1.3 Specification esittelee DNS-SD Service Registration Protocol -protokollan, jonka avulla laitteet voivat mainostaa tarjoavansa palveluita välttäen samalla monilähetyksen käyttöä etsinnässä. nRF Connect SDK tarjoaa tarvittavat SRP-asiakastoiminnot.

**Transport Control Protocol**

Vaikka nRF Connect SDK:lla on ollut TCP-tuki Zephyrin kautta ([IP-pinon tuetut ominaisuudet](https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/zephyr/connectivity/networking/overview.html#ip-stack-overview)), Thread 1.3 -määritys määrittelee joukon parametreja ja ominaisuuksia, jotka tekevät TCP:stä tehokkaamman rajoitetuissa IEEE 802.15.4 -verkoissa. Thread-pohjaisten TCP-sovellusten parissa työskentelevät käyttäjät voivat ottaa käyttöön vaihtoehtoisen OpenThread-projektiin sisällytetyn TCP-pinon toteutuksen.

Katso [Thread: CLI](https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/samples/openthread/cli/README.html#ot-cli-sample)overlay-tcp.conf-esimerkin määritystiedostosta esimerkki vaihtoehtoisen TCP-toteutuksen ottamisesta käyttöön.

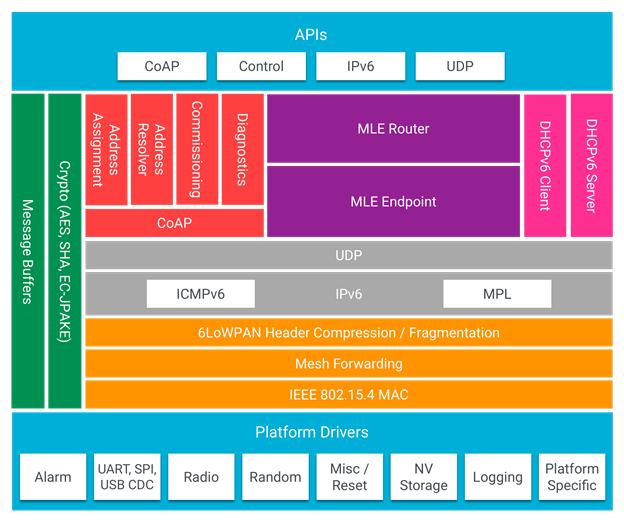
**Thread 1.3 -tuen rajoitukset**

nRF Connect SDK tukee vain kokeellisessa tilassa Thread 1.3 -määrityksen määrittelemää TCP-protokollaa.

**Openthread-arkkitehtuurit (10)**

Käydään läpi ensin yleisellä tasolla OpenThread-pinoarkkitehtuuria (Kuvassa 4 (11)) ja sitten monimutkaisempia toteutuksia

**OpenThread-pinoarkkitehtuuri**

OpenThreadin kannettava luonne ei tee oletuksia alustan ominaisuuksista. OpenThread tarjoaa koukut parannettujen radio- ja salausominaisuuksien käyttöön, mikä vähentää järjestelmävaatimuksia, kuten muistia, koodia ja laskentajaksoja. Tämä voidaan tehdä alustakohtaisesti, mutta silti mahdollisuus käyttää oletusasetuksia vakiokokoonpanoon.

*Kuva 4: OpenThread -arkkitehtuuri*

**Single-chip, multiprotocol (SoC)**

nRF52 (Kuva 5(12))- ja nRF53 (Kuva 6 (13))-sarjan laitteet tukevat useita langattomia tekniikoita, mukaan lukien IEEE 802.15.4 ja Bluetooth® Low Energy (Bluetooth LE).

Yksisiruisessa moniprotokollamallissa sovelluskerros ja OpenThread toimivat samalla prosessorilla.

Tällä suunnittelulla on seuraavat edut:

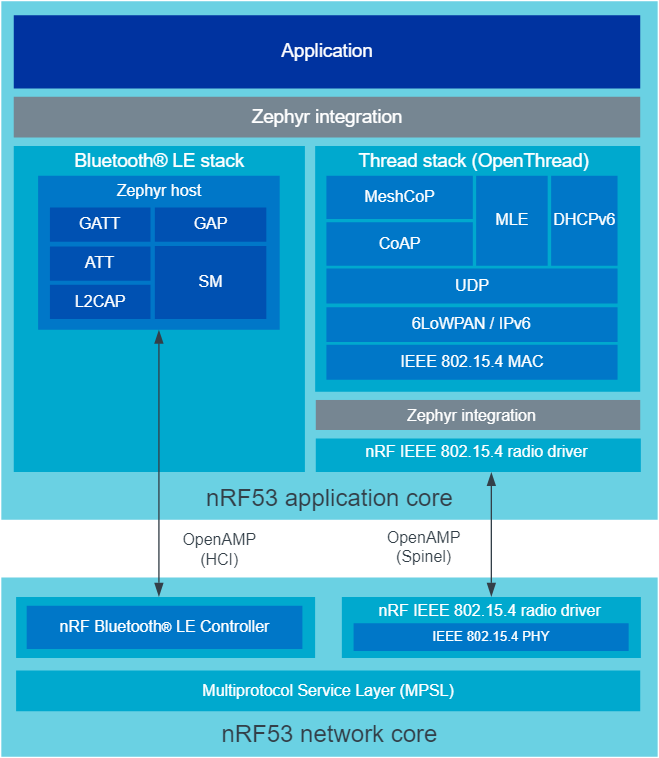
* Se hyödyntää erittäin integroidun SoC:n etuja, mikä johtaa alhaiseen hintaan ja alhaiseen virrankulutukseen.
* Se mahdollistaa Threadin ja Bluetooth LE:n ajamisen samanaikaisesti yhdellä sirulla, mikä vähentää tuoteluettelon kokonaiskustannuksia

Sillä on seuraavat haitat:

* Kuva, joka sisältää kohteen teksti, kuvakaappaus, ohjelmisto, Fontti

  Kuvaus luotu automaattisestiBluetooth LE -toiminta voi heikentää Threadin yhteyksiä, jos sitä ei toteuteta tehokkuutta ajatellen.

*Kuva 5. Multiprotocol Thread ja Bluetooth LE -arkkitehtuuri nRF52 sarjan laitteissa*



*Kuva 6. Multiprotocol Thread ja Bluetooth LE -arkkitehtuuri nRF53 sarjan laitteissa*

**Apuprosessorimallit**

Apuprosessorimalleissa sovellus toimii yhdellä prosessorilla (host prosessori) ja kommunikoi toisen prosessorin kanssa, joka tarjoaa thread-radion. Viestintä tapahtuu sarjayhteyden kautta käyttäen standardoitua host-ohjainprotokollaa (Spinel). Katso [OpenThread-yhteisprosessoriviestintä](https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/protocols/thread/overview/communication.html#ug-thread-communication) saadaksesi lisätietoja viestinnästä.

OpenThread toimii joko radioprosessorilla tai host-prosessorilla riippuen siitä, valitaanko verkkoyhteisprosessorin (NCP) vai radion rinnakkaisprosessorin (RCP) malli. Käytimme projektin matter demossa nRF52840donglea Radio Co-processsor -tilassa. Joten jätetään NCP -tila selvittämättä.

**Radio Co-Processor (RCP)**

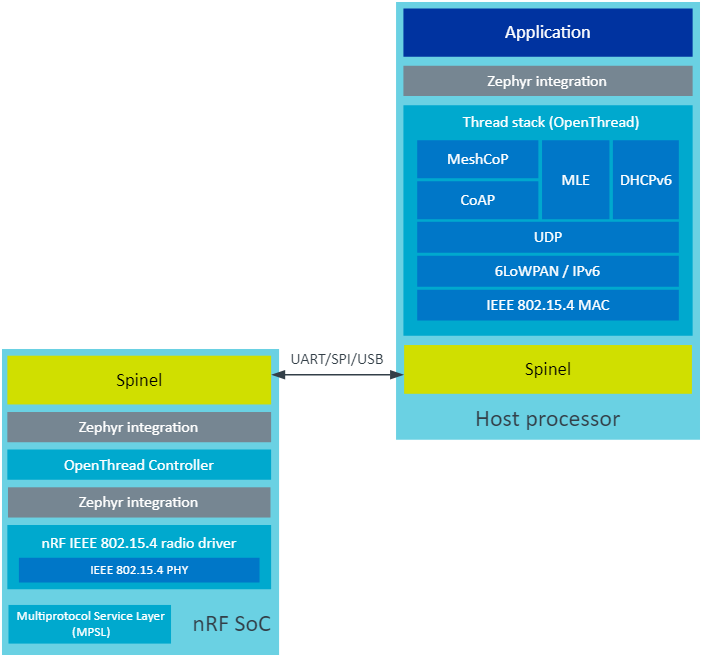
Tämä on muunnelma NCP-suunnittelusta, jossa OpenThreadin ydin toimii host-prosessorilla, ja vain minimaalinen "ohjain" toimii laitteessa, jossa on Thread-radio. Tässä suunnittelussa isäntäprosessori ei yleensä nuku, jotta varmistetaan Thread-verkon luotettavuus.

Tämä malli on hyödyllinen laitteille, jotka ovat vähemmän herkkiä tehonrajoituksille.

Tällä suunnittelulla on seuraavat edut:

* OpenThread voi käyttää tehokkaamman host-prosessorin resursseja.
* Kun valitset kehittyneen ja tehokkaan host-prosessorin, sovellukset voivat olla hyvin monimutkaisia.
* On mahdollista käyttää radion apuprosessoria, joka on vähemmän suorituskykyinen kuin mitä NCP-suunnittelussa tarvitaan, mikä vähentää kustannuksia.

Sillä on seuraavat haitat:

* Host-prosessori tulee herättää jokaisessa vastaanotetussa kehyksessä, vaikka kehys olisi välitettävä naapurilaitteeseen.
* RCP-ratkaisu voi olla vähemmän reagoiva kuin NCP-ratkaisu, koska jokainen kehys tai komento on välitettävä isäntäprosessorille sarjalinkin kautta, kuten kuvasta 7 voidaan todeta (14).

*Kuva 7. Radio Co-Processor-arkkitehtuuri*

### Zigbee mesh

**Zigbee**

Zigbee on kannettava, vähän virtaa käyttävä ohjelmistoverkkoprotokolla, joka tarjoaa yhteyden mesh-verkon kautta IEEE 802.15.4 -radioprotokollaan. Se määrittää myös sovelluskerroksen, joka tarjoaa yhteentoimivuuden kaikkien Zigbee-laitteiden välillä.

