

# Sähköpaja

## Radiomoduli nRF24L01+

### *Työn tavoite*

Työssä tutustutaan nRF24L01+ piiriin perustuvan radiomodulin toimintaan. Radiomodulia käytetään mikrokontrollerien välisessä viestinnässä. Työn esimerkkejä voidaan käyttää pohjana omien viestintäsovellusten kehittämiseen.

### *Tarvikkeet*

- Kaksi tai useampi Arduino-IDE:llä ohjelmoitavaa alustaa. Esimerkiksi Arduino Uno tai Teensy. Erot vaikuttavat lähinnä SPI-väylän paikallistamiseen.
- Teensy 2.0 alustalla tarvitaan myös 3.3V regulaattori.
- Yksi radiomoduli nRF24L01+ jokaista alustaa kohden.
- Kytkejäjohtoja.g
- USB-johtoja alustojen kytkemiseen.

Useamman kuin kahden noodin toimintaan kannattaa rekrytoida muiden ryhmien apua; jos kaikki tekevät samaa työtä ja käyttävät samaa taajuutta, piirit luonnollisesti keskustelevat keskenään.

### *1. Toiminta*

Radiomodulit kytketään SPI-väylällä mikrokontrolleriin. Peruspinnien (SCK/MOSI/MISO) lisäksi radiomoduli käyttää kahta signaalia piirin ohjaamiseen; ChipEnable (CE) käytetään modulin toiminnan kytkemiseen päälle ja pois. ChipSelectNegative (CSN) käytetään SlaveSelect signaalina piirin viestinnän ohjaamiseen.

Käytetyn radiomodulin radiopiiri on melko monimutkainen ja käyttää omia mekanismejaan viestien välittämiseen. Olennaisimpia näistä ovat:

Radiopiiri toteuttaa viestien automaattisen kuittauksen ja uudelleenlähetyksen. Tämä on näppärä ominaisuus ja säästää paljon vaivaa. Tätä tulisi käyttää ensisijaisesti oman kuittausmenettelyn toteuttamisen sijaan. Käyttäjälle tämä näkyy RF24::write() kutsua tehdessä paluuarvossa; jos paluuarvo on true, lähetys kuitattiin. Jos arvo on false, viesti ei useammasta yrityksestäkään huolimatta mennyt läpi.

Radiopiiri puskuroi lähetyksiä ja vastaanottoja. Käyttäjän ei siis tarvitse kytätä radiopiiriä jatkuvasti. Puskureihin tallentuu myös miltä kuunnellulta osoitteelta (tai ”putkelta”, pipe) viesti on tullut. Viestin saatavuuden voi testata RF24::available() kutsulla, joka kertoo missä putkesta viesti on. Piiri voi siis vastaanottaa vain tunnetuilta osoitteilta (putkilta) ja nämä kuuluu asettaa etukäteen RF24::openReadingPipe() ja RF24::openWritingPipe() kutsuilla.

Radiopiiri voi olla joko lähetyk- tai kuuntelutilassa. Tilat tulee vaihtaa manuaalisesti; RF24::startListening() ja RF24::stopListening(). Jos energiatahokkuus on tärkeää, radiota tulisi pitää ”standby” tilassa mahdollisimman paljon. Tämä kuitenkin vaatii tarkan suunnittelun että vastaanottajat ovat aina kuuntelutilassa kun lähettäjät haluaa lähettää.

Radiopiirillä on 128 mahdollista radiokanavaa. Oletuksena käytetään kanavaa 76. Jos useampia käyttäjiä käyttää samaa radiopiiriä, tulee tahattoman ristiviestinnän estämiseksi käyttää eri kanavaa.

Paras, joskin monisanaisin, referenssi piiriin toimintaan on sen valmistajan piiristä tekemä datalehti:

[https://www.nordicsemi.com/kor/content/download/2726/34069/file/nRF24L01P\\_Product\\_Specificatio\\_n\\_1\\_0.pdf](https://www.nordicsemi.com/kor/content/download/2726/34069/file/nRF24L01P_Product_Specificatio_n_1_0.pdf)

[https://www.nordicsemi.com/kor/node\\_176/2.4GHz-RF/nRF24L01P](https://www.nordicsemi.com/kor/node_176/2.4GHz-RF/nRF24L01P)

