



CanSat - opettajan ohjeet



Kustannusarvio ja tarvittavat linkit koottuna	3
Eri vaiheissa tarvittavat Arduino-tiedostot	4
CanSatin esittely	5
CanSat-sarja – komponenttiluettelo ja ulostulokaavio	6
Ensisijainen tehtävä	7
Ohjelmiston käynnistäminen ja Arduino Unon testaaminen	9
CanSat-laajennuksen rakentaminen	12
Ensisijaisen tehtävän anturit	18
Paineanturi (MPX4115a)	18
Lämpötila-anturi A (LM35DZ)	19
Lämpötila-anturi B (NTC)	20
CanSat-pakkauksen testaaminen	22
Anturien käyttäminen	24
Antureiden kalibrointi	25
Korkeuslaskelmat	26
Radion käyttäminen	29
Lähetin-vastaanotin	30
CanSat-laajennuspiirilevyn radio	32
Radion testaaminen	33
Terminal-ohjelma	34
Laskuvarjon suunnitteleminen	35
Laskuvarjon suunnitteluun liittyvät laskelmat	36
Laskeutumiseen liittyvä fysiikka	37
Puolipallon muotoinen laskuvarjo	38
Ristin muotoinen laskuvarjo	39
Liitovarjon muotoinen laskuvarjo	39
Litteä laskuvarjo	39
CanSatin rakenteen suunnittelu	41
Kiinnike	42
Antennin rakenne	43
Toissijainen tehtävä	44
GPS-moduuli	44

Alkuperäinen lähde: <https://www.narom.no/undervisningsressurser/the-cansat-book/v3/>

Kustannusarvio ja tarvittavat linkit koottuna

Alta löytyy taulukoituna tämän ohjeen CanSatin rakentamiseen tarvittavat kustannusarviot ja komponenttien hankkimis paikat kaikkien tavaroiden osalta. Kyseisiä komponentteja saa muualtakin, mutta Heurekalla käytössä oleviin CanSat -pakkeihin hankinnat tehtiin tämän taulukon mukaan. Lisäksi taulukon alta löytyy koottuna tarvittavat linkit eri ohjelmien ja ajureiden asentamiseen.

Komponentti	Mistä ostettu	Tilausnumero	Hinta
Arduino Uno R3	Farnell	2075382	19,02€
LM35DZ (lämpötila-anturi)	Farnell	146-9236	1,65€
Kondensaattori 1uF	Farnell	945-1455	0,0526€
Vastus 75 ohm	Farnell	934-2257	0,0356€
Vastus 10 kOhm	Farnell	934-1110	0,0356€
Piikkirima (2 kpl)	Farnell	109-7955	1,25€ / kpl
Jumpperi 2.54mm (5 kpl)	Farnell	421-8176	0,293€ / kpl
USB2.0 kaapeli A/B	Farnell	2444227	5,50€
MPX4115A (paineanturi)	Farnell	145-7150	18,42€
NTCLE100E3103JBO (NTC-anturi)	Farnell	118-7031	0,489€
Koaksiaalikaapeli RG316	Farnell	183-8802	23,42€
Paristoliitin (vain 10kpl paketeissa)	Farnell	1183124	6,15€ (sis.10kpl)
Naaras piikkirima 7 paikkaa	Farnell	2308805	0,848€
Naaras piikkirima 2 paikkaa	Farnell	3419046	0,52€
MicroSD muistikortti 8GB	Muistikauppa	TS8GUSDHC4	8,90€
Radio - APC220	DFRobot	TEL 0005	40\$
MMA7341LC 3-akselinen kiihtyvyyssanturi ±3/9g	Pololu.com	1247	<i>ei saatavilla tällä hetkellä</i>
Laskuvarjo 18 inch Nylon	Apogeerockets	29112	7,79\$
Muistikortinlukija - Logger, OpenLog	Sparkfun	DEV-13712	15,50\$

Monien tuotteiden hinnat pienenevät, kun ostettu määrä kasvaa. Taulukon hinnat ovatkin korkeimmat mahdolliset kustannukset. Muistikortiksi kelpaa mikä tahansa: tallennettavan datan määrä ei vie kovin paljon tilaa. Alennuksista löytyykin 2GB, 4GB ja 8GB muistikortteja melko usein. Laskuvarjon voi puolestaan valmistaa itse ja Arduino Unoja saattaa löytyä edullisemmin muualtakin. Osa tavaroista puolestaan jää uudelleen käyttöön seuraavia ryhmiä varten.

Näiden lisäksi tarvitaan

- CanSat Shield -lisäosa versio 3 (Naromilta saa pyydettäessä)
- Kolvit
- Juotostinaa
- Tietokoneet
- Tölkkejä tai muita tölkin kokoisia ratkaisuja
- Perustyökaluja

Tietokoneilla puolestaan tulee olla ladattuina seuraavat ohjelmat ja laajennukset:

- Arduino Ide (<http://arduino.cc/>)
- USB-TTL -moduulin VCP-ohjaimet osoitteesta Silabs.com (CP210x USB-UART-sillan VCP-ohjaimet).
 - Napsauta linkkiä ja lataa käyttöympäristösi sopiva tiedosto. Kun olet ladannut tiedoston, pura se ja aja CP210xVCPInstaller_x64.
 - On suositeltavaa asentaa ensin USB-TTL-muuntimen ohjaimet ja kytkeä sitten USB-muunnin ja siihen liitetty APC-moduuli. Kun moduuli on kytketty, Windows tunnistaa sen ja määrittää tarvittavat ohjaimet. Jos Windows ei tunnista laitetta tai ilmoittaa, ettei laite toimi kunnolla, irrota laite USB-portista ja kytke se uudelleen.
- Radion ohjelmointiin tarvittava ohjelma apc220cfg.ino. Voit ladata sen [tästä](#).
- Terminal -ohjelma radioyhteyden tarkasteluun: <https://sites.google.com/site/terminalbpp/>

Tämän lisäksi alla olevista ohjeista löytyy valmiita mallitiedostoja Arduino-ohjelmointia varten, näitä ei kuitenkaan tarvitse erikseen asentaa koneelle.

Eri vaiheissa tarvittavat Arduino-tiedostot

- Arduino Unon testaamiseen tarvittava tiedosto: [Arduinotest textfile](#)
- CanSat pakin testaus tiedosto (voi käyttää myös varsinaisia testejä tehdessä): [Laajennuksen testaus](#).
- Radion ohjelmointiin tarvittava tiedosto ladataan [tästä](#).

CanSatin esittely

Mikä CanSat on?

CanSat on oikean satelliitin mallinnettu versio, joka on sovitettu limsatölkin (330 ml) kokoon ja muotoon. Oppilaiden haasteena on saada kaikki satelliitissa olevat alijärjestelmät, kuten virta, anturit ja viestintäjärjestelmä, mahtumaan näin pieneen kokoon. Sen jälkeen CanSat laukaistaan muutaman sadan metrin korkeuteen raketilla tai pudotetaan korokkeelta, lennokista tai ilmapallosta. Näin alkaa sen tehtävä eli tieteellisen kokeen suorittaminen ja turvallinen laskeutuminen.

CanSatien avulla oppilaat saavat ainutlaatuisen mahdollisuuden päästä käytännössä kokemaan oikea avaruusprojekti. Oppilaat ovat vastuussa kaikesta projektiin liittyvästä: sen tehtävän valitsemisesta, CanSatin suunnittelusta, komponenttien integroimisesta, satelliitin tietokoneen ohjelmoimisesta, testauksesta, laukaisuun valmistautumisesta ja tietojen analysoinnista.

CanSat-pakkaus

CanSat-pakkaus perustuu Arduino Uno -piirilevyyn ja anturilaajennukseen. Arduino on avoimeen lähdekoodiin perustuva sähköinen testausympäristö, joka perustuu joustavaan ja helppokäyttöiseen laitteistoon ja ohjelmistoon. Se on tarkoitettu taiteilijoille, suunnittelijoille, harrastelijoille ja kaikille vuorovaikutteisten laitteiden tai ympäristöjen tekemisestä kiinnostuneille. Arduino voi seurata ympäristöä erilaisista antureista saatujen tietojen avulla ja vaikuttaa ympäristöönsä ohjaamalla valoja, moottoreita ja muita toimilaitteita. Esimerkkejä Arduino projekteista:

- automaattinen yövalo, joka syttyy hämärässä
- murtohälytin
- termostaatti
- viivaa seuraava robotti
- amatööriradio ja morsetin / radiomajakka
- graafinen laskin, joka tekee syötetyistä sarjoista kaavion LCD-näyttöön
- Wi-Fi-yhteydellä ohjattava auto.

Lisää ideoita löydät osoitteesta <http://arduino.cc/playground/Projects/Ideas>

Anturikortin ovat kehittäneet Aalborgin yliopistossa professori Jens Dalsgaard Nielsen ja Simon Jensen. Laajennuksessa on seuraavat ominaisuudet:

- radio (APC220) ja antenni
- paineanturi (MPX4115A)
- lämpötila-anturi (LM35DZ)
- lämpötila-anturi (NTC 10k)
- kolmen akselin kiihtyvyysanturi $\pm 3/9$ g (MMA7341LC) *(ei saatavilla tällä hetkellä)*
- SD-muistikortti (OpenLog)

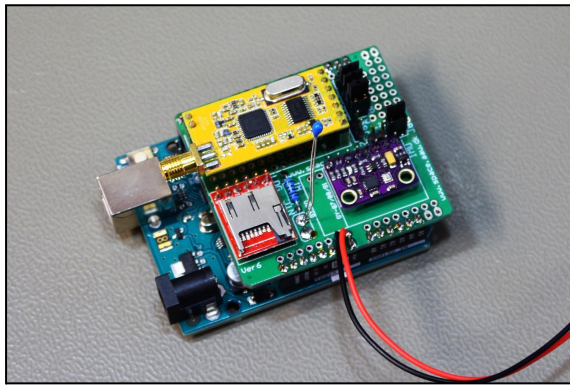
CanSat-sarja – komponenttiluettelo ja ulostulokaavio

CanSat-sarjan tiedot

Anturikortin ovat kehittäneet Aalborgin yliopistossa professori Jens Dalsgaard Nielsen ja Simon Jensen. Laajennuksessa on seuraavat ominaisuudet:

- radio (APC220) ja antenni
- paineanturi (MPX4115A)
- lämpötila-anturi (LM35DZ)
- lämpötila-anturi (NTC 10k)
- kolmen akselin kiihtyvyysanturi $\pm 3/9$ g (MMA7341LC) *(ei saatavilla tällä hetkellä)*
- SD-muistikortti (OpenLog)

Oikealla olevassa kuvassa on osaluettelo. Voit avata kuvasta suuremman version [täältä](#).