**Yleistä**

Zigbee on IEEE 802.15.4 -pohjainen langaton verkkostandardi, jota käytetään pohjimmiltaan kaksisuuntaiseen tiedonsiirtoon anturien ja ohjausjärjestelmien välillä. Zigbee on lyhyen kantaman langaton viestintästandardi, kuten Bluetooth ja Wi-Fi, ja se kattaa 10–100 metrin kantaman.

**Zigbee -kehitysalustat**

EmberZNet on Silicon Labsin Zigbee-toteutus. Se koostuu Zigbee-ydinpinosta, Zigbee Cluster Library -tuesta ja sovelluskehyksestä. Simplicity Studion AppBuilderin avulla kehittäjät voivat helposti luoda Zigbee-sovelluksen, jota voidaan käyttää jollakin Silicon Labsin kehityssarjoista.

Z-Stack on Simple Link™ CC13x2 / CC26x2 Software Development Kitin osa. Tämä komponentti mahdollistaa Zigbee® 3.0 -spesifikaatioihin perustuvien tuotteiden kehittämisen. Z-Stack on TI:n täydellinen ratkaisu sertifioitujen Zigbee 3.0 -ratkaisujen kehittämiseen CC13x2- ja CC26x2-alustoille.

**Tekninen informaatio**

**Zigbee-laitetyypit :**

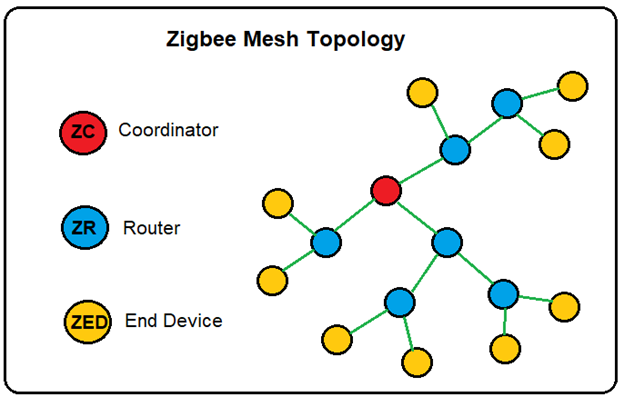
* **Zigbee-koordinaattori (ZC)**
* **Zigbee-reititin (ZR)**
* **Zigbee Endpoint Device (ZED)**

Zigbee-verkostolla on täsmälleen yksi **Zigbee-koordinaattori (ZC)**, joka vastaa verkoston muodostamisesta ja koordinoinnista. Muodostukseen kuuluu PAN-tunnuksen (personal Area Network identifition) valitseminen verkon tunnistamiseksi ja fyysisen radiokanavan määrittäminen käytettäväksi. Koordinaattori vastaa myös verkkoon liittyvien reitittimien ja päätelaitteiden määrittämisestä ja todentamisesta. Koordinaattori on Trust Center ja tallentaa kaikki tärkeät tiedot Zigbee-verkosta, mukaan lukien salausavaimet ja linkkiavaimet.

Zigbee **Router (ZR)** edustaa välisolmuja, jotka auttavat tiedon välittämisessä verkon solmujen välillä. Ne ovat tärkeitä Zigbee-verkon rakentamisessa, jossa paketteja vaihdetaan.

Zigbee-reitittimet parantavat mesh-verkkoa lisäämällä verkon kantamaa (välittämällä paketteja koko mesh-verkossa), lisäämällä vaihdettavien pakettien **luotettavuutta** ja tarjoamalla välineet lisäsolmuille liittyä verkkoon (koordinaattorin avulla). ). Hyvä esimerkki **Zigbee Routerista voi olla älypistoke** (virtalähteenä seinäpistorasia). Tässä laitteessa on loputon päävirtalähde ja se on siksi aina päällä. Se, että tämä älykäs pistoke on täyden toiminnan laite (full-function device (FFD), tekee siitä ihanteellisen liikenteen reitittämiseen ja vahvan ehdokkuuden rajoitetusti toimivien (RFD) päätelaitteiden ohjaimeksi (parent device).

Zigbee**-päätelaite (ZED)** ovat solmuja, jotka on liitetty loogisesti Zigbee-reitittimeen (ZR), ja ne ovat tyypillisesti laitteita, kuten valoja, pistokkeita, antureita, kytkimiä, termostaatteja jne., ja ne kommunikoivat vain Zigbee-reitittimen (vanhemman) kanssa. joihin ne liittyvät. ZED:t ovat usein akkukäyttöisiä laitteita ja nukkuvat suurimman osan ajasta. He eivät voi kommunikoida suoraan muiden Zigbee Endpoint Devices (ZED) -laitteiden kanssa. Tässä mielessä ZED-laitteita pidetään rajoitetusti toimivina laitteina (RFD).

**Zigbee Stack**Kuva, joka sisältää kohteen teksti, numero, Fontti, ohjelmisto

Kuvaus luotu automaattisesti

**Fyysinen kerros**

Fyysinen kerros sisältää fyysisen radio- ja MAC-kerroksen rajapinnan, radion päälle/poisohjauksen, moduloinnin ja demoduloinnin, kanavan valinnan, linkin laadun arvioinnin ja energian havaitsemisen. Zigbee-radiot jakavat 2,4 GHz ISM-kaistan Wifi:n kanssa ja 16 kanavaa (11–26) käyttäen 2 MHz leveä ja 5 MHz kantoaaltoero kanavien välillä.

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti, Sähkönsininen

Kuvaus luotu automaattisesti

**MAC-kerros**

MAC-otsikossa on 2-tavuinen kenttä "Frame Control". Bitti 0–2 ilmaisee kehystyypin. Tyypillisesti kehystyyppejä on neljä:

* **Majakka**, jota käytetään verkkojen skannaukseen
* **Data**, käytetään tiedon siirtämiseen korkeammista kerroksista
* **ACK**, tunnustus
* Kuva, joka sisältää kohteen teksti, kuvakaappaus, numero, Fontti

  Kuvaus luotu automaattisesti**MAC**-komento, MAC-kerroksen ohjauskomennot, kuten MAC-assosiaatiomenettely.

Jokaisen MAC-kehyksen lopussa on kaksi tavua CRC, joita käytetään varmistamaan paketin eheys.

Koska projektissa toteutetussa Matter-verkossa käytetään Thread-verkkoa, niin en jatka tämän enempää Zigbeestä enempää, jos on kiinnostusta tutustua, niin katso seuraava linkki (15) <https://www.blackhillsinfosec.com/understanding-zigbee-and-wireless-mesh-networking/>

Mikäli Matter onnistuu saavuttamaan sille asetetut tavoitteet, tullee Zigbee todennäköisesti häviämään seuraavan 20 vuoden aikana käytöstä, tai ainakin älykoti ohjainympäristöstä.

### Matter

Matter (16) on Connectivity Standards Alliancen (CSA ) luoma uusi avoimen lähdekoodin standardi kodin automaatiolle. Sen tavoitteena on luoda yhteensopiva ja turvallinen älykodin ekosysteemi määrittelemällä yhteinen kieli ja protokolla laitteiden väliseen kommunikaatioon.

Ottamalla käyttöön Matter-standardin eri valmistajat voivat luoda laitteita, jotka ovat täysin yhteensopivia muiden Matter-laitteiden kanssa merkistä riippumatta.

Standardin tarkoituksena on tarjota turvallisempi ja luotettavampi viestintä laitteiden välillä, ja se pyrkii myös vähentämään älykotijärjestelmien luomisen ja ylläpidon monimutkaisuutta ja kustannuksia tarjoamalla yhtenäisen avoimen lähdekoodin standardin, jota kaikki valmistajat voivat vapaasti käyttää.

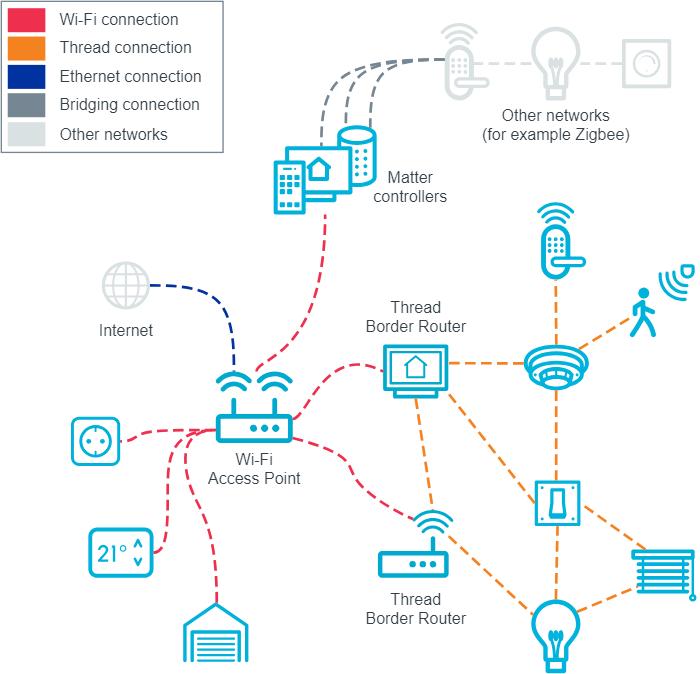
Matter-verkko voi koostua Ethernet-, Wi-Fi- ja Thread-laitteista. Matter yhdistää nämä laitteet toisiinsa paikallisessa Matter-rakennelmassa (infrastruktuurissa), jonka avulla laitteet voivat keskustella toistensa kanssa käyttämällä samaa Matter-sovelluskerrosta huolimatta siitä, että niiden alla käytetään erilaisia verkkoteknologioita. Kaikki viestintä tapahtuu IPv6:n kautta, vaikka Matter-verkko pystyy toimimaan ilman Internetiin liitettyä IPv6-infrastruktuuria, esimerkiksi palomuuriverkossa. Bluetooth LE:tä voidaan käyttää Matter-laitteen käyttöönottoon Matter-verkkoon.

### Matter-verkon topologia

Matter-verkkotopologia (17) viittaa Matter-laitteiden ja IPv6-verkkojen välisten yhteyksien rakenteeseen. Eri IPv6-verkot voivat kommunikoida keskenään keskuskeskittimien kautta, esimerkiksi Thread Border Router tai Wi-Fi Access Point.

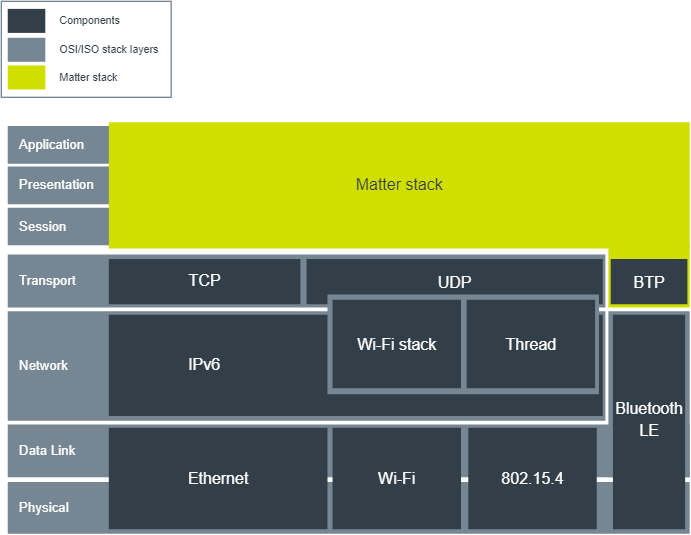
Matter mahdollistaa yhteyden muodostamisen muihin protokolliin perustuviin ulkoisiin verkkoihin, kuten Zigbee, Matter-siltojen kautta. Kuvassa 8 (18) harmaat viivat

Kuva 8. havainnollistaa yleistä Matter-topologiaesimerkkiä kahdella Thread Border Router -reitittimellä, joista yksi on valinnainen Internet-yhteys, ja kolme Wi-Fi-laitetta kuvan yläosassa, jotka toimivat siltakytkiminä muille protokollalaitteille ja Matter-ohjaimina. Eli Kuvassa 8 havainnollistetaan sekä Matter over Thread (oranssit viivat) että Matter over Wi-Fi (punaiset viivat, missä Wi-Fi Access pointista vasemmalla alhaalla olevat pistorasia, lämpötermostaatti ja autotallin oven ohjaus).



*Kuva 8: Esimerkki Matter-verkon topologiasta*

### Matter protokollan yleinen rakenne

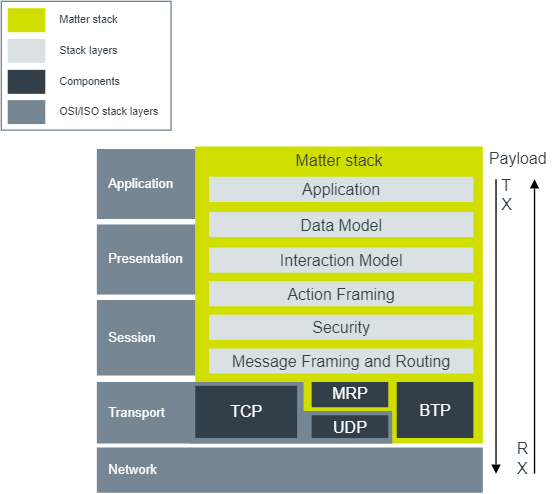
****Matter määrittää sovelluskerroksen (19) IPv6-pohjaisten siirtoprotokollien päälle. Tämä mahdollistaa viestien reitittämisen taustalla olevista fyysisistä ja linkkikerroksista riippumatta, kuvassa 9 (20).

*Kuva 9: Matter protokollan arkkitehtuuri.*

.

### Matter-pinon yleinen rakenne

Matter-sovelluskerros voidaan jakaa useisiin komponentteihin seuraavan kuvan 10 (21) mukaisesti

****

*Kuva 10: Matter-pinon perusrakenne.*

Alimmalla tasollaan materiaalipino on vuorovaikutuksessa kuljetuskerroksen kanssa. Hyötykuorma virtaa alas lähettävän laitteen protokollapinosta ja vastaanottavan laitteen protokollapinosta ylöspäin.

**sovellus**

Sovelluskerros (Application) määrittää lopputuotteen toimintalogiikan. Esimerkiksi valo syttyy ja sammuu painettaessa ennalta määrättyä näppäintä tai se reagoi tiettyyn ääni komentoon.

**Tietomalli**

Tietomallikerros (Data Model) kuvaa Matter-solmun tuetut etätoiminnot käyttämällä attribuuttien, komentojen ja tapahtumien käsitteitä, jotka on ryhmitelty loogisiksi lohkoiksi, joita kutsutaan klustereiksi.

**Vuorovaikutusmalli**

Vuorovaikutusmalli (Interaction Model) määrittää, kuinka näitä tietoja vaihdetaan solmujen välillä vuorovaikutuksen kautta. Vuorovaikutusmallikerros määrittää, mitä vuorovaikutuksia voidaan suorittaa asiakkaan ja palvelinlaitteen välillä.

**Toiminnan kehystys**

Action Framing -kerros muuntaa vuorovaikutukseen kuuluvat viestit vuorovaikutusmallista sarjoitetuiksi binääripaketeiksi.

**Turvallisuus**

Suojauskerros ottaa koodatut kehykset Action Framing -kerroksesta, salaa ne ja liittää ne viestin todennuskoodilla.

**Viestien kehystys ja reititys**

Message Framing and Routing kerros on vastuussa hyötykuorman muodostamisesta pakollisilla ja valinnaisilla otsikkokentillä. Nämä otsikkokentät määrittävät sekä viestin ominaisuudet että sen loogiset reititystiedot.

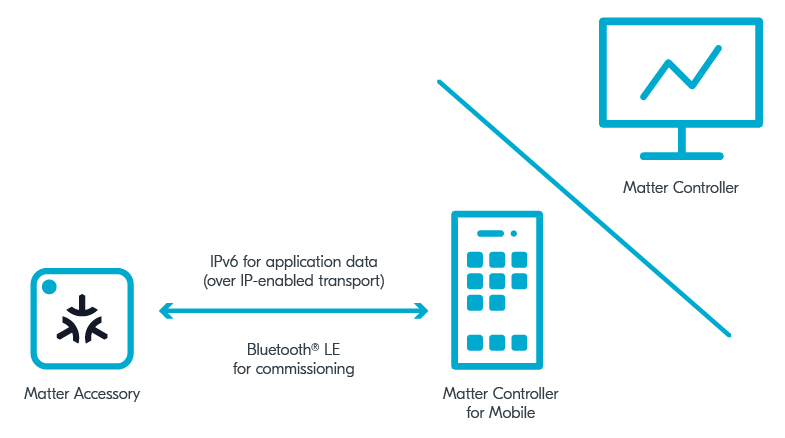
**Kuljetus ja IP-kehystys**

Tämä kerros hallitsee hyötykuorman siirtämistä **IP**-verkon kautta vertaislaitteeseen. Se käyttää joko **TCP** (Transmission Control Protocol) -protokollaa tai **UDP** (User Datagram Protocol) -protokollan ja Matter's Message Reliability Protocol (MRP) -protokollan yhdistelmää.

**MRP** toteuttaa uudelleenlähetykset, vahvistaa toimituksen ja varmistaa, että päällekkäiset viestit hylätään. Käyttöönoton aikana voidaan käyttää Bluetooth LE:n Bluetooth-siirtoprotokollaa tämän kerroksen sijaan.

### Matter-ohjain

Matter-ohjain (22) on solmu Matter-verkossa ja rooli Matter-kehitysympäristössä . Ohjainlaitetta käytetään Matter-lisälaitteen pariliittämiseen ja ohjaamiseen verkon kautta, vuorovaikutuksessa sen kanssa Bluetooth LE:n ja tavallisen IPv6-tiedonsiirron avulla.

Kuvassa 11 (23) näkyy tuetut Matter-ohjaintyypit nRF Connect SDK:ssa.

*Kuva 11: Matterin käyttämiä ohjaintyyppejä*

Matter-ohjain on vuorovaikutuksessa lisälaitteiden kanssa seuraavien protokollien avulla:

* Bluetooth LE käyttöönottoprosessin aikana, verkon valtuustietojen turvalliseen välittämiseen ja lisälaitteen liittämiseen Thread-verkkoon käyttöönoton aikana.  Kun käyttöönotto on valmis, laite on liittynyt IPv6-verkkoon ja on varustettu kaikilla tiedoilla, joita tarvitaan turvalliseen toimintaan Matter-verkossa muiden IPv6-laitteiden kanssa.
* Säännöllinen IPv6-viestintä sen jälkeen, kun lisälaite on liittynyt Thread- tai Wi-Fi-verkkoon - ollakseen vuorovaikutuksessa toistensa kanssa vaihtamalla sovellusviestejä. Esimerkiksi anturin lämpötilamittausten raportoimiseen.

Matter-sovellusten testaamiseen nRF Connect SDK:ssa suosittelemme Linuxin tai macOS:n CHIP-Toolin käyttöä Matter-ohjaimena, joka on yhteensopiva Matterin nRF Connect SDK -toteutuksen kanssa.

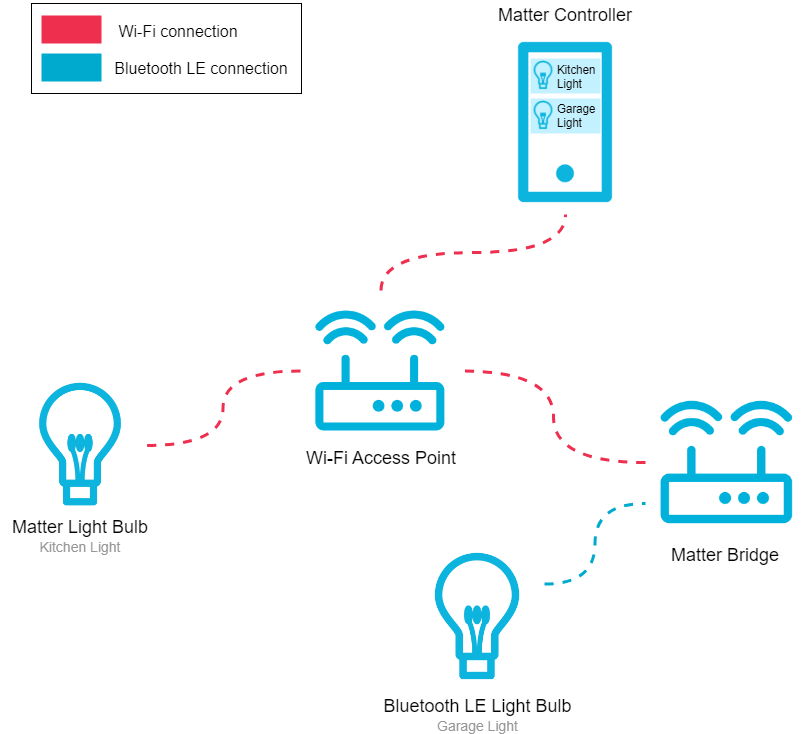
### Matter Bridge

Matter Bridge (24) on laite, joka mahdollistaa non-Matter IoT -laitteiden käytön Matter-rakenteessa yhdessä alkuperäisten Matter-laitteiden kanssa. Tällä tavalla käytettyä non-Matter laitetta kutsutaan *siltalaitteeksi*. Matter-silta paljastaa silloitetut laitteet Matter-solmuille verkon päätepisteinä. Tämä altistuminen on läpinäkyvää, mikä tarkoittaa, että Matter-solmut pystyvät kommunikoimaan non-Matter sillattujen laitteiden kanssa ikään kuin ne olisivat alkuperäisiä Matter-laitteita.

Matter-silta muuntaa Matter-tietomallin tietyn osajoukon ja toiselle protokollalle ominaisen dataesityksen välillä. Sillan tehtävänä ei ole kääntää kaikkea liikennettä verkkokerroksessa käytettyjen protokollien välillä. Matter-spesifikaatio kuvaa vain sen osan siltatoiminnasta, joka liittyy tiukasti Matter-tietomalliin ja kommunikaatioon alkuperäisten Matter-solmujen kanssa.

Matter-määrittelyssä ei voi määritellä:

* Sillattujen laitteiden käyttämät yhteysteknologiat, joten ne voivat käyttää mitä tahansa tekniikkaa (esim. Bluetooth LE tai Zigbee) kunhan myös Matter Bridge tukee sitä.
* Sillan tukemat Matter-laitetyypit.
* Käyttöliittymä, jota sovellus käyttää siltalaitteiden lisäämiseen tai poistamiseen.

Siltatuotteen valmistajan on määriteltävä nämä vaatimukset ja toteutettava tarvittavat toiminnallisuudet.

*Kuva 12: Esimerkki Matter topologiasta, missä sillattu Bluetooth LE -laite*

Kuvassa 12 (25) on esimerkki verkkotopologiasta Matter-ohjaimella, joka käyttää kahta Matter-päätepistettä: Matter-lamppua ja Bluetooth LE -lamppua, joka on sillattu Matter-verkkoon Matter Bridge -laitteella.

**Periaatteita**

Matter-silta käyttää erityistä Matter-päätepistettä, joka täyttää Aggregator-laitetyypin vaatimukset edustaakseen siltausominaisuuksia. Tämä päätepiste käyttää Descriptor-klusteria PartsList-attribuutin kanssa kaikkien päätepisteiden tallentamiseen, jotka edustavat siltaan kytkettyjä laitteita.

Lisäksi silta edustaa jokaista silloitettua laitetta, joka käyttää yhtä tai useampaa Matter-päätepistettä, riippuen silloitetun laitteen tukemista toiminnoista. Jokaisen sillan paljastaman sillallisen laitteen on täytettävä kaikki vastaavaa Matter-laitetyyppiä koskevat vaatimukset. Toisin kuin alkuperäiset Matter-solmut, silloitetussa laitteessa on päätepisteessä käytössä erityinen Bridged Device Basic Information -klusteri, joka tarjoaa perustietoja laitteesta.

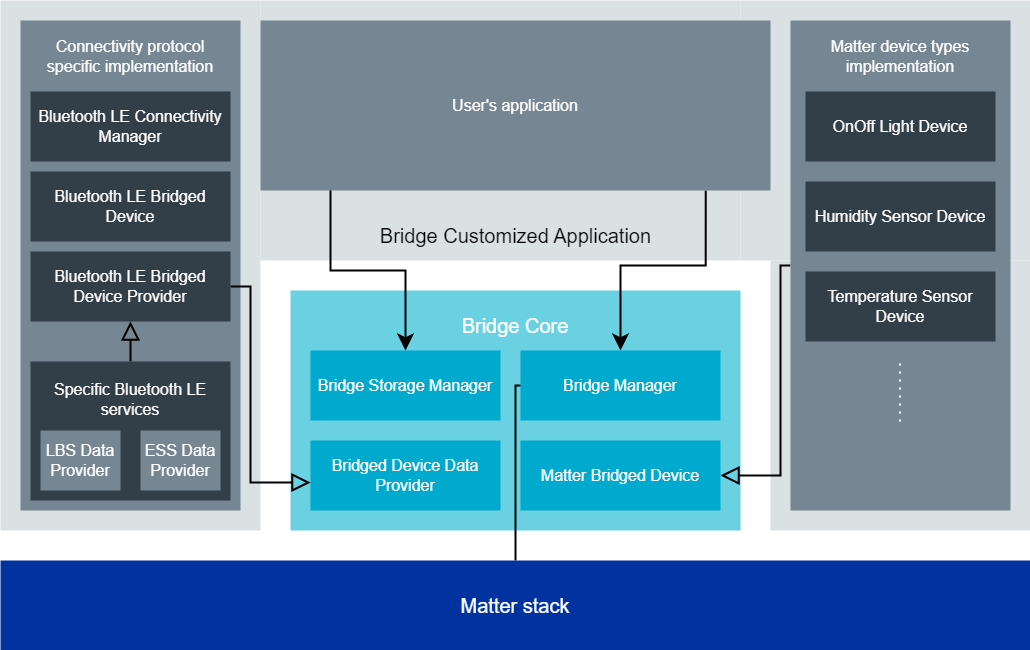
Koska siltalaitteita voidaan lisätä sillalle tai poistaa siitä ajon aikana, sillan paljastamien päätepisteiden luetteloa voidaan päivittää dynaamisesti.

**Toteutus nRF Connect SDK:ssa**

nRF Connect SDK sisältää Matter Bridge -ytimen referenssitoteutuksen, joka sijaitsee hakemistossa samples/matter/common/bridge. Ydintoteutus ei ole riippuvainen tietystä yhteysteknologiasta, ja se tarjoaa perussiltatoiminnot ja yleisen API:n. Ydinosan lisäksi nRF Connect SDK sisältää myös esimerkin [Matter Bridge](https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/applications/matter_bridge/README.html#matter-bridge-app) -sovelluksesta, jota voit käyttää referenssinä oman sovelluksesi luomiseen.

Matter-silta kääntää Matter-tietomallin ja muille protokollille ominaisen datan esityksen. Ei ole mahdollista tarjota yleistä käännöstä Matterin ja kaikkien muiden IoT-protokollien välillä. nRF Connect SDK -sovellus toteuttaa tuen Bluetooth LE -protokollalle, mutta voit myös poistaa Bluetooth LE -tuen käytöstä ja käyttää ydintoteutusosaa luodaksesi sovelluksen, joka yhdistää Matterin mihin tahansa yhteysteknologiaan.

Siltasovellusta voidaan räätälöidä eri tavoin riippuen tuetuista yhteysprotokollasta ja siltaukseen käytetyistä Matter-laitetyypeistä tai laitteen ohjaamiseen käytetystä käyttöliittymästä. Seuraavassa kuvassa näkyy siltaytimen ohjelmistokomponentit ja niiden suhde käyttäjän sovellukseen:



*Kuva 13: Matter Bridge - ohjelmistoarkkitehtuuri*

Kuvassa 13 (26) Bridge-ytimeen sisältyy neljä suurta ohjelmistomoduulia:

* Matter Bridged Device- Abstrakti luokka, joka määrittelee yhteiset toiminnot kaikille Matter-sillallisille laitteille.
* Bridged Device Data Provider- Abstrakti luokka, joka määrittelee yhteiset toiminnot kaikille siltalaitteiden tiedontarjoajille.
* Bridge Manager- Sillan pääkomponentti. Se välittää tietoa silattujen laitteiden, niiden tiedontarjoajien, käyttäjän sovelluksen ja Matter-pinon välillä.
* Bridge Storage Manager- Ohjelmistomoduuli, jolla tiedot siltaan liitetyistä laitteista saadaan pysyviksi. Sen avulla on mahdollista palauttaa sovelluksen tila siltalaitteen uudelleenkäynnistyksen jälkeen.

### Near Field Communications (NFC)

Near Field Communication (NFC) (27) on tekniikka pienten tietomäärien langattomaan siirtoon kahden laitteen välillä. Se käyttää hyvin yksinkertaisia ​​viestintäprotokollia ja mahdollistaa siksi nopeamman yhteyden kuin Bluetooth LE. Koska NFC kuitenkin käyttää magneettista induktiota tiedonsiirron mahdollistamiseen, laitteiden on oltava hyvin lähellä (< 10 cm) yhteyden muodostamiseksi, kuten kuvassa 14 (28).



*Kuva 14: Yksinkertaistettu kuvaus NFCn käytöstä*

**Tunnisteiden ohjelmointi ja lukeminen**

NFC tukee erilaisia ​​toimintatiloja. Lukija/kirjoitustilassa yksi NFC-laitteista (pollauslaite *)* on aktiivinen ja pyytää ja lukee tietoja toisesta laitteesta (tunniste *)* . NFC-tunniste on passiivinen. Se ei synnytä sähkömagneettista kenttää, mutta se moduloi pollauslaitteen synnyttämää kenttää. Tunniste sisältää tietoja, kuten URL-osoitteen, jonka kyselylaite voi lukea ja kirjoittaa.

Suurin osa Nordic Semiconductorin nRF5-sarjan laitteista sisältää oheislaitteen nimeltä NFC-tunniste. Tämä oheislaite toimii NFC-A-tekniikalla ja tukee "kuuntelu"-viestintätilaa. Kuuntelutila tarkoittaa, että tunniste "kuuntelee" (tai odottaa) kyselylaitteita, mutta se ei muodosta yhteyttä aktiivisesti.

**Tagin luku**

Tietojen lukemiseen NFC-tunnisteesta tarvitaan kyselylaite (kutsutaan myös NFC-lukijalaitteeksi). Useimmissa tapauksissa tämä on mobiililaite, jossa on NFC-liitäntä, kuten älypuhelin tai tabletti.

**NFC-tiedonsiirtomuoto (NDEF)**

NFC-viestintä käyttää NDEF (NFC Data Exchange Format) -viestejä tietojen vaihtamiseen. NDEF on binäärimuoto, jota käytetään yleisesti NFC-laitteissa (kuten älypuhelimissa) ja NFC-tunnisteissa.

NDEF:n määrittelemää päätietosäiliötä kutsutaan *NDEF-sanomaksi* , kuva 15 (29). NDEF-viestit koostuvat yhdestä tai useammasta erityyppisestä NDEF-tietueesta . Tyyppi ilmaisee tietueen sisältämän tiedon tyypin, ja sanoman tietuetyyppien sarja määrittää sanoman tyypin. Esimerkiksi URI-sanoma sisältää yhden tietueen, joka koodaa URL-merkkijonon.

***Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti, numero

Kuvaus luotu automaattisesti***NDEF-data on jäsennelty viesteissä. Jokainen viesti koostuu yhdestä tai useammasta tietueesta, jotka koostuvat otsikosta ja tietueen hyötykuormasta. Tietueen otsikko sisältää metatietoja muun muassa hyötykuorman tyypistä ja pituudesta. Tietueen hyötykuorma muodostaa tietueen todellisen sisällön.

*Kuva 15: NDEF-sanoman ja tietueen rakenne*

NDEF-tietueen otsikko koostuu seuraavista kentistä:

| **Ala** | **Pituus** | **Edellytetään** | **Kuvaus** |
| --- | --- | --- | --- |
| Liput ja TNF | 1 tavu | Joo | Katso Liput ja TNF. |
| Tyyppi Pituus | 1 tavu | Joo | Määrittää hyötykuormatyypin kentän pituuden. Pakollinen, mutta voi olla nolla. |
| Hyötykuorman pituus | 1 tai 4 tavua | Joo | Määrittää hyötykuorman pituuden. Joko 1 tavu tai 4 tavua, riippuen SR-lipusta. Pakollinen, mutta voi olla nolla. |
| ID:n pituus | 1 tavu | ei | Pakollinen, jos IL-lippu on asetettu. Määrittää Payload ID -kentän koon. |
| Hyötykuorman tyyppi | muuttuja | ei | Pakollinen, jos Type Length on > 0. Määrittää NDEF-tietueen hyötykuorman tyypin. |
| Hyötykuorman tunnus | muuttuja | ei | Pakollinen, jos IL-lippu on asetettu ja tunnuksen pituus on > 0. Määrittää NDEF-tietueen hyötykuorman tunnuksen. |

Kuva, joka sisältää kohteen teksti, Fontti, kuvakaappaus, numero

Kuvaus luotu automaattisesti**Liput ja TNF**

*Kuva 16: Liput ja TNF-tavu*

Liput ja TNF-tavu (siis yhteensä 8 bittiä), kuten kuvasta 16 (30), näkyy sisältävät seuraavat liput:

**MB (viestin alku) ja ME (viestin loppu):**

Määrittää NDEF-tietueen sijainnin viestissä. MB-lippu asetetaan aktiiviseksi ( siis arvo 1) viestin ensimmäiselle tietueelle. ME-lippu asetetaan viestin viimeiselle tietueelle. Jos se on, viestin ainoa tietue molemmat liput asetetaan.

**CF (Chunk Flag):**

Käytetään lohkotuille hyötykuormille ( joka on jaettu useisiin tietueisiin). Asetetaan kaikkiin muihin paitsi viimeiseen tietueeseen. nRF Connect SDK ei tue lohkoamista.

**SR (Short Record):**

Käytetään määrittämään hyötykuorman koko. Jos lippu on asetettu, hyötykuorma pituus on 1 tavu, muuten sen pituus on 4 tavua. Huomaa nRF Connect SDKssa , NDEF-generaattori tukee vain 4 tavun hyötykuormaa.

**IL (ID:n pituus):**

Ilmaisee, onko otsikossa (Header) ID Length -kenttä, jos asetettu ID Length -kenttä näkyvissä.

**TNF (Type Name Format):**

Määrittää hyötykuormakentän rakenteen ja sen tulkinnan. Seuraavat arvot ovat sallittuja [viittaus tiettyyn tyyppiin].

| **Value** | **Type Name Format** |
| --- | --- |
| 0x00 | Empty |
| 0x01 | NFC Forum well-known type [NFC RTD] |
| 0x02 | Media-type [RFC 2046] |
| 0x03 | Absolute URI [RFC 3986] |
| 0x04 | NFC Forum external type [NFC RTD] |
| 0x05 | Unknown |
| 0x06 | Unchanged |
| 0x07 | Reserved |

**Tietueen hyötykuorma**

Hyötykuorman sisältö on sovelluskohtaista ja liittyy tietueen tyyppiin. Esimerkiksi URI-tietueissa hyötykuorma sisältää sen sivun verkko-osoitteen, joka kyselylaitteen tulee avata.

Huomaa, että NDEF-tietueen hyötykuorma voi sisältää sisäkkäisen NDEF-sanoman. Tämän sisäkkäisen viestin on oltava täysi NDEF-sanoma, joka koostuu yhdestä tai useammasta NDEF-tietueesta, joissa on asianmukaiset MB- ja ME-liput.

**Tag NDEF Exchange Protocol (TNEP)**

Tag NDEF Exchange Protocol (TNEP) on sovellustason protokolla sovellustietoyksiköiden lähettämiseen tai hakemiseen kyselylaitteen (lukija) ja NFC-tunnisteen välillä. Tietoyksiköt ovat NDEF (NFC Data Exchange Format) -viestejä.

Protokolla toimii NDEF-sovelluskerroksen ja tunnisteen välillä. Tunnisteen on käytettävä protokollaa, joka tukee NDEF-luku- ja kirjoitusmenettelyä (eli yksi NFC-foorumitunnistetyypeistä paitsi Type 1 -tagi). Pollauslaitteen on tuettava tätä protokollaa, jotta se voi lukea ja kirjoittaa NDEF-viestejä tunnisteeseen.

TNEP tukee kaikkien NFC Forum -tunnistetyyppien NFC-tunnisteita Type 2 Tagista Type 5 Tagiin. Pollauslaite kommunikoi tunnisteen kanssa käyttämällä protokollaa, joka toimii tietyn tunnistetyypin kanssa.

nRF Connect SDK tarjoaa kirjastot TNEP:n toteuttamiseen sekä kyselylaitteelle että tunnisteelle. Katso [Tag NDEF Exchange Protocol (TNEP)](https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/libraries/nfc/tnep/index.html#lib-nfc-tnep) kaikista toimitetuista kirjastoista.

### ANT/ANT+ protokolla

ANT™ on käytännöllinen langaton sensoriverkkoprotokolla, joka toimii 2,4 GHz ISM-kaistalla. Suunniteltu erittäin alhaiselle teholle, helppokäyttöisyyteen

käytön, tehokkuuden ja skaalautuvuuden ansiosta ANT käsittelee helposti vertais-, tähti-, yhdistetty tähti-, puu- ja kiinteäverkkotopologioita.

ANT tarjoaa luotettavan tietoliikenteen, joustavan ja mukautuvan verkkotoiminnan sekä ylikuulumisen suojan.

ANT-protokollapino on erittäin kompakti, vaatii minimaalisia mikrokontrolleriresursseja ja vähentää huomattavasti järjestelmäkustannuksia.

ANT tarjoaa fyysisten, verkko- ja kuljetusten OSI-kerrosten huolettoman käsittelyn. Lisäksi se sisältää avaimen matalan tason tietoturvaominaisuuksille, jotka muodostavat perustan käyttäjän määrittämille kehittyneille verkkoturvatoteutuksille.

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti, numero

Kuvaus luotu automaattisestiANT varmistaa riittävä käyttäjähallinnan keventäen samalla huomattavasti laskentataakkaa tarjoamalla yksinkertaisen, mutta tehokkaan langattoman verkkoratkaisun.

*Kuva ANT Layers Standard ANT/HOST ja System on Chip Laite*

ANT:n ja isäntäsovelluksen välinen käyttöliittymä on suunniteltu äärimmäistä yksinkertaisuutta ajatellen siten, että ANT voidaan helposti ja nopeasti toteuttaa uusiin laitteisiin ja sovelluksiin.

Langattoman protokollan monimutkaisuuden kapselointi ANT-piirisarjaan vähentää huomattavasti sovelluksen isäntäohjaimen taakkaa, jolloin edullinen 4- tai 8-bittinen mikro-ohjain (MCU) voi luoda ja ylläpitää monimutkaisia langattomia verkkoja.

Tiedonsiirrot voidaan ajoittaa deterministisesti tai tapauskohtaisesti.

Pursketila mahdollistaa suurten tallennettujen tietomäärien tehokkaan siirron PC:lle tai muulle tietokonelaitteelle ja sieltä pois.

Tyypillinen ANT-yhteensopiva laite koostuu sovellusisäntä-MCU:sta, joka on liitetty ANT-moduuliin, piirisarjaan tai siruun.

Isäntä-MCU perustaa ja ylläpitää viestintä istunnon muihin ANT-yhteensopiviin etälaitteisiin yksinkertaisen, kaksisuuntainen, sarjaviestiprotokolla. [ANT Message Protocol and Usage](https://www.thisisant.com/search/eyJsb29zZV9lbmRzIjoieWVzIiwicmVzdWx0X3BhZ2UiOiJzZWFyY2giLCJzZWFyY2hfbW9kZSI6ImFsbCIsImtleXdvcmRzIjoiYW50IHByb3RvY29sIn0).

**ANT:ia, ANT+:aa ja ANT+ Alliancea hallinnoi  the wireless division of Garmin Canada.**

Tässä projektissa ANT jäi vähän sille asteelle, että on olemassa eikä pystytty kokeilemaan, kun ei saanut mitään tietoja ulos ennen kuin sai rekisteröinnin hyväksytettyä.

Kuten yllä mainittiin, kyseinen protokolla on Carminin omaisuutta ja ylläpitämää, jos haluat kokeilla kyseistä protokollaa Nordic Semiconductorin tuotteilla, niin joudut ensin rekisteröitymään käyttäjäksi ja pyytämään Carminilta oikeuksia sisällön lukemiseen. Sitten kun Carmin antaa oikeudet pääset näkemään dokumenttien sisällön.

ANT/ANT+ -protokollat kehitetty lähinnä Carminin urheilukellojen ja -laitteiden sekä (mm. sykemittareiden) niiden antureiden väliseen kommunikointiin. Jos on aktiivinen tunnus voit lukea enemmän esim. [ANT Message Protocol and Usage](https://www.thisisant.com/search/eyJsb29zZV9lbmRzIjoieWVzIiwicmVzdWx0X3BhZ2UiOiJzZWFyY2giLCJzZWFyY2hfbW9kZSI6ImFsbCIsImtleXdvcmRzIjoiYW50IHByb3RvY29sIn0).

# Työskentely projektissa

Yritysprojektissa meni ensimmäiset kaksi viikkoa kehitysalustojen nRF5340dk ja Thingy:53 sekä siihen mitä tietoliikenne protokollia saadaan toimimaan Nordicin sovelluksilla toimimaan . Koska Nordicin ohjelmistotyökalut ja sovellukset olivat jo edellisten projektien kautta kohtuullisen tuttuja ei niiden asennuksiin kulunut varsinaisesti aikaa. Tosin oli hyödyllistä käydä läpi The Bluetooth Low Energy Fundamentals (31), koska siinä käytiin läpi Bluetooth LE toimintoja ja muuta hyödyllistä. Mikä auttoi ymmärtämään Thread verkossa ja Matter verkossa tapahtuvia tunnistautumis proseduureja.

Sitten seuraavilla viikoilla, annetun vastuualueen mukaan (Matter-Hub) yritin löytää kohtuu hintaista Matter hubia tai oikeastaan Thread Border Routeria mikä tulee fyysisesti thread-verkon ja Wi-Fi access pointin rajapintaan. Kuten kuvassa 17 (32) näkyy punaisella ympyrällä sisällä.

Kuva, joka sisältää kohteen diagrammi, teksti, Fontti, viiva

Kuvaus luotu automaattisesti

*Kuva 17: Matter system with Thread Border Router*

Parin kolmen päivän jälkeen totesin, että joitakin kaupallisia järjestelmiä löytyy, mutta ne olivat aika järkyttävän hintaisia. Esimerkiksi pelkän lampun hinta oli noin 20–100 € luokkaa ja niiden ohjainyksiköt noin 70 €:sta ylöspäin. Ja suurin osa vaati jonkun mobiili-pilvipalvelun tai sovelluksen asentamista. Tai sitten järjestelmä toimii wifi, Zigbee-, Thread-, Bluetooth -verkossa ja on Matter yhteensopiva. Mutta joskus tulevaisuudessa.

Eli täytynee rakentaa oma Thread Border Router, onneksi niille löytyy ohjeita, kuinka rakentaa oma. Tosin vaatii Raspberry Pi:n ja jonkun siihen kiinnitettävän radio.

Tutkittuani eri vaihtoehtoja huomasin, että nordicilla oli jo toimiva systeemi, missä Raspberry Pin lisäksi oli nordicin nRF52840 dongle 802.15.4 radiona.

## Thread Border Routerin rakentaminen

nRF Connect SDK:n (33) mukaan Nordic ei tarjoa täydellistä Thread Border Router -ratkaisua. Kehitystarkoituksiin voit käyttää Googlen julkaisemaa [OpenThread Border Routeria](https://openthread.io/guides/border-router) (OTBR), avoimen lähdekoodin Border Router -toteutusta, jonka voit määrittää joko tietokoneellesi Dockerin avulla tai Raspberry Pi:llä.

OPENTHREADin sivuilla (34) OpenThread > Guides > Border Router, kerrotaan ensin mikä yleensä mikä Thread Border Router (TBR) on ja miten se toimii sekä mitä tarvitsen sen rakentamiseen. Tosin ilmeisesti kyseinen sivusto on oikestaan Thread nettikurssi ja vanhahko (vaikka on päivitetty 1.8.2023). Koska siinä käsketään asentaan Raspberry Pihin 32-bit Buster OS. Tästä tulee myöhemmin ongelmia Chip-tool asennuksessa. Tämä ohje on vähän sekava, tosin kyllä tällakin saadaan OTBR asennettua. Mutta myöhemmin tulee ongelmia. -> Käytä mielummin saman sivuston ohjetta (35) OpenThread Border Router Build and Configuration. Ole tarkkana mitä asennat, koska tässä kahden eri alustan asennus ohjeet. Raspberry Pin ja BeagleBone Blackin

Eli jos rakennat OTBRn Rasperry PI + nRF52840 dongle, niin rakenna ohjeen mukaan. Mutta pienin muutoksin, muistikortti mahdollisimman suurikokoinen, ulkoinen SSD, jos on. Raspberry Pi mielellään Pi 4 2G:n muistilla (oma on Raspberry Pi 4 8G). Asenna Ubuntu 22.04 Desktop, helpottaa asennettaessa tarvittavia kirjastoja ja Python-pohjaisia ohjelmia. Kun asennat ohjelmia kannattaa tehdä se kotona ja kiinni Ethernet kaapelilla. Kuten varoittavat kohdassa 3 asenna OTBR. **Varoitus:**Ennen kuin jatkat, varmista, että määritetty laitealustasi on yhdistetty Internetiin Ethernetin kautta. Skripti **bootstrap** poistaa alustan Wi-Fi-liittymän käytöstä ja komentosarja **setup** vaatii Internet-yhteyden useiden pakettien lataamiseen ja asentamiseen.

Jos ei satu olemaan nRF52840 donglea niin voi käyttää myös nRF52840DK korttia, joskin asenna tämä nrf connect SDKn kautta esim. VS Codessa.

Huomaa myös se, että tämä ratkaisu ei toimi esimerkiksi koulun Wi-Fi-verkossa, vaan vaatii RA-guardin deactivointia. **Note:** Make sure that [RA-guard](https://tools.ietf.org/html/rfc6105) is not enabled on your Wi-Fi router! RA-guard is enabled in many enterprise networks for security. But for your personal/home Wi-Fi router, it is usually disabled by default. Consult the network administrator if you are not sure about it.

Koti verkossa yleensä toimii tai sitten puhelimen hotspotin kautta.

**Asenna OTBR:**

1. **Hae ja asenna Raspberry Pin OS image** SD-kortille, ohjeet löytyvät sinisen linkin takaa.

* Jos on tarpeeksi suuri SD-kortti/ SSD niin tarkista/asenna ainakin git, west, cmake, python3, build-essential package, screen, yms.
* Jos on paljon tilaa, esim. SSD-levy, tee nRF Connect SDK manuaalinen asennus linuxille.

1. **Rakenna ja Flash RCP** (Radio Co-Processor), huomaa voi käyttää

* nRF52840 donglea . Dongle asennetaan terminaalin kautta

tai

* nRF52840DK. DK asennetaan SEGGER J-Link OB kautta.

1. **Asenna OTBR**

**Kloonaa OTBR-arkisto**

Kuva, joka sisältää kohteen teksti, Fontti, valkoinen, kuvakaappaus

Kuvaus luotu automaattisesti

**Asenna riippuvuudet**

Kuva, joka sisältää kohteen teksti, Fontti, valkoinen, muotoilu

Kuvaus luotu automaattisesti

**Käännä ja asenna**

Ota Border Routing käyttöön, kun käytät:

Ethernet

Kuva, joka sisältää kohteen teksti, Fontti, valkoinen, muotoilu

Kuvaus luotu automaattisesti

tai

Wi-Fi

Kuva, joka sisältää kohteen teksti, Fontti, valkoinen, kuvakaappaus

Kuvaus luotu automaattisesti

1. **Liitä ja määritä RCP-laite**

Liitä RCP ( nRF5240 dongle tai DK) USB:lla OTBRn (Raspberry Pi)

tarkista RCP-laitteen sarja portti

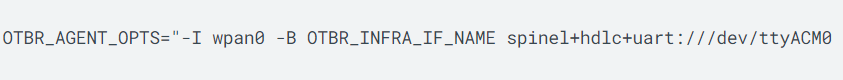
Kuva, joka sisältää kohteen teksti, Fontti, valkoinen, Grafiikka

Kuvaus luotu automaattisesti

tarkista otbr-agentin asetukset

Kuva, joka sisältää kohteen teksti, Fontti, valkoinen, muotoilu

Kuvaus luotu automaattisesti

 määritystiedosto otbr-agent

Kuva, joka sisältää kohteen teksti, Fontti, valkoinen, muotoilu

Kuvaus luotu automaattisesti

1. **Tarkista palvelut**

Tarkista tarvittavat palvelut

Kuva, joka sisältää kohteen teksti, Fontti, valkoinen, muotoilu

Kuvaus luotu automaattisesti

Kuva, joka sisältää kohteen teksti, Fontti, valkoinen, muotoilu

Kuvaus luotu automaattisesti

Kuva, joka sisältää kohteen teksti, Fontti, valkoinen, muotoilu

Kuvaus luotu automaattisesti

Muodosta Thread-verkko

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti

Kuvaus luotu automaattisesti

Odota muutama sekunti, OTBRn pitäisi toimia Leaderinä

Thread verkossa ja säikeen verkkotiedoissa on off-mesh-routable (OMR) -etuliite:

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti, dokumentti

Kuvaus luotu automaattisesti

## Chip-tool asennus Matter over Thread

Koska nRF Connect SDK tukee Matter-pinoa, joka on rakennettu vähän virtaa kuluttavan, 802.15.4-yhteensopivan Thread-verkon päälle ja edellä rakennettu OTBR toimii Thread-verkossa, oli aika luontevaa ajatella Matter over Thread systeemiä.

Kuva, joka sisältää kohteen teksti, kuvakaappaus, diagrammi, Fontti

Kuvaus luotu automaattisestiJotta voit ohjata Matter-lisälaitetta etänä, sinun on määritettävä Matter-ohjain PC:lle tai käyttämällä jotakin yhteensopivasta [kaupallisesta ekosysteemistä](https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/protocols/matter/overview/dev_model.html#ug-matter-overview-dev-model-ecosystems). nRF Connect SDK:n Matter-pino toimii minkä tahansa kaupallisen Matter-ekosysteemin kanssa, kunhan nämä ekosysteemit ovat yhteensopivia virallisen Matter-toteutuksen kanssa (esimerkiksi [Apple Home](https://www.apple.com/home-app/accessories/), [Google Home](https://developers.home.google.com/matter), [Samsung SmartThings](https://support.smartthings.com/hc/en-us/articles/11219700390804-SmartThings-x-Matter-Integration-) tai [Amazon Alexa](https://developer.amazon.com/en-US/docs/alexa/smarthome/matter-support.html)).

*Kuva 11. Matterin käyttämiä ohjaimia*

Matter-ohjain on solmu Matter-verkossa. Ohjainlaitetta käytetään muodostamaan pariliitos Matter-lisälaitteeseen ja sen ohjaamiseen verkon kautta etäyhteydellä, vuorovaikutuksessa Bluetooth LE:n ja tavallisen IPv6-viestinnän avulla, kuva 11 (23).

**Matter over Thread**

Jotta IPv6-viestintä Matter-lisälaitteen kanssa Thread-verkon kautta voidaan ottaa käyttöön, Matter-ohjain tarvitsee Thread Border Routerin. Tämä johtuu siitä, että kuvatuilla Matter-ohjaintyypeillä ei ole 802.15.4 Thread -liitäntää. Border Router siltaa Thread-verkon ohjaimen verkkoliitännän, esimerkiksi Wi-Fi:n kanssa.

Matter over Thread -kehitysympäristössä Matter-ohjain voidaan määrittää jollakin seuraavista tavoista:

* Erillään laitteesta, joka käyttää Thread Border Router -reititintä.
* Samassa laitteessa kuin Thread Border Router.

Koska samassa laitteessa oleva Matter-ohjain (chip-tool) vaatii aikasuuren tilan noin 14 GB asennettiin se erilliseen linux läppäriin.

**Matter over Thread: OTBR ja ohjaimen määrittäminen eri laitteissa**

Suositeltu tapa Matter over Threadille on ajaa TBR ja Matter-ohjain erillisissä laitteissa (36). Kun Raspberry Pi:hen on asennettu OTBR, tämä lähestymistapa tukee useimpia Matterin käyttämiä toimintoja. Se mahdollistaa esimerkiksi mobiiliohjaimen käytön tarjoamalla yhteyden Wi-Fi-verkon ja Thread-verkon välille.

Tällaisissa asetuksissa Raspberry Pi käyttää OTBR-reititintä, joka tarjoaa viestinnän Thread-solmujen ja Matter-ohjaimen välillä. Ohjain voidaan asentaa Linux-PC:lle. Sekä OTBRn että Matter-ohjaimen on tuettava IPv6-viestintää runkoverkon, esimerkiksi Wi-Fi:n tai Ethernetin, kautta.

Tässä PC-pohjaisessa asennuksessa Matter-ohjain asennetaan Linux-käyttöjärjestelmää käyttävään PC:hen, ja käytetään erillistä Wi-Fi-access point ja CHIP Tool Matter -ohjainta. Tätä asennusta suositellaan käytettäväksi nordicin nRF-tuotteiden kanssa, kuva 18(37).

Kuva, joka sisältää kohteen diagrammi, teksti, Fontti, viiva

Kuvaus luotu automaattisesti

*Kuva 18: Matter over Thread, Chip-tool asennus Linux -PC*

**Laitteiston vähimmäis vaatimukset**

Tarvitset seuraavan laitteiston käyttääksesi tätä asetusta:

* 1x PC Ubuntulla (20.04 tai uudempi)
* 1x Raspberry Pi Model 3B+ tai uudempi (sekä SD-kortti, jossa on vähintään 8 Gt muistia)
* 1x Wi-Fi Access Point, joka tukee IPv6:ta (ilman reitittimen IPv6 Router Advertisement Guard -toimintoa)
* 1x nRF52840 DK tai nRF52840 Dongle - Radio Co-Processor (RCP) -laitteelle
* 1x yhteensopiva Nordic Semiconductor's DK - Matter-lisävarustelaitteelle (yhteensopiva ja ohjelmoitu jonkin [Matter-näytteen](https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/samples/matter.html#matter-samples) kanssa )

**Ympäristön asetukset**

1. Ohjelmoi kehitysalusta Matter näytteellä
2. Määritä OTBR
3. Määritä CHIP-tool Linuxille

kopioi gitillä Matter-koodi

**git clone --recurse-submodules http://github.com/project-chip/connectedhomeip.git**

Jos on jo vanha versio Matter-koodista, päivitä sen alihakemistot

**git submodule update --init**

edellytysten asentaminen Linuxiin (Ubuntu/Debian)

**sudo apt-get install git gcc g++ pkg-config libssl-dev libdbus-1-dev \**

**libglib2.0-dev libavahi-client-dev ninja-build python3-venv python3-dev \**

**python3-pip unzip libgirepository1.0-dev libcairo2-dev libreadline-dev**

UI Builds

jos rakennat -with-ui muunnelman, asenna myös SDL2:

**sudo apt-get install libsdl2-dev**

**CHIP-tool build ja käyttäminen**

Ennen kuin voit käyttää CHIP-työkalua, sinun on käännettävä se lähdekoodista Linuxissa (amd64/aarch64). Jos haluat käyttää sitä Raspberry Pi:llä, sen on käytettävä 64-bittistä käyttöjärjestelmää.

Varmista yhteensopivuus rakentamalla aina CHIP-työkalu ja Matter-laite samasta arkiston versiosta connectedhomeip. Siis tarkoittaa nRF Connect SDK, jos näyte tai koodi esim. SDK versiosta 2.4.0 esimerkki koodi pitää hakea C:\ncs\v2.4.0\nrf\applications\matter\_weather\_station.

**CHIP-tool rakentaminen (building CHIP-tool)**

**$ cd connecthomeip**

**$ ./scripts/examples/gn\_build\_example.sh examples/chip-tool BUILD\_PATH**

**CHIP-tool Käyttäminen**

**$ cd /connecthomeip/BUILD\_PATH**

**$ ./chip-tool**

**Määritä Matter-laite**

Kytke Matter-laite USB-johdolla johonkin laitteeseen esim. win10 PC ja lue sen UART, PuTTy-ohjelmalla, screen linux-terminaalilla tai nRF Connect for Desktop ja sieltä Serial Terminal -> valitse laite ja COM-portti.

Laitteen käynnistyttyä voit lukea laitteen arvo uart-päätteestä

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti, dokumentti

Kuvaus luotu automaattisesti



**Käyttöönotto Thread-verkkoon Bluetooth LE:n kautta**

Yhdistä laite Thread-verkkoon seuraavalla tavalla:

**$ ./chip-tool pairing ble-thread <node\_id> hex:<operational\_dataset>** **<pin\_code> <discriminator>**

esim. yllä olevaa lokitiedostoa käyttämällä saadaan

**./chip-tool pairing ble-thread 1 hex:0e08000000000001000035060004001fffe00708fde6ab956770268d0c0402**

**a0f7f8051000112233445566778899aabbccddeeff030e4f70656e54687265616444656d6f0410445f2b5ca6f2a93**

**a55ce570a70efeecb000300000f0208111111112222222201021234 20202021 3840**

missä

<node\_id> **1**, eli voi olla mikä tahansa käyttäjän valitsema luku, yleensä 1 seuraava laite 2 jne.

<operational\_dataset> saadaan thread-verkosta OTBR **$ sudo ot-ctl dataset active** **-x**, mistä saadaan hexaluku

<pin\_code> **20202021** uart lokipäätteeltä Setup Pin Code

<discriminator> **3840** saadaan uart lokipäätteltä Setup Discriminator

Jos laite yhdistyy thread-verkkoon siitä, muodostuu pitkä listaus mistä ei juuri huomaa, että paritus onnistui. Kuitenkin jos ei ilmoita virhettä yleensä laitteen paritus on onnistunut. Laitan lokitiedoston liitteeksi.

**[1697719833.939256][3686:3686] CHIP:DL: NVS set: chip-counters/total-operational-hours = 0 (0x0)**

**[1697719833.939266][3686:3686] CHIP:DL: Inet Layer shutdown**

**[1697719833.939274][3686:3686] CHIP:DL: BLE shutdown**

**[1697719833.940000][3686:3686] CHIP:DL: System Layer shutdown**

**m@m-Lenovo:~/connectedhomeip/BUILD\_PATH$**

Tässä tapauksessa ei virhettä ja paritus onnistui

Tämän jälkeen voidaan Matter-ohjaimella hakea Matter-laitteelta tietoja, sammuttaa tai sytyttää valoja, aukaista ovia yms.

Tässä tapauksessa Matter-laitteena on Thingy:53, johon on asennettu matter\_weather\_station.

Joten haetaanpa vaikka lämpötila laitteelta.

Eli jos olet oikeassa hakemistossa

**$ cd ~/connectedhomeip/BUILD\_PATH$**

**$ ./chip-tool temperaturemeasurement read measured-value 1 1**

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Fontti

Kuvaus luotu automaattisestitästä kyselystä saatiin seuraava arvo :



Data = 2619 eli 26,19°C tämä oli kuitenkin yli 3 ° pielessä, mutta saatiin kuitenkin luettua arvo Matter -ohjaimella, mikä tässä oli tavoitteena.

Liitteenä Thingy:53 uart lokit, missä uart logit:

* Laitteen käynnistyessä.
* Onnistuneessa parituksessa.
* Lämpötilan hakutapahtumassa.

sekä

Linux chip-tool logit :

* Onnistunut paritus matter-laitteen kanssa (Thingy:53)
* Onnistunut lämpötilan haku Thingy:53:lta.

# Projektin tulokset

Projektin tulos oli onnistunut, vaikka ehkä visuaalisesti ei mitenkään näyttävä.

Onnistuin rakentamaan Open Thread Border Routerin (Raspberry Pi 4 ja nRF52840dongle Radio Co-Processor) RCP ja lisäämään Matter-laitteen laitteen Thread-verkkoon käytten Bluetooth LE paritusta sekä CHIP-toolilla Matter over Thread hakemaan Thingy:53:lta lämpötila tiedon.

Tosin vanhat ja puutteelliset ohjeet aiheuttivat ongelmia miltei koko ajan. Esimerkiksi koulun Wi-Fi verkossa ei pystynyt parittamaan laitteita, kun se esti BLE viestiin vastaamisen. Kotiverkossa ei ongelmaa eikä käyttämällä puhelimen hotspottia. Tietenkin voidaan sanoa, että lue tarkasti ohjeet. Kyllähän se siellä OPENTHREAD sivustolla luki, mutta enpä huomannut. Koska en ollut ennen törmännyt kyseiseen ongelmaan, siis **IPv6 Router Advertisement Guard.**

# Yhteenveto

Laitteisto ongelmia milloin VS Code hukkaa kaikkien laitteiden tiedot tai ei suostu flash:ään ohjelmaa kortille tai sitten itsepintaisesti herjaa, että kirjaudu Githubiin, vaikka kaikki git-liitännäiset oli poistettu VS Codesta. En tiedä on mikä maksaa, mutta ollut koko ajan jotakin ongelmaa liittyen Nordicin sovellus ohjelmiin kaikkien kolmen projektin aikana. Tai sitten asennukset jäävät vajaiksi, eli asennusohjelma ei asenna kaikkien ohjelmien alihakemistoja tai ne jäävät tyhjiksi.

Sitten nordicilla on kattavat dokumentit, mutta siellä olevaa vanhaa tai virheellistä tietoa ei ole poistettu. Kumminkin kun puhutaan esimerkiksi Matter-sovelluksista tai sen asennusohjeista ne pitäisi olla ajan tasalla. Puhumattakaan sekavista asennusohjeista, kun niissä on kaikkien käyttöjärjestelmien asennusohjeet samalla sivulla sekaisin samassa kasassa.

Esimerkiksi vanhan ohjeen vuoksi jouduin asentamaan Raspberry Pin OS:n muutaman kerran. Tämä on tätä, mutta näissä hukkaa paljon aikaa. Ja esimerkiksi kun asennetaan connecthomeip ohjelmistoa, joka on miltei 14 GB paketti ja kun ohjeissa puhutaan, että mahtuu 8 GB SD-kortille, niin tämmöiset pitäisi saada pois. Ja kun asennat kaikki tarvittavat puhtaaseen OS:n siinä, tuhrautuu miltei 5 tuntia aikaa.

Jää vähän huono kokemus Nordicin laitteista kokonaisuudessaan. Varsinkin tässä 3 projektissa ohjeet, joilla pitäisi päästä alkuun Matter käytössä, ovat todella epämääräisiä ja osin virheellisiä.

Jos vertaa kilpailijan esimerkiksi Silicon Labsin ESB32 tuotteelle tekemä vastaava demo. Katso

<https://www.youtube.com/watch?v=mVoL8H65qss>, tässä videossa on todella helppo yksityiskohtainen asennus verrattuna Nordicin vastaavaan. Ja valmiiksi konffatut asennusohjelmat …

## Mitä tekisin toisin

No aluksi hommasin puhtaan läppärin, mille voisi asentaa puhdas Ubuntu Linux. Sitten asentaisin nuo Nordicin tarvitsemat sovellus ohjelmat, linux-versiot. Ja yritän saada mahdollisimman yksinkertaisella ohjelmalla toimimaan tuo matter-verkko ja sitten koittaisin lisätä useampia matter-laitteita samaan verkkoon.

Ja linux pohjaisen chip-toolin tilalla koittaisin tehdä Android studiolla android pohjaisen matter-controllerin puhelimelle. Vaikka ei ole ehkä niin monipuolinen kuin linux chip-tool se on kuitenkin helpompi käyttää ja visuaalisempi, kuin vastaava linux pohjainen.

lähteet

1. Nordic Semiconductor 2023. nRF5340 DK. Hakupäivä 16.10.2023. <https://www.nordicsemi.com/Products/Development-hardware/nRF5340-DK/Download>.

2. Nordic Semiconductor 2023. Nordic Thingy:53. Hakupäivä 16.10.2023. [https://www.nordicsemi.com/Products/Development-hardware/Nordic-Thingy-53](https://www.nordicsemi.com/Products/Development-hardware/Nordic-Thingy-53%20) .

3. www.segger.com 2023. J-Link OB , The on-board debug probe. Hakupäivä 16.10.2023. <https://www.segger.com/products/debug-probes/j-link/models/j-link-ob>.

4. Nordic Semiconductor 2023. nRF Sniffer for 802.15.4. Hakupäivä 16.10.2023. <https://www.nordicsemi.com/Products/Development-tools/nRF-Sniffer-for-802154.>

5.Wireshark.org 2023. The world’s most popular network protocol analyzer. Hakupäivä 16.10.2023. [https://www.wireshark.org/.](https://www.wireshark.org/.%20)

6. Nordic Semiconductor 2023. nRF5340DK, kuva. Hakupäivä 17.10.2023.

<https://circuitstate.com/wp-content/uploads/2021/05/nRF5340-DK-Development-Board-Annotated-1.jpg>.

7. Nordic Semiconductor 2023. Bluetooth LE – protokollapino. Hakupäivä 17.10.2023. <https://academy.nordicsemi.com/courses/bluetooth-low-energy-fundamentals/lessons/lesson-1-bluetooth-low-energy-introduction/topic/what-is-bluetooth-le/>.

8. Nordic Semiconductor 2023. Bluetooth mesh concepts. Hakupäivä 18.10.2023. <https://developer.nordicsemi.com/nRF_Connect_SDK/doc/2.4.0/nrf/protocols/bt_mesh/concepts.html>.

9. Nordic Semiconductor 2023. Thread. Hakupäivä 18.10.2023. <https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/protocols/thread/index.html>.

10. Nordic Semiconductor 2023. OpenThread -arkkitehtuurit. Hakupäivä 18.10.2023. <https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/protocols/thread/overview/architectures.html>.

11. Nordic Semiconductor 2023. OpenThread -arkkitehtuurit, kuva *OpenThread architecture adapted from*[*openthread.io*](https://openthread.io/platforms#system_architecture). Hakupäivä 18.10.2023. <https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/_images/ot-arch_2x.png>.

12. Nordic Semiconductor 2023. OpenThread -arkkitehtuurit, kuva Multiprotocol Thread and Bluetooth LE architecture on nRF52 Series devices. Hakupäivä 18.10.2023. <https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/_images/thread_platform_design_multi.svg>.

13. Nordic Semiconductor 2023. OpenThread -arkkitehtuurit, kuva Multiprotocol Thread and Bluetooth LE architecture on nRF53 Series devices. Hakupäivä 18.10.2023. <https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/_images/thread_platform_design_nRF53_multi.svg>.

14. Nordic Semiconductor 2023. OpenThread -arkkitehtuurit, kuva *Radio co-processor architecture.* Hakupäivä 18.10.2023. <https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/_images/thread_platform_design_rcp.svg>.

15. Felch, Ray 27.08.2021. // Blog: Understanding Zigbee and wireless mesh Networking. Hakupäivä 18.10.2023. <https://www.blackhillsinfosec.com/understanding-zigbee-and-wireless-mesh-networking>

16. CSA-iot.org 2023. matter, The Foundation for Connected Things. Hakupäivä 20.10.2023. <https://csa-iot.org/all-solutions/matter/>.

17. Nordic Semiconductor 2023. Matter network topology and concepts. Hakupäivä 20.10.2023. <https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/protocols/matter/overview/network_topologies.html>.

18. Nordic Semiconductor 2023. Matter network topology, kuva *Matter topology example.* Hakupäivä 20.10.2023. <https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/_images/matter_network_topologies.svg>.

19. Nordic Semiconductor 2023. Matter architecture. Hakupäivä 22.10.2023. <https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/protocols/matter/overview/architecture.html>.

20. Nordic Semiconductor 2023. Matter architecture, kuva *Matter architecture overview*. Hakupäivä 22.10.2023. <https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/_images/matter_architecture.svg>.

21. Nordic Semiconductor 2023. Stack architecture , kuva *Matter stack overview* . Hakupäivä 22.10.2023. <https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/_images/matter_components.svg>.

22. Nordic Semiconductor 2023. Matter network topology and concepts, matter controller. Hakupäivä 22.10.2023. <https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/protocols/matter/overview/network_topologies.html>.

23. Nordic Semiconductor 2023. Matter network topology and concepts, matter controller, kuva *Controllers used by Matter*. Hakupäivä 22.10.2023. <https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/_images/matter_setup_controllers_generic.png>.

24. Nordic Semiconductor 2023. Matter Bridge . Hakupäivä 22.10.2023. <https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/protocols/matter/overview/bridge.html>.

25. Nordic Semiconductor 2023. Matter Bridge, kuva *Example Matter topology using a bridged Bluetooth LE device* . Hakupäivä 22.10.2023. <https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/_images/matter_bridge_topology.svg>.

26. Nordic Semiconductor 2023. Matter Bridge, kuva *Matter bridge software architecture.* Hakupäivä 22.10.2023. <https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/_images/matter_bridge_architecture.svg>.

27. NFC FORUM 2023. Technical Overview. Hakupäivä 23.10.2023. <https://nfc-forum.org/learn/nfc-technology/>.

28. Nordic Semiconductor 2023. Near Field Communication (NFC), kuva *Simplified overview of how NFC can be used.* Hakupäivä 23.10.2023. <https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/_images/nfc_overview.svg>.

29. Nordic Semiconductor 2023. Near Field Communication (NFC), NDEF message and record format, kuva *Structure of an NDEF message and record*. Hakupäivä 23.10.2023.

<https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/_images/ndef_msg.svg>.

30. Nordic Semiconductor 2023. Near Field Communication (NFC), NDEF message and record format, Flags and TNF , kuva *Flags and TNF byte*. Hakupäivä 23.10.2023.

<https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/_images/ndef_header_flags.svg>.

31. Nordic Semiconductor 2023. The Bluetooth Low Energy Fundamentals course. Hakupäivä 19.10.2023. <https://academy.nordicsemi.com/courses/bluetooth-low-energy-fundamentals/>.

32. Nordic Semiconductor 2023. Matter system, kuva. Hakupäivä 19.10.2023. <https://developer.nordicsemi.com/nRF_Connect_SDK/doc/1.9.99-dev1/nrf/_images/matter_otbr_controller_same_device.svg>

33. Nordic Semiconductor 2023. Thread tools / Tread Border Router. Hakupäivä 19.10.2023.

<https://developer.nordicsemi.com/nRF_Connect_SDK/doc/2.4.0/nrf/protocols/thread/tools.html#thread-border-route>.

34. OPENTHREAD 2023. Thread Border Router - Bidirectional IPv6 Connectivity and DNS-Based Service Discovery. Hakupäivä 19.10.2023. <https://openthread.io/codelabs/openthread-border-router#1>

35. OPENTHREAD 2023. OpenThread Border Router Build and Configuration. Hakupäivä 19.10.2023. <https://openthread.io/guides/border-router/build>.

36. Nordic Semiconductor 2023. Matter over Thread: Configuring Border Router and Linux OS controller on separate devices. Hakupäivä 23.10.2023.

<https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/protocols/matter/getting_started/testing/thread_separate_otbr_linux_macos.html>.

37. Nordic Semiconductor 2023. Matter over Thread: Configuring Border Router and Linux OS controller on separate devices, kuva. Hakupäivä 23.10.2023.

<https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/_images/matter_otbr_controller_separate_pc.svg>.

**Thingy:53 ja chip-tool loKeja** Liite 1

Thingy:53:n uart loki käynnistyksessä :



Chip-tool-loki Linux PC, kun onnistunut paritus Matter-laitteen kanssa:



Thingy:53 loki onnistuneessa parituksessa:



Chip -tool loki, kun haettu onnistuneesti lämpötilatietoa Thingy:53:lta



Thingy:53:n loki lämpötilan haku