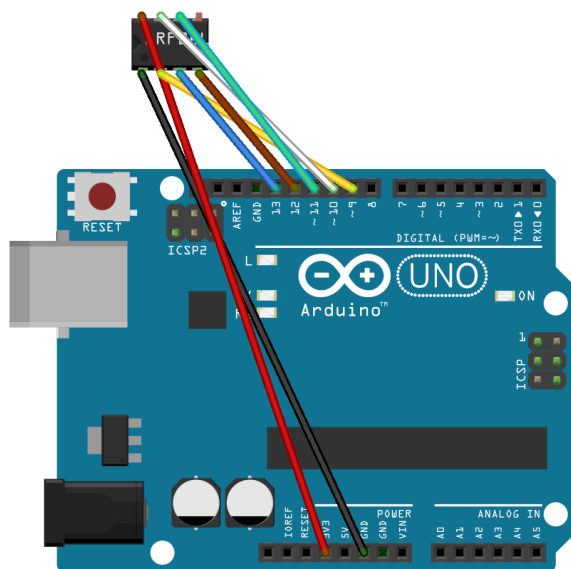
Ohjelmistot on toteutettu RF24 -kirjastolla, joka löytyy osoitteesta:

<https://github.com/maniacbug/RF24/>

## 2. Kytkeminen

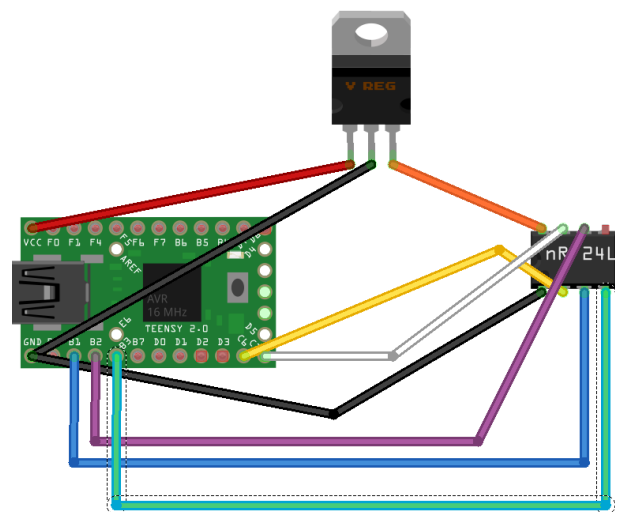
Työssä käytetään Arduino-alustojen SPI-väylää. Väylä on kytketty alustan johonkin pinneihin. Arduino Unon tapauksessa pinnit ovat 11-13. Pinnit on esitelty taulukossa 1 alla. Teensyn kytkemisessä ongelmana on että Teensy 2.0:ssa ei ole 3.3V regulaattoria.

- Arduino: <http://arduino.cc/en/Reference/SPI>
- Teensy: [http://www.pjrc.com/teensy/td\\_libs\\_SPI.html](http://www.pjrc.com/teensy/td_libs_SPI.html)



Made with Fritzing.org

Kuva 1: Radiomodulin kytkentä Arduino Unoon.



Made with Fritzing.org

Kuva 2: Radiomodulin kytkentä Teensy 2.0 alustaan.

Radiomoduli		Teensy	Arduino	
Nimi	Pinni	Pinni	Nimi	Kuvaus
GND	1	GND	GND	Maa
VCC	2	regulaattorilta	3.3V	Käyttöjännite. Tärkeää! Oltava 3.3V!
CE	3	9 (PC6)	9	ChipEnable, radiotoiminta päälle/pois.
CSN	4	10 (PC7)	10	ChipSelectNegative, valitsee piirin.
SCK	5	1 (PB1)	13	SerialClock, SPI-kello-signaali.
MOSI	6	2 (PB2)	11	MasterOut SlaveIn, SPI-datasignaali.
MISO	7	3 (PB3)	12	MasterIn SlaveOut, SPI-datasignaali.
IRQ	8			Keskeytyssignaali, ei käytössä esimerkeissä.

*Taulukko 1: Radiomodulin ja Arduinon pinnit.*

Kuvassa 1 on esitetty miten radiomodulin kytkentä tehdään Arduino Uno alustalla. Kuvassa 2 on kytkentä Teensy 2.0 alustalla, mukaanlukien 3.3V regulaattori. Punainen ja musta linja ovat käyttöjännitteitä varten. Signaalit menee muissa linjoissa (värejä ei ole käytetty konsistentisti). Pinnit 9 ja 10 on vaihdettavissa; ne tulee ilmoittaa RF24-kirjastolle radiomodulin olion luomisen yhteydessä ("RF24 radio(9,10)").