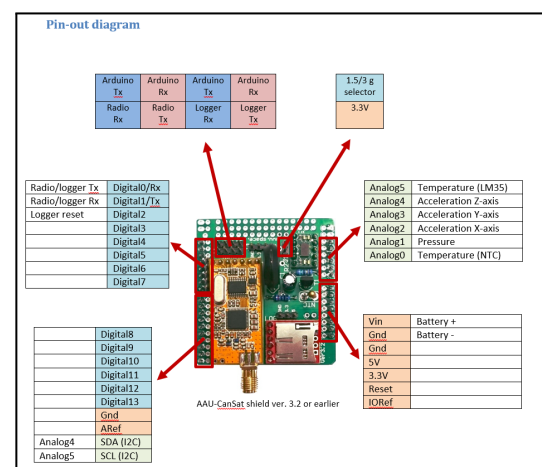


	Component	Where to buy (a suggestion)	Ordering number
1	Arduino Uno R3	Farnell	2075382
1	Resistor 10 kOhm		934-1110
2	36 Breakaway headers		109-7955
5 (7)	Jumper 2.54mm 2p		421-8176
1	USB2.0 Cable A/B		8706794
1	Vishay, NTCLE300E31035B		1888081
1	(1/2) Coax Cable RG316		183-8802
1	Battery Connector (10 stk)		1183124
1	Female Headers 7p		2308805
1	Female Headers 2p		3419046
1	MicroSD Card 8GB		
1	Radio - APC220	DFRobot	TEL0005
1	Digital 10- DOF Sensor GY-91	Banggood.com (China)	
1	Parachute 18 Inch Nylon	Eller QXL.no	
1	Logger - OpenLog	Apogeerockets	29112
1	Logger - OpenLog	Sparkfun	DEV-13712
1	Aalborg CanSat shield vs. (4 -)7	Aalborg Uni./NAROM	

Anturilaajennus on suunniteltu sopimaan Arduino Uno R3 -piirikortin päälle. Ulostulokaavio näkyy alla olevassa kuvassa. Voit avata kuvasta suuremman version [täältä](#).

Kuten kuvassa näkyy, seuraavat anturit käyttävät Arduino Unon analogisia ulostuloja:

- A0: lämpötila (NTC 10k)
- A1: paine
- A2: kiihtyvyys, x-suunta *(ei käytössä)*
- A3: kiihtyvyys, y-suunta *(ei käytössä)*
- A4: kiihtyvyys, z-suunta *(ei käytössä)*
- A5: lämpötila (LM35DZ)



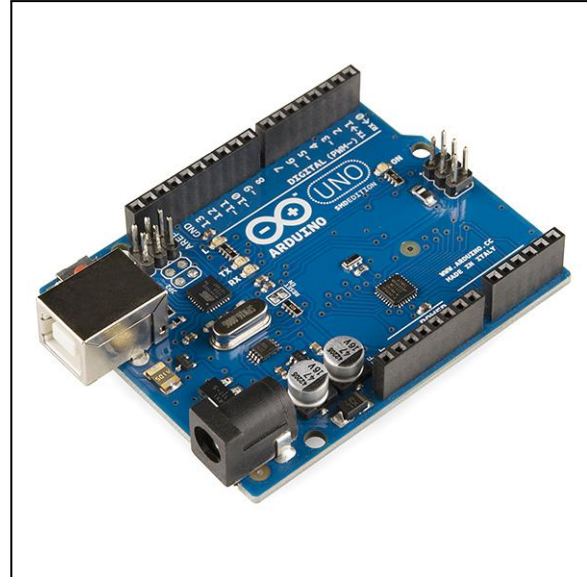
Ulostulokaavio ([kuva isompana](#))

Ensisijainen tehtävä

Aloittaminen

Arduino Uno ATmega328:aan perustuva mikro-ohjain. Siinä on 14 digitaalista tulo-/lähtönastaa (joista kuutta voidaan käyttää pulssinleveysmodulaatiolähtönä (PWM)), 6 analogista tuloa, 16 MHz:n kide, USB-liitäntä, virtaliitäntä, ICSP-liittimet ja palautuspainike. Se sisältää kaiken mikro-ohjaimen tueksi tarvittavan. Voit liittää sen tietokoneeseen USB-kaapelilla ja käyttää virtalähteenä joko verkkovirtasovittinta tai paristoa.

Yhdysvaltojen ulkopuolella Arduino-piirilevyt myydään nyt nimellä Genuino, joten Genuino Uno vastaa tekstissä mainittua Arduino Unoa.



Arduino Uno R3 (Wikipedia, [kuva isompana](#))

Aloita varaamalla valmiiksi ARDUINO UNO, tietokone ja normaali USB-kaapeli (A–B) eli sellainen, jota käytetään USB-tulostimen kytkemiseen.

Avoimen lähdekoodin Arduino-ympäristössä koodin kirjoittaminen ja lataaminen I/O-piirilevyyn on helppoa. Se toimii Windows-, Mac OS X- ja Linux-järjestelmissä. Ympäristö on kirjoitettu Javalla ja se perustuu Processing- ja AVR-gcc-ohjelmistoihin ja muihin avoimen lähdekoodin ohjelmistoihin.

Ohjelmiston asentaminen

Ennen kuin kytket Arduino-piirilevyn tietokoneeseen, varmista, että piirilevyn käyttämiseen tarvittavat ohjelmistot ja ohjaimet on asennettu.

Voit ladata Arduinon viimeisimmän integroidun kehitysympäristöohjelmiston (IDE) osoitteesta <http://arduino.cc> napsauttamalla ylävalikon Downloads (Lataukset) -linkkiä. Voit valita käyttöjärjestelmäsi sopivan tiedoston.

- Lataa ja aja Arduino IDE:n viimeisin versio (versio 1.6.12).
- Jos käytät Windowsia ja sinulla on tietokoneen pääkäyttäjän oikeudet, lataa Windows-asennusohjelma. Jos sinulla ei ole tarvittavia oikeuksia, lataa zip-tiedosto ja pura paketti tietokoneen paikalliseen hakemistoon.

On suositeltavaa käyttää asennusohjelmaa, jos sinulla on pääkäyttäjän oikeudet. Näin voit asentaa Arduino-piirilevyn ohjaimet automaattisesti.

Piirilevyn kytkeminen

Arduino Uno käyttää virtalähteenä automaattisesti joko tietokoneen USB-liitäntää tai ulkoista virtalähdettä. Kytke Arduino-piirilevy tietokoneeseen USB-kaapelin avulla. Virran vihreän LED-merkkivalon (PWR) pitäisi syttyä.

Ohjainten asentaminen

Jos käytät zip-tiedostoa tai poistit Install driver (Asenna ohjain) -kohdan valinnan asennuksen aikana, sinun on ehkä asennettava ohjaimet manuaalisesti, jotta voit kytkeä Arduino-piirilevyn.

Arduino Uno -ohjainten asentaminen Windows-käyttöjärjestelmään:

- Kytke piirilevy tietokoneeseen ja odota, kunnes Windows aloittaa ohjainten asennusprosessin. Joissakin tietokoneissa ohjaimet asentuvat automaattisesti. Jos niin ei käy, noudata seuraavia ohjeita.
- Napsauta Käynnistä-valikkoa ja avaa Ohjauspaneeli/Järjestelmä ja Tietoturva/Laitehallinta. Jos käytössä on Windows 10, valitse Setup/Units/other (Asennus/Yksiköt/muut).
- Etsi portit (COM ja LPT). Luettelossa pitäisi näkyä avoin portti Arduino UNO (COMxx). Joissakin tapauksissa Arduino Unon tilalla näkyy Unknown device (Tuntematon laite).
- Napsauta hiiren kakkospainikkeella Arduino UNO (COMxx) -porttia ja valitse Päivitä ohjainohjelmisto.
- Valitse sitten Etsi ohjainohjelmistoa tietokoneesta.
- Etsi ja valitse Arduino Unon ohjaintiedosto ArduinoUNO.inf. Se on Arduino-ohjelmiston latauskansion Drivers-hakemistossa (ei FTDI USB Drivers -alihakemistossa).
- Tämän jälkeen Windows asentaa ohjaimen.

Arduino-sovelluksen käynnistäminen

Kaksoisnapsauta Arduino-sovellusta kansiossa, johon purit sovelluksen paketin.

Ohjelmiston käynnistäminen ja Arduino Unon testaaminen

Arduino-sovelluksen käynnistäminen

Kaksoisnapsauta Arduino-sovellusta kansiossa, johon purit sovelluksen paketin.

Mallikoodin avaaminen

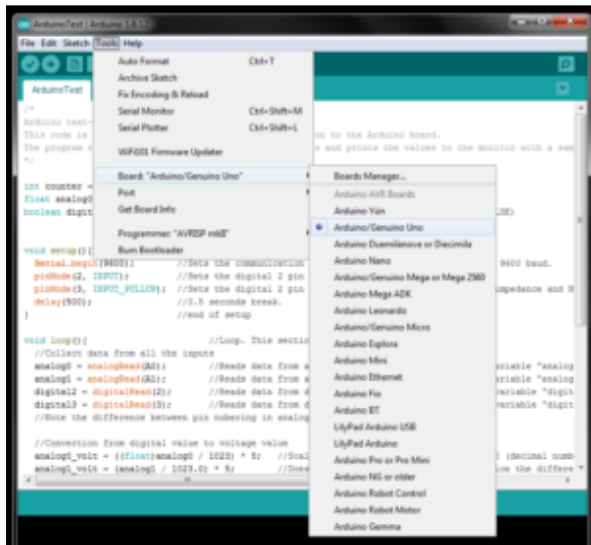
NAROM on luonut koosteen ohjelmistokoodista, joita kurssin aikana käytetään.

Voit käyttää koodia Arduino Unon testaamiseen kopioimalla tekstitiedoston [Arduinotest textfile](#) (avaa linkki) tekstin ja liittämällä sen juuri avaamaasi Arduino-luonnosikkunaan. (Katso oikealla oleva [kuva isompana](#).)



Arduino IDE ja testikoodi

Piirilevyn valitseminen



Valitse Arduinoasi vastaava piirilevy valikon kohdasta Tools (Työkalut) → Board (Kortti). (Katso vasemmalla oleva [kuva isompana](#).) Muuten tiedonsiirto Arduino Uno -piirilevyyn ei onnistu.

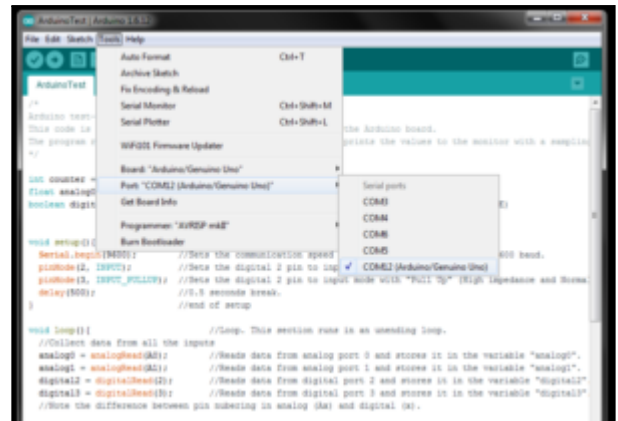
Sarjaportin valitseminen

Valitse Arduino-piirilevyn sarjalaite valikon kohdasta Tools (Työkalut) → Serial Port (Portti).

Piirilevyn valitseminen

Käytettävissä on luultavasti useita COM-sarjaportteja. Arduinon COM-sarjaportin numero on luultavasti suurin. Voit varmistaa asian irrottamalla Arduino-piirilevyn ja avaamalla valikon uudelleen. Arduino-piirilevyn portin numero on se, joka katoaa. Kytke piirilevy uudelleen ja valitse oikea sarjaportti.

Arduino IDE:n versiossa 1.6.12 oikea sarjaportti on nimetty (Arduino/Genuino Uno) porttiluettelossa.



COM-sarjaportin valitseminen ([kuva isompana](#))

Ohjelman lataaminen



Koodin lataaminen ([kuva isompana](#))

Napsauta kääntäjässä Upload (Lähetä) -painiketta (vasemmalla oleva oikealle osoittava nuoli, katso vasemmalla oleva kuva).

Kun käytät uutta luonnosta ensimmäisen kerran, järjestelmä pyytää sinua joskus tallentamaan luonnoksen kansioon, joka on nimetty luonnoksen mukaan. Muussa tapauksessa luonnoksen ajo ei onnistu odotetusti.

Odota muutama sekunti – Arduinon RX- ja TX-LED-valojen pitäisi vilkkua. Jos lataus onnistuu, tilapalkkiin tulee teksti Done uploading (Lähetetty).

Tietojen lukeminen

Avaa Serial Monitor (sarjamonitori) -ikkuna, jossa voit tarkastella Arduino Uno-piirilevyn lähettämiä tietoja.



Serial Monitorin avaaminen ([kuva isompana](#))

Voit avata Serial Monitorin (sarjamonitorin) napsauttamalla työkalupalkin oikealla puolella olevaa kuvaketta. (Katso oikealla oleva kuva.) Serial Monitor (sarjamonitori) aukeaa uuteen ikkunaan, jossa näet Arduino-piirilevystä tulostetun sarjatekstin virran.

Arduino Uno -piirilevyn testaaminen

Voit varmistaa testaamalla, että vastaanottamasi tiedot ovat oikeat. Aseta jokainen kanava maahan ja virtaan ja katso Serial Monitorin (sarjamonitorin) tulostetta. Tätä varten Arduino Uno -piirilevyn sisääntuloliitännät ja virtaliitännät on kytkettävä yhteen johdolla.

Voit käyttää koodia Arduino Unon testaamiseen kopioimalla tekstitiedoston [Arduinotest textfile](#) (avaa linkki) tekstin ja liittämällä sen juuri avaamaasi Arduino-luonnosikkunaan. (Katso oikealla oleva [kuva isompana](#).)

- Kytke johto A0-porttiin.
- Kytke johdon toinen pää GND-porttiin.
- Serial Monitorin Analog0-arvon pitäisi olla 0,0 voltia.
- Irrota johto GND-liitännästä ja kytke se 5V-liitäntään.
- Nyt Analog0-arvon pitäisi olla noin 5,0 voltia.
- Irrota johto 5V-liitännästä ja kytke se 3.3V-liitäntään.
- Nyt Analog0-arvon pitäisi olla noin 3,3 voltia.
- Toista sama A1-, D2- ja D3-liitännöille.

Saatko saman arvon digitaalisesta portista sekä 3,3 voltilla että 5 voltilla?

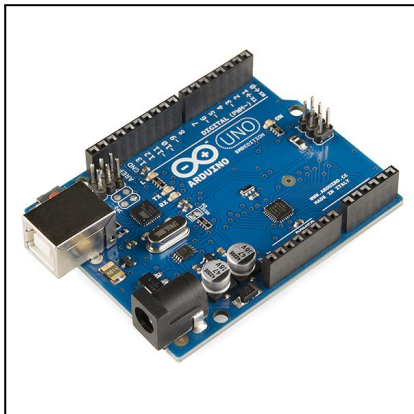
CanSat-laajennuksen rakentaminen

Komponenttien kokoaminen ja CanSat-laajennuksen rakentaminen

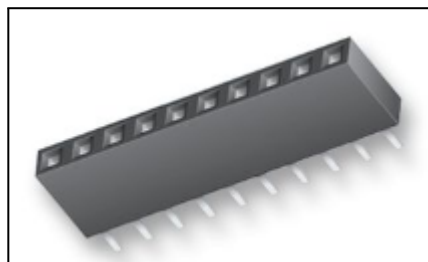
Tässä osiossa näytetään vaiheittain, kuinka CanSat-laajennuksen komponentit kootaan.

Osaluettelo

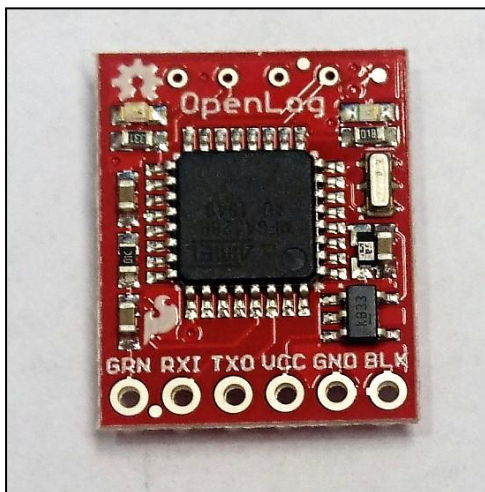
Arduino Uno R3



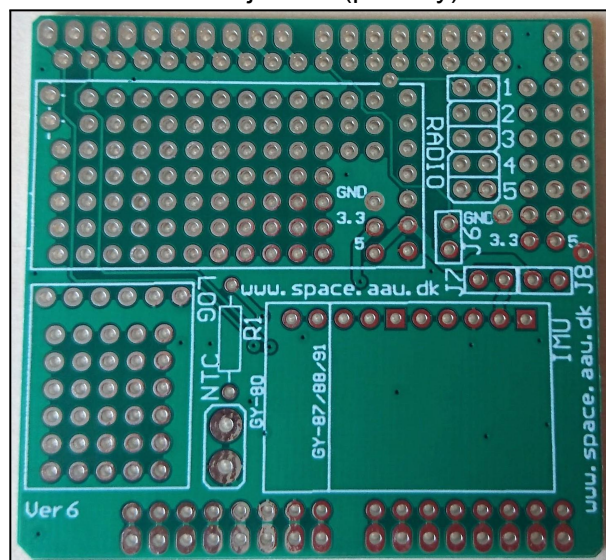
Piikkirima (naarasliitäntä) 7-nastainen ja 2-nastainen



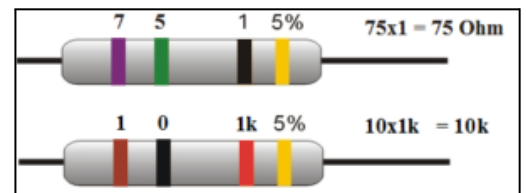
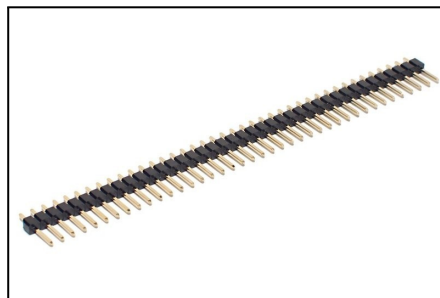
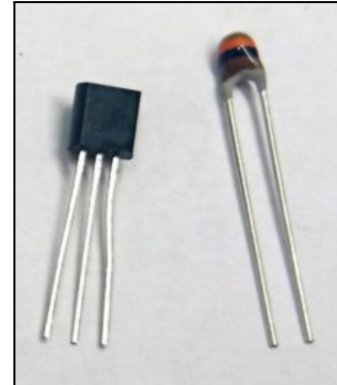
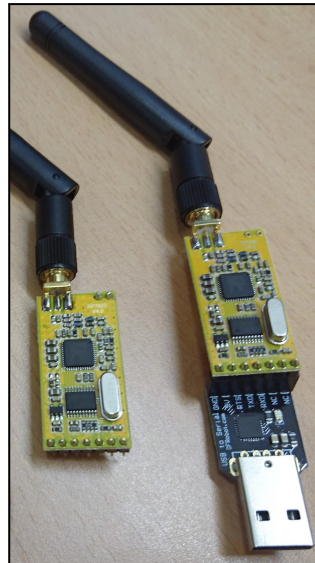
OpenLog-dataloggeri (ja SD-kortti)



CanSat-laajennus (piirilevy)



Lämpötila-anturit LM35DZ ja
NTC (NTCLE100E3103JB0)



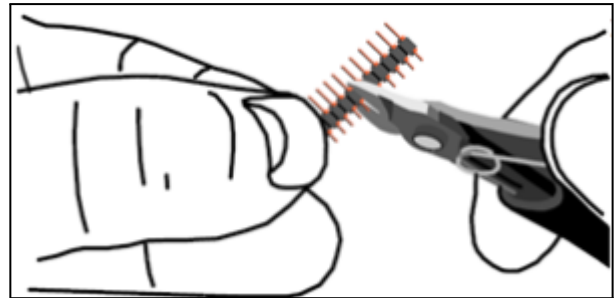
MPX4115A-paineanturi | Piikkirima (uroslitittä) 2 x 32-nastainen | Vastukset 1x 75 Ohm 1x 10 kOhm

MMA7361L-kiihtyvyyssanturi (3-akselinen) (ei saatavilla tällä hetkellä)

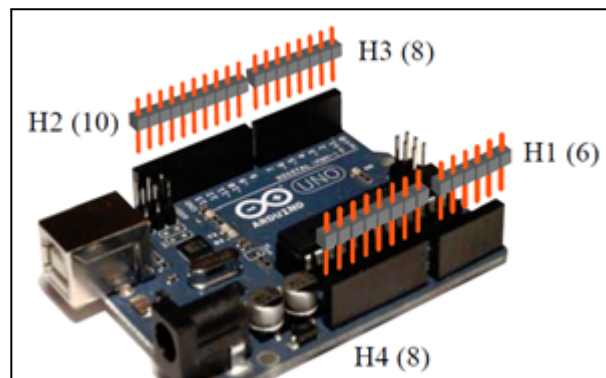
Kokoamisohje

1. Leikkaa piikkirima seuraavan pituisiin kappaleisiin:

- 6 nastaa (H1)
- 10 nastaa (H2)
- 8 nastaa (H3)
- 8 nastaa (H4)

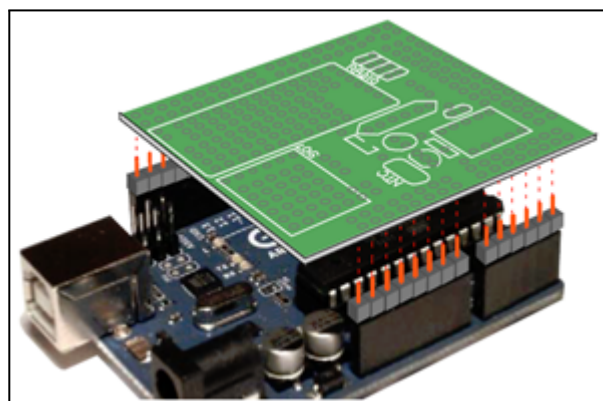


2. Aseta piikkirimat Arduino-piirilevyyn lyhyet nastat ylöspäin.

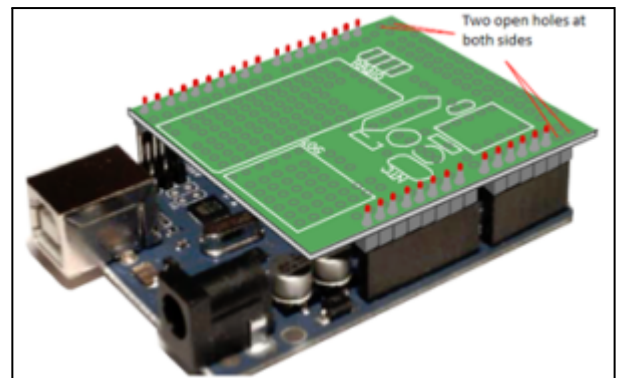


3. Kiinnitä laajennuspiirilevy Arduino Unon päälle.

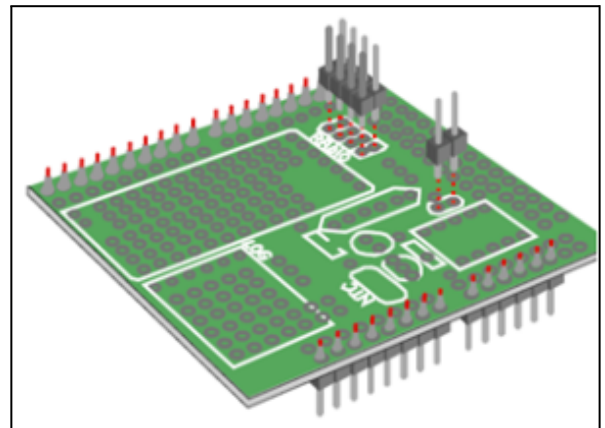
Huom. Piirilevy menee paikoilleen vain yhdessä asennossa.



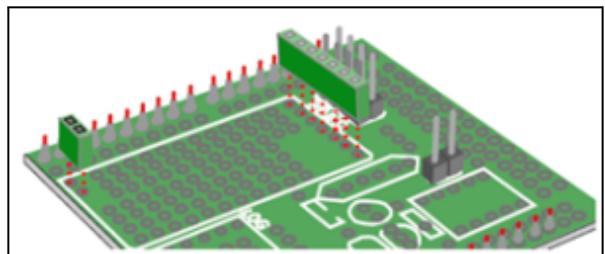
4. Juota kaikki nastat piirilevyn päällyspuolelle ja irrota piirilevy sitten Arduino Unosta. Älä kuumenna nastoja liian kauaa juottaessasi niitä. Arduino Uno -piirilevy voi vahingoittua, jos se altistuu liialle kuumuudelle.



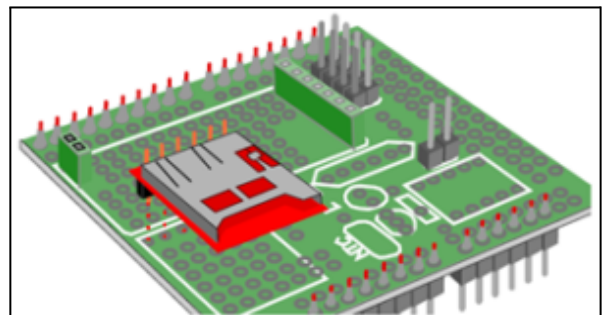
5. Leikkaa kaksi 4-nastaista piikkirimaa ja yksi 2-nastainen piikkirima. Liitä ne laajennuspiirilevyn päälle ja juota ne pohjapuolelta (juotospuolelta) kuvan mukaisesti. (Piikkirimat asetetaan kohtiin J1, J2 ja J3.)



6. Naaraspuoleisilla piikkirimoilla radio on helppo kiinnittää ja irrottaa ohjelmointia varten. Jos et ole käyttänyt tätä sarjaa aiemmin, on suositeltavaa käyttää naaraspuoleisia piikkirimoja. Ne voidaan poistaa tarvittaessa, jotta kokoonpanosta tulee kestävämpi. Aseta piikkirimat piirilevyn päälle ja juota ne juotospuolelta.

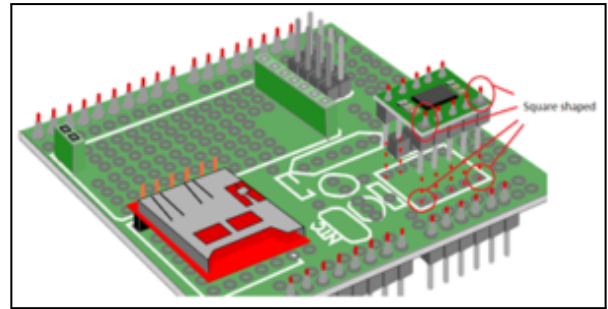


7. Juota dataloggeri laajennuslevyyn 6-nastaisen piikkiriman avulla. Aseta piikkirima lyhyet nastat alaspäin niin, että nastat menevät laajennuslevyn läpi. Varmista, että dataloggeri on asennettu oikein päin (katso kuva). On suositeltavaa käyttää piirilevyn ja dataloggerin välissä kuumaliimaa tukemassa loggeria. Jos dataloggeria ei tueta, se saattaa helposti irrota ja vahingoittaa juotospisteitä.

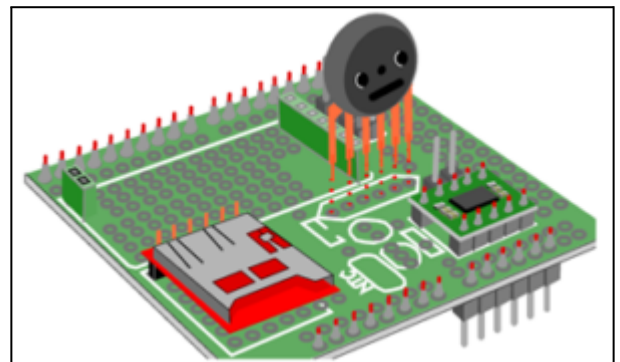


8. TÄTÄ OSIOTA EI VALITETTAVASTI VOIDA TOTEUTTAA, KOSKA SOPIVAN KIIHTYVYYSANTURIN VALMISTUS ON LOPETETTU.

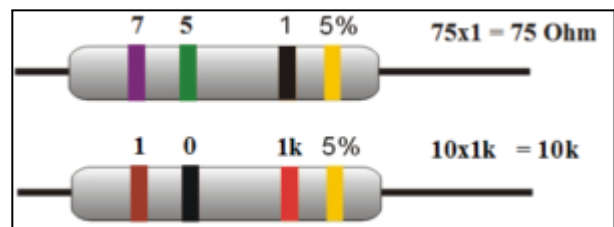
Juota kiihtyvyyssanturi laajennuspiirilevyn kohtaan U1 käyttämällä kahta 5-nastaista piikkirimaa. Aseta anturi huolellisesti oikein päin. Kiihtyvyyssanturin mustan IC-sirun on oltava päällä (osoitettava ylöspäin). Huomaa myös, että kaksi laajennuspiirilevyn ja anturin juotospisteistä on neliön muotoisia ympyrän sijaan. **Nämä pisteet on asetettava kohdakkain.**



9. Aseta paineanturi laajennuspiirilevylle ja juota se pohjapuolelta. Aseta anturi huolellisesti oikein päin (kuvan mukaisesti).

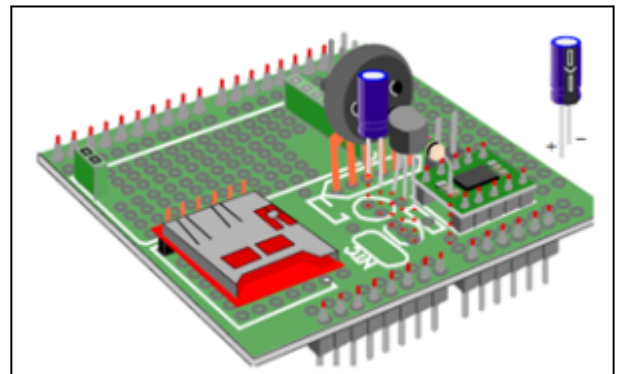


10. Juota 75 ohmin vastus kohtaan R2. Huom. Sarjassa on kaksi erilaista vastusta. Varmista, että käytössä on 75 ohmin vastus eikä 10 kilo-ohmin vastus.

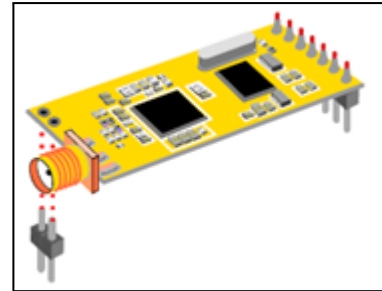


11. Aseta lämpötila-anturi (LM35DZ) kohtaan T1. Anturin oikea asento on merkitty piirilevyyn. Jos asennat anturin väärin päin, se voi palaa.

12. Jatka asentamalla 1 μF :n kondensaattori kohtaan C1. Kondensaattorin asento on ilmoitettu piirilevyssä plusmerkillä (+). Kondensaattorin pisin nasta on positiivinen. Negatiivinen nasta on myös merkitty (valkoisella) kondensaattorin kylkeen.

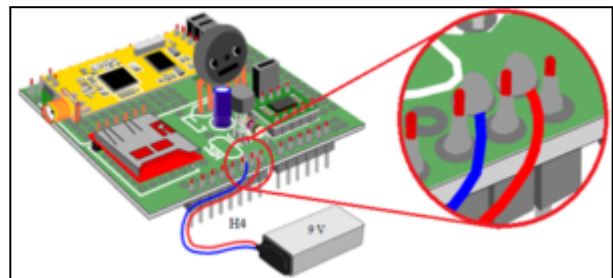


13. Juota 2-nastainen piikkirima yhteen radioista. Nastojen tarkoitus on ainoastaan tukea radion liitää ja vähentää siihen kohdistuvaa räsitusta. Kiinnitä radio laajennuspiirilevyyn.

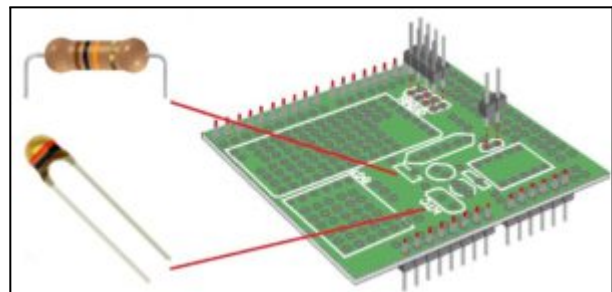


14. Juota pariston liitin laajennuspiirilevyyn. **Kytke punainen johto Vin-porttiin ja musta johto GND-porttiin.**

Johdon kiinnityskohdat irtoavat helposti. Voit välttää tämän kiinnittämällä johdot piirilevyyn kuumaliimalla, mikä vähentää kiinnityskohtaan kohdistuvaa räsitusta.



NTC-lämpötila-anturi ja 10 kilo-ohmin vastus
Kaksi komponenttia on vielä asentamatta:
NTC-lämpötila-anturi ja 10 kilo-ohmin vastus. NTC-lämpötila-anturi on valinnainen, mutta sen asentaminen on suositeltavaa. Sen vasteaika on lyhyempi kuin LM35DZ-anturin.



Huom. Jos käytät NTC-anturia, sinun on asennettava myös 10 kilo-ohmin vastus (R1), jotta anturi toimii.

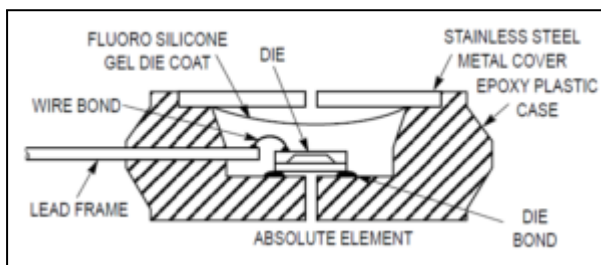
CanSat-laajennuspiirilevy on nyt valmis testattavaksi.

Ensisijaisen tehtävän anturit

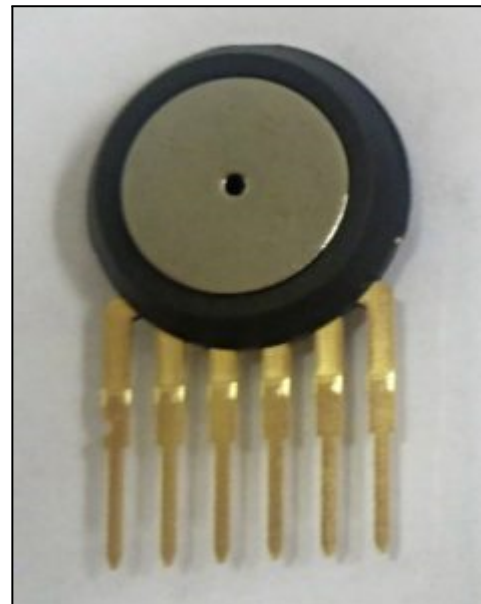
Tämän oppaan osiossa Ensisijainen tehtävä käsitellään paineanturi ([MPX4115A](#)) sekä lämpötila-anturit ([LM35DZ](#) ja [NTC](#)).

Kiihtyvyyssanturi ei kuulu ensisijaiseen tehtävään, joten se kuvataan oppaan Toissijainen tehtävä -osiossa.

Paineanturi (MPX4115a)



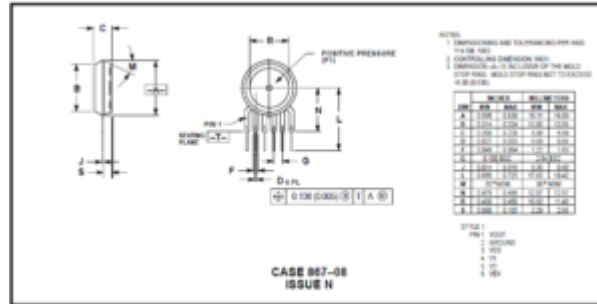
Paineanturin poikkileikkaus



Paineanturi

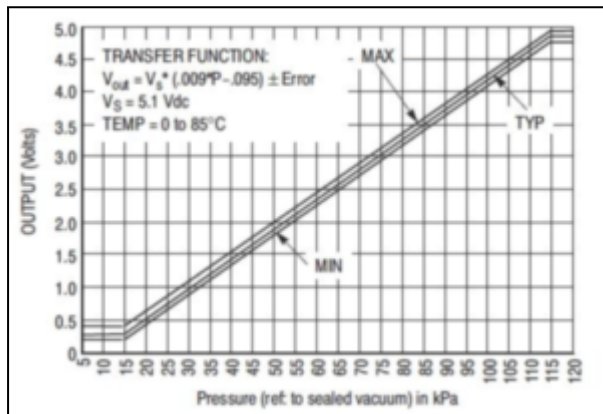
Käytettävä paineanturi on Motorolan valmistama MPX4115A. Se käyttää silikonista pietsosähköistä anturielementtiä. Oikeanpuoleinen kuva on anturin poikkileikkaus.

Kun materiaali on pietsosähköinen, sen vastus muuttuu, kun siihen kohdistetaan mekaanista räsitusta. Tässä tapauksessa pietsosähköinen materiaali on silikoni. Silikonin vastuksen muutokset ovat suurempia kuin esimerkiksi teräksen, joten se soveltuu hyvin esimerkiksi paineanturin anturielementiksi.



Paineanturin kotelotyyppi. Voit avata kuvasta suuremman version [täältä](#).

Yläpuolella olevassa kuvassa on anturin yksityiskohtainen kuvaus. Siinä näkyvät anturin mitat ja kaavio liitännöistä. Jotta mitattuja jännitteitä voidaan verrata ympäristön painearvoihin, tarvitaan anturin siirtofunktio. Siirtofunktio kuvaa anturin lähtöjännitteen ja vastaavan paineen matemaattista suhdetta. Funktio mainitaan anturin teknisissä tiedoissa. Alavasemmalla olevassa kuvassa näkyy anturin [teknisten tietojen](#) mukainen lähtöjännitteen ja ympäristön ilmanpaineen välinen suhde.



Lähtöjännite V_{out} on:

$$V_{out} = V_s(0.009P - 0.095)$$

Kun P on ilmanpaine (kPa) ja V_s on syöttöjännite.

Paineanturin siirtofunktio. Voit avata kuvasta suuremman version [täältä](#).

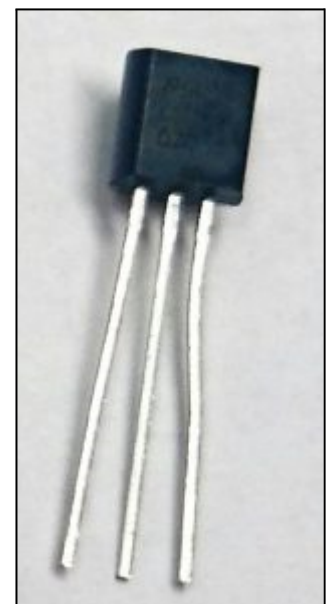
Lämpötila-anturi A (LM35DZ)

LM35DZ-lämpötila-anturi on tehty celsiusasteiden (tai kelvinien) lineaarista mittaamista varten.

Anturi on suunniteltu lämpötiloille 0–100 °C.

Lähtöjännite on 250 mV 25 °C:n lämpötilassa ja anturin herkkyys on 10 mV/°C.

Näillä tiedoilla ja [LM35:n teknisillä tiedoilla](#) voit määrittää anturin siirtofunktion. Voit tehdä sen tietokoneella tai laskimella.



Yleinen yhtälö anturin lähtöjännitteelle lämpötilan funktiona on

Anturin lähtö [V] = herkkyys [V/°C] * lämpötila [°C] + lähtö 0 °C:ssa (poikkeama)

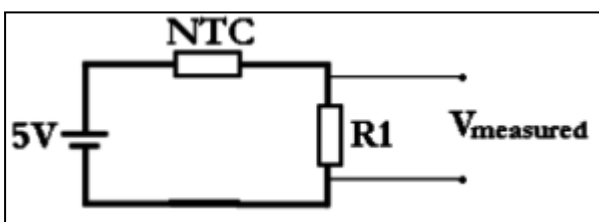
Tämä on edellä mainitulla herkkyydellä ja lämpötilassa $t_{\text{celsiusasteina}}$ on:

$$\text{Sensoroutput}[V] = 0.01 \frac{V}{^{\circ}C} * (t - 25^{\circ}C) + 0.250V = 0.010 \frac{V}{^{\circ}C} (t - 25^{\circ}C) + 0.250V$$

Lämpötila-anturi B (NTC)

Toinen CanSatin lämpötila-antureista on Vishay/BC componentsin valmistama NTCLE203E3103JBO. Se on Negative Temperature Coefficient- eli NTC-termistori. Sen lämmönjohtavuus paranee lämpötilan noustessa. Useimmat keraamiset materiaalit käyttäytyvät samalla tavalla. Muut materiaalit taas toimivat päinvastoin, ja niiden lämmönjohtavuus heikkenee lämpötilan noustessa. Sen vuoksi useimmat NTC-termistorit tehdään puolijohtavista materiaaleista, joiden sähkönjohtavuus sijoittuu eristeiden ja johtimien väliin ja joilla on tiettyjä erityisominaisuuksia.

Kun näitä materiaaleja lämmitetään, materiaalin elektronit saavat energiaa, ja entistä useammat elektronit saavat tilaa liikkua. Näin materiaalin sähkönjohtavuus paranee. Kun materiaalin sähkönjohtavuus paranee, sen resistanssi pienenee. Näin ollen lämpötilan nostaminen vähentää resistanssia. Tämän käänteisen suhteen vuoksi anturia kutsutaan negatiivisen lämpötilakertoimen (NTC) vastukseksi.



NTC-mittauspiiri

Lämpötila-anturi on kytketty anturipiirikortissa sarjaan vastuksen (R_1)

kanssa. Vastuksen jatkuva resistanssi on 10 kΩ, kuten yläpuolella olevan kuvan yksinkertaistetusta kaaviosta näkyy. Kun vastukset kytketään sarjaan, piirin virta on sama kaikkialla. Virtapiirin kokonaisresistanssi R_T voidaan laskea seuraavasti:

$$R_T = R_1 + R_{NTC}$$

$$I = \frac{U}{R} = \frac{5V}{(R_1 + R_{NTC})}$$

Sama virta I kulkee kiinteän vastuksen R_1 läpi ja tuottaa jännitteen $V_{measured}$ koko vastukseen. Tämä voidaan ilmaista seuraavalla yhtälöllä:

$$I_{R1} = \frac{U_{R1}}{R_1} = \frac{V_{measured}}{R_1}$$

Koska virta on sama kaikkialla piirissä, voidaan määrittää seuraava yhtälö:

$$I_{R1} = I_{NTC}$$

$$\frac{5V}{(R_1 + R_{NTC})} = \frac{V_{measured}}{R_1}$$

Tästä saadaan mitatun jännitteen $V_{measured}$ ja NTC-lämpötila-anturin resistanssin välinen suhde R_{NTC} .

$$V_{measured} = \frac{5V \cdot R_1}{(R_1 + R_{NTC})}$$

$$R_{NTC} = \frac{5V \cdot R_1}{V_{measured}} - R_1$$

Koko siirtofunktion määrittämiseksi tarvitsen myös lämpötilan ja anturin resistanssin R_{NTC} välisen suhteen. Tämä on ilmoitettu anturin [teknisissä tiedoissa](#).

CanSat-pakkauksen testaaminen

Kun CanSat-laajennuspiirilevy on valmis, varmista testaamalla, että kaikki komponentit toimivat oikein. Seuraavaksi käydään testaustoimet läpi vaiheittain:

1. Asenna laajennuspiirilevy Arduino Unon päälle. Varmista ennen laajennuspiirilevyn asentamista, ettei virtaa ole kytketty.
2. Käynnistä CanSat kytkemällä siihen paristo

Arduino Unossa on vihreä ON-valo. Valon pitäisi aina palaa, kun Arduino Uno on kytketty virta.

Jos virtavallo ei syty, testaa paristo ja varmista, ettei laajennuspiirilevyssä ole oikosulkuja.

3. Irrota paristo, kytke USB-kaapeli ja tarkista ON-valo uudelleen.
4. Avaa uusi Arduino-luonnos ja syötä seuraavan linkin kooditeksti: [Laajennuksen testaus](#). Anna uuden luonnoksen nimeksi ShieldTest.ino. Jos olet tehnyt niin jo aiemmin, avaa luonnostiedosto ShieldTest.ino Arduino IDE:ssä (versio 1.0 tai uudempi) ja lataa luonnos Arduino Uno -piirilevyyn. Tämä on testiohjelma, jolla näet kaikkien CanSat-sarjan anturien mittaustulokset.
5. Käynnistä Arduino IDE:ssä Serial Monitor (sarjamonitorin).

Näyttöön pitäisi tulla oikealla olevaa kuvaa vastaavia tietoja.

```
Counter: 0 / Time(s): 0.00 / Temp: 2.20 V / NTC: 2.06 V / Pressure: 2.03 V / Acceleration (x,y,z): 2.96 V, 1.91 V, 2.07 V,  
Counter: 1 / Time(s): 0.50 / Temp: 1.58 V / NTC: 2.02 V / Pressure: 1.99 V / Acceleration (x,y,z): 1.55 V, 1.89 V, 2.04 V,  
Counter: 2 / Time(s): 0.99 / Temp: 1.54 V / NTC: 1.98 V / Pressure: 1.96 V / Acceleration (x,y,z): 1.53 V, 1.87 V, 2.02 V,  
Counter: 3 / Time(s): 1.50 / Temp: 1.51 V / NTC: 1.96 V / Pressure: 1.94 V / Acceleration (x,y,z): 1.51 V, 1.86 V, 2.00 V,  
Counter: 4 / Time(s): 2.00 / Temp: 1.48 V / NTC: 1.93 V / Pressure: 1.92 V / Acceleration (x,y,z): 1.50 V, 1.84 V, 2.03 V,  
Counter: 5 / Time(s): 2.50 / Temp: 1.48 V / NTC: 1.93 V / Pressure: 1.91 V / Acceleration (x,y,z): 1.49 V, 1.84 V, 2.02 V,  
Counter: 6 / Time(s): 3.00 / Temp: 1.47 V / NTC: 1.92 V / Pressure: 1.91 V / Acceleration (x,y,z): 1.48 V, 1.82 V, 2.01 V,  
Counter: 7 / Time(s): 3.50 / Temp: 1.47 V / NTC: 1.92 V / Pressure: 1.90 V / Acceleration (x,y,z): 1.48 V, 1.83 V, 2.00 V,  
Counter: 8 / Time(s): 4.00 / Temp: 1.45 V / NTC: 1.91 V / Pressure: 1.90 V / Acceleration (x,y,z): 1.47 V, 1.82 V, 1.97 V,  
Counter: 9 / Time(s): 4.50 / Temp: 1.47 V / NTC: 1.92 V / Pressure: 1.90 V / Acceleration (x,y,z): 1.47 V, 1.82 V, 1.96 V,
```

Serial Monitor -ikkunan tulostetietoja.

6. Mittaustietojen kuvaus

Counter:	Laskee CanSatin tekemien mittausten määrän.
Time:	Aika CanSatin käynnistämisestä sekunteina.
Temp:	Näyttää ensimmäisen lämpötila-anturin (LM35DZ) tiedot.
NTC:	Näyttää valinnaisen lämpötila-anturin (NTC) tiedot.
Pressure:	Näyttää paineanturin (MPX4115A) tiedot.
Acceleration:	Näyttää kiihtyvyyssanturin (MMA7361L) tiedot. Tämä anturi antaa kolme lukemaa, yhden jokaiselle akselille (x, y ja z). (ei käytössä, joten tiedot eivät pidä paikkansa)

Oletusarvoisesti näytteenottotaajuus on 2 Hz (kaksi riviä sekunnissa) ja mitatut arvot ilmoitetaan voltteina.

7. Testiohjelmalla voidaan ohjata lähettämällä merkki Serial Monitorin kautta. Syötä merkki komentoriville ja paina Enter-näppäintä.

Testiohjelma hyväksyy seuraavat komennot:

1 – Määrittää näytteenottotaajuudeksi 1 Hz (1 näyte sekunnissa).

2 – Määrittää näytteenottotaajuudeksi 2 Hz (2 näytettä sekunnissa).

5 – Määrittää näytteenottotaajuudeksi 5 Hz (5 näytettä sekunnissa).

R – Palauttaa laskurin nollaan.

V – Muuttaa kaikkien antureiden tulosteet volteiksi.

S – Muuttaa kaikkien antureiden tulosteet mitatuiksi arvoiksi, joiden yksikkö on celsius, pascal tai G (9,81N).

8. Testaa kaikkia antureita yrittämällä manipuloida niitä.

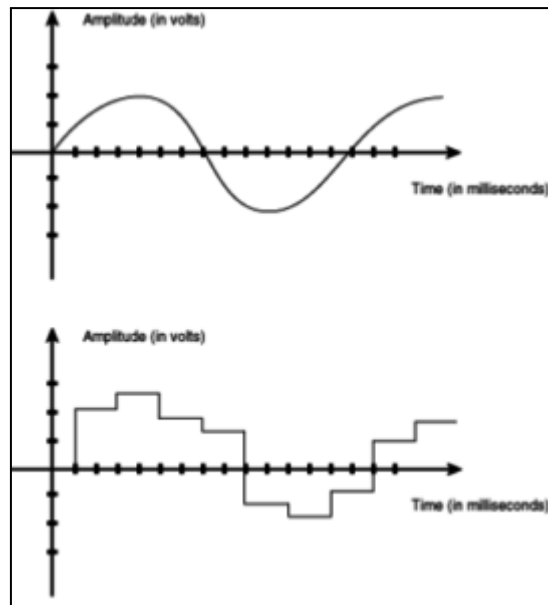
Reagoivatko kaikki anturit?

Jos saat tietoja kaikista antureista ja ne näyttävät toimivan oikein, päätoimintojen testaus on valmis. Radion ja dataloggerin toiminta testataan myöhemmin tässä oppaassa.

Huom. Antureissa voi olla poikkeamia todellisiin olosuhteisiin verrattuna. Ohjeet anturin kalibrointiin ovat [tässä](#).

Anturien käyttäminen

Johdanto



Analoginen ja digitaalinen signaali. Voit avata kuvasta suuremman version [täältä](#).

Tässä oppaassa käytettyyn CanSat-sarjaan sisältyy anturipiirilevy. Anturiin on kytketty neljä anturia: paineanturi, kaksi lämpötila-anturia ja kolmen akselin kiihtyvyysanturi (*ei käytössä*). Anturit synnyttävät jännitteen mittaamiensa parametrien arvojen perusteella. Anturien arvo voi olla mitä tahansa tietyllä arvoalueella. Tällaista signaalia kutsutaan analogiseksi signaaliksi ja se on kuvattu oikeanpuoleisen kuvan yläosassa. Mitatuista jännitteistä voidaan laskea niitä vastaavat arvot esimerkiksi muuntamalla jännite lämpöasteiksi.

Tässä osiossa kerrotaan, kuinka anturin mittaama analoginen jännite muunnetaan fyysiseksi määräksi, jolla on oikea yksikkö.

Analisisesta digitaaliseksi

Kaikilla digitaalisilla komponenteilla on yksilölliset signaalit. Digitaalinen signaali poikkeaa analogisesta signaalista siten, että digitaalinen signaali voi saada vain tiettyjä erillisiä arvoja. Oikeanpuoleisessa kuvassa on esitetty nämä kaksi erilaista signaalia. Ylemmässä kaaviossa on analoginen (jatkuva) signaali ja alemmassa kaaviossa on digitaalinen (erillinen) signaali.

Pöytätietokoneet ja kannettavat tietokoneet sekä CanSatin pieni tietokone (jota kutsutaan mikro-ohjaimeksi) voivat käsitellä vain digitaalisia signaaleja. Analoginen signaali voidaan muuntaa digitaaliseksi A/D-muuntimella (ADC), joka muuntaa analogisen signaalin digitaaliseksi signaaliksi.

A/D-muunnin on integroitu mikro-ohjaimeen ja siinä on kahdeksan sisääntulokanavaa. A/D-muunnin on 10-bittinen. Se muuntaa analogisen signaalin digitaaliseksi 10-bittisellä binaariluvulla. Yksi bitti edustaa yhtä binaarilukua, ja sen arvo voi olla 0 tai 1. Sitä voi edustaa myös *korkea* tai *matala jännite* tai *kytketty* ja *ei kytketty*. Ohjelmoinnissa arvoja 0 ja 1 edustavat usein myös *tosi* (1) ja *epätosi* (0).

Jokaisella binaariluvun numerolla voi olla kaksi arvoa, 0 tai 1. 10-bittisellä binaariluvulla voi olla $2^{10} = 1024$ eri arvoa, ja se voi edustaa kokonaislukuja 0–1023. Mikro-ohjain voi ymmärtää digitaalisia lukuja ja käyttää niitä laskutoimituksiin, joita voidaan ohjelmoida suorittimeen ohjelmakoodia kirjoittamalla. Yksi esimerkki koodista on TestShield-koodi, jota käytettiin tämän oppaan aiemmassa osassa.

A/D-muunnin lukee jokaisen CanSatin anturin ja muuntaa kaikki analogiset arvot 10-bittisiksi luvuiksi. 0 voltia on binaarilukuna 0000000000 = 0 ja 5 voltia on binaarilukuna 1111111111 = $2^{10} + 2^9 + 2^8 + 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 1 = 1023$.

Kun viiden voltin syöttöjännite kuvataan 1024 tasolla (0 mukaan lukien), tarkkuus on $5V/1023 = 4,89 \text{ mV}$. Tämä tarkoittaa, että 10-bittisellä A/D-muuntimella pienin jännitemuutos, jonka voimme mitata, on 4,89 mV. Tämä on tärkeää tietää, kun alamme työskennellä antureiden herkkyyden kanssa.

Analogisia (lämpötila-)antureita käytettäessä toimintajärjestys on seuraava:

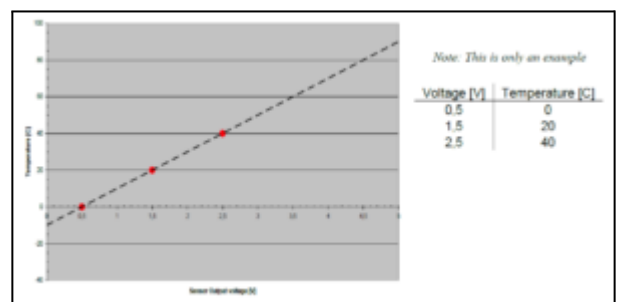
1. Lämpötila-anturi muuntaa mitatun lämpötilan jännitteeksi. Signaali on tässä vaiheessa analoginen.
2. Anturin analoginen signaali (jännite) on kytketty yhteen Arduino-piirilevyn analogisista porteista.
3. A/D-muunnin muuntaa analogisen signaalin digitaalseksi signaaliksi, jota suoritin voi käsitellä.
4. Mikro-ohjaimen sisällä signaali tallennetaan 10-bittiseksi binaariluvuksi, jota voidaan käyttää laskemisessa tai joka voidaan lähettää Arduino TX -portin ja radion tai USB-portin kautta ulkoiseen vastaanottimeen.

Antureiden kalibrointi

Joissakin tapauksissa siirtofunktio mitatun fyysisen parametrin (esimerkiksi lämpötilan) ja anturin lähtöjännitteen välillä on tuntematon ja vaikea laskea komponentin teknisten tietojen perusteella.

Yksinkertainen tapa esimerkiksi lämpötila-anturin siirtofunktion määrittämiseksi on mitata lähtöjännite tietyissä (3–4) määritetyssä lämpötilassa, tehdä tuloksista oikeanpuoleisen kuvan mukainen kaavio ja selvittää siirtofunktio regressioanalyysin avulla.

Käytössä olevan [NTC-anturin](#) tapauksessa voidaan olettaa, että suhde on lineaarinen tutkittavalla alueella. Linearisella regressioanalyysillä voidaan selvittää parhaiten sopiva lineaarinen yhtälö.



Lineaarisuus tarkoittaa, että jännitteen ja parametrin (esim. lämpötilan) suhde voidaan esittää lineaarisella kaavalla:

$$y = a \cdot x + b$$

Parametri (esim. lämpötila) = A · jännite + B,

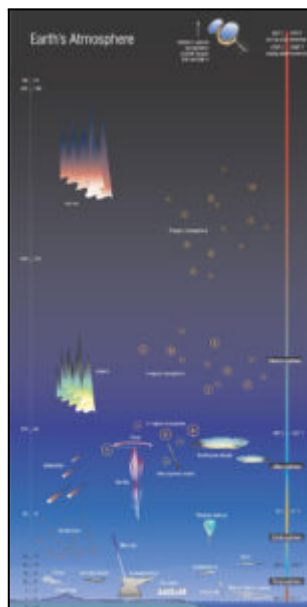
Käyrän avulla voidaan arvioida A:n ja B:n arvot. Tässä kaaviossa voit määrittää mitatut jännitteet x-akselille ja parametrin

(tässä tapauksessa lämpötilan) y-akselille. Kaavion vieressä olevassa taulukossa on tämän anturin esimerkkimittauksia. Nämä mittaustulokset on myös lisätty kaavioon. Toinen vaihe on piirtää pisteiden läpi suora viiva. Tee se käyttämällä CAS-työkalusi (esim. Excelin tai TiNspiren) ominaisuuksia, jotta tulos on mahdollisimman tarkka. Mitä useampia pisteitä käytät, sitä tarkempi tulos on. Viivan avulla voit määrittää edellisen lineaarisen vakiokaavan A:n ja B:n arvon.

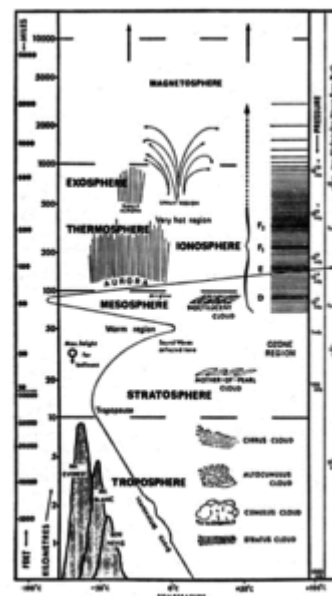
A: on viivan kulma

B: on leikkauspiste y-akselin kanssa.

Korkeuslaskelmat



Maapallon ilmakehä ([isompana](#))



Maapallon ilmakehän kerrokset ([isompana](#))

Ilmakehä on kaikkialla ympärillämme. Se on planeettaamme ympäröivä ohut kaasukerros. Ilmakehä koostuu pääosin typestä (78 %) ja hapesta (21 %). Jäljelle jäävä prosentti koostuu

vesihöyrystä, hiilidioksidista ja muista hivenkaasuista. Maapallon ilmakehässä on eri kerroksia, joista kaikilla on eri ominaisuudet (lämpötila, paine, koostumus jne.)

Eri kerrokset ja ihmisten ja sään eri toiminnot näissä kerroksissa on esitetty kahdessa oheisessa kuvassa.

Toisin kuin CanSat, useimmat satelliitit toimivat eksosfäärissä. Siellä ilmakehän tiheys on hyvin alhainen. CanSat kuitenkin toimii troposfäärissä, joka on ilmakehän alin kerros. Troposfäärissä on noin 80 prosenttia ilmakehän koko massasta, ja se yltää 10 kilometrin korkeuteen. Suuri osa sääilmiöistä, joita näemme päivittäin (esimerkiksi tuuli ja pilvet), syntyy tässä kerroksessa.

Kuten oikeanpuoleisesta kuvasta näkyy, ilmakehän lämpötila ja paine vaihtelevat korkeuden mukaan.

Ympäristön lämpötila voi kuitenkin nousta ja laskea siirryttäessä ilmakehän kerrosten välillä.

Troposfäärissä lämpötilan ja korkeuden suhde on lineaarinen. Kun nousee kilometri merenpinnasta, lämpötila laskee keskimäärin 6,5 celsiusastetta.

Seuraava yhtälö antaa suhteen:

$$h = h_1 + \frac{T - T_1}{\alpha}$$

missä

T – lämpötila kelvineinä

T_1 – alkulämpötila h_1 korkeudessa

h – korkeus metreinä

h_1 – alkukorkeus

α – lämpögradientti: $- 0.0065 \text{ K/m}$

Paineen ja korkeuden suhde on jonkin verran monimutkaisempi. Paine vaihtelee paitsi korkeuden myös lämpötilan mukaan.

Aloitetaan paineen ja lämpötilan suhteella:

$$\frac{p}{p_1} = \left(\frac{T}{T_1}\right)^{-\frac{g_0}{\alpha R}}$$

missä

p – paine pascaleina

p_1 – alkupaine pascaleina

g_0 – putoamiskiihtyvyys: $9,81 \text{ m/s}^2$

R – tietty kaasuvakio: $287,06 \text{ J/kg} \cdot \text{K}$

Kun tämä kaava lisätään lämpötilan ja korkeuden suhteeseen, saamme seuraavan lausekkeen korkeudelle lämpötilan ja paineen funktiona:

$$h = \frac{T_1}{\alpha} * \left(\left(\frac{p}{p_1}\right)^{-\frac{\alpha R}{g_0}} - 1 \right) + h_1$$

Radion käyttäminen

Johdanto

Telemetria on tekniikka, jolla voidaan siirtää tietoja mittalaitteista etäyhteyden avulla. Termi "telemetria" tulee kreikan sanoista "tele", joka tarkoittaa kaukaista, ja "metron", joka tarkoittaa mittaa. Telemetria on olennainen osa raketti- ja satelliittiteknologiaa. Tiedot lähetetään langattomasti radioaalloilla. Maassa radiovastaanottimet keräävät signaalit. Suurilla avaruusjärjestöillä on maa-asemien verkostoja, jotka kattavat koko maapallon. Ne seuraavat, valvovat ja vastaanottavat satelliittien telemetriatietoja.



Oppilaat seuraavat laskeutuvaa CanSatia ja vastaanottavat radiosignaaleja CanSatista antennin avulla. [\(kuva isompana\)](#)

Telemetriatiedot voidaan jakaa kahteen ryhmään: sisäisistä ja ulkoisista lähteistä saatuihin tietoihin. Raketit ja satelliitit on varustettu lukemattomilla antureilla, jotka mittaavat sisäisiä parametreja. Niiden mittaukset saattavat liittyä lämpötilaan, paineeseen, korkeuteen, virrankäyttöön ja muihin kohteisiin. Näiden anturien tietoja kutsutaan "ylläpitotiedoiksi". Niiden avulla valvotaan satelliittien kuntoa, mikä on olennaista järjestelmän toiminnan kannalta.

Tutkijoita kiinnostavat eniten ulkoisista lähteistä saatavat tiedot. Nämä tiedot kerätään hyötykuormasta. Tutkimussatelliitin hyötykuorma sisältää yleensä antureita ja muita laitteita, jotka mittaavat ja luovat tietoja planeetastamme, avaruusympäristöstä, auringosta, tähdistä ja muista asioista, jotka vaihtelevat tehtävän mukaan. Näitä tietoja kutsutaan "tehtävän tiedoiksi" tai "tutkimustiedoiksi". CanSatin tapauksessa tällaisia tietoja olisivat anturipiirilevystä saatavat tiedot. Nämä tiedot lähetetään maa-asemaan tutkijoiden tarkasteltavaksi.

CanSatin telemetriatoiminnot voidaan jakaa kolmeen erilliseen komponenttiin: tietojen lähettämiseen, vastaanottamiseen ja käsittelyyn.



CanSat-sarjan radiomoduulit [\(kuva isompana\)](#)

CanSatin lähetinpiirilevy kerää tietoja ja lähettää ne radiosignaalina.

Maa-asema vastaanottaa tämän signaalin ja lähettää sen kannettavaan tietokoneeseen, mihin se tallennetaan ennen kuin tiedot käsitellään tutkimustietoina.

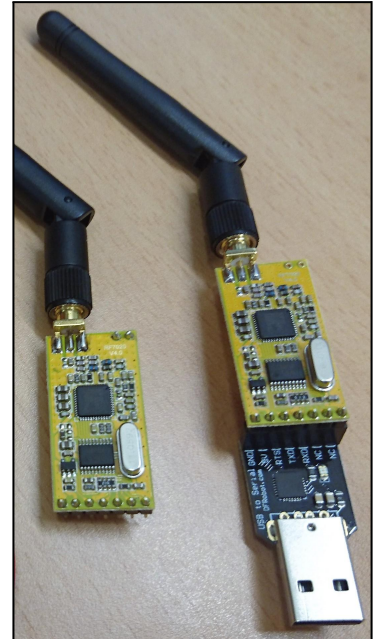
Lähetin-vastaanotin

Lähetin-vastaanotin on laite, jossa on samaan koteloon ja samoihin piireihin yhdistetty sekä lähetin että vastaanotin. Lähetyksen ja vastaanotto toimintojen piirit voivat olla myös erilliset.

Arduino CanSat -sarjassa käytetään [APC220:tä](#), joka on lähetin-vastaanotin. Se on erittäin monipuolinen, mutta vähän virtaa käyttävä ratoratkaisu, joka on helppo asentaa ja integroida mihin tahansa projektiin, joka edellyttää langatonta radiolinkkiä.

Se sopii täydellisesti kauko