Kytkennoistä puuttuu virransyötöt alustoille. Ne luonnollisesti tulee myös kytkeä.

### 3. Ohjelmointi

Radiomodulilla viestinnässä RF24-kirjasto hoitaa piirille viestinnän ja radiopiiri käsittelee itsenäisesti valtaosa radioviestinnän tehtävistä. Suurin käyttäjälle jäävä työ on suunnitella miten viestinnän tulee toimia. Kuka lähettää ja milloin, mitä osoitteita käytetään.

Yksinkertaista kahden noodin paria esitellään `paja_radio_pingpair_esim.ino` ohjelmassa. Toinen noodi on lähettäjä (ping-noodi) joka sekunnin välein lähettää viestin toiselle noodille joka toimii vastaanottajan roolissa (pong-noodi). Roolivalinta tehdään pinnillä 7; jos pinni on vedetty alas, noodi toimii ping-noodina, muutoin pong-noodina. Kumpikin noodi lähettää sarjaportitse PC:lle debug-tietoa siirretyistä viesteistä. Viestien kuormana toimii 32-bittinen millisekuntilaskuri, jota käytetään arvioimaan radiolin-kin viivettä.

Mikäli useampi ryhmä testaa ping-paria yhtäaikaaisesti, noodit saattavat pingailla toisiaan melko sekavasti; käytössä on sama taajuus ja samat osoitteet. Jos näin käy, muokatkaa osoitteita ja/tai vaihtakaa taajuutta. Taajuuden vaihto tapahtuu käyttämällä `RF24::setChannel()` kutsua.

Radiopiirin radiokanavien testailuun on esimerkkiohjelma `paja_radio_scanner.ino`. Siinä käsketään radiopiiri käymään taajuuksia läpi ja poimimaan vain havaitseeko piiri kanavalta viestintää (tai häiriö-  
tehoa). Kokeilkaa ohjelmaa ja tarkastelkaa millaiselta ympäristön ilmatila näyttää. Taajuuskanavat on jaettu  $2400 + [\text{kanavanumero}] \text{ MHz}$ .

Monimutkaisempaa esimerkkinä on `paja_radio_starping.ino`. Tämä on useamman osapuolen ping-viritys; yksi (pong-)noodi toimii kohteena jolle muut (ping-noodit) lähettelevät viestejä. Pong-noodi vastaa viesteihin, kuten ping-pair esimerkissäkin. Erona on että ping-noodeja voi olla useampi. Koodi tukee viittä ping-noodia (rajoitteena on radiopiirin tukema määrä osoitteita/putkia).

Star-ping esimerkin malli vastaa tilannetta jossa on yksi tukiasema, esim kytketty verkkovirtaan kiinteässä asemassa, ja monta mittapistettä jotka voivat olla esim. akkukäyttöisiä ja liikkuvia. Mittapistet voivat lähettää tuloksensa keskusasteelle joka kerää ja raportoi ne ylöspäin. Se että sensorit ovat aktiivisia lähettäjiä mahdollistaa että sensorit voivat nukkua (=minimitehonkulutus) kunnes on niiden oman näkemyksen mukaan sopiva hetki lähettää, ja pitävät radion päällä vain sen ajan. Tällöin tukiasema joutuu pitämään radiota jatkuvasti kuuntelutilassa.

Käänteinen asetelma on luonnollisempi; ”keskusäly” ottaa yhteyttä mittapisteille. Tämä kuitenkin olisi huonompi ratkaisu mittapisteiden kannalta. Ne joutuisivat pitämään radiotaan kokoajan kuuntelutilassa.

Star-ping-ohjelmassa pong-noodin rooli valitaan kytkemällä pinni 7 maahan. Ping-noodin osoite tallennetaan EEPROM-muistiin, jolloin noodit muistavat identiteettinsä resetointien yli. Identiteetti asetetaan ja vaihdetaan kirjoittamalla ping-noodille sarjaportilla numero 1-6. Tämän jälkeen noodit tulevat resetoida jotta asetukset tulevat voimaan.

Ohjelmaa kannattaa testata useamman ryhmän kesken, jottei alustat ja radiomodulit kaikki kulu yhden tähtiverkon kokoamiseen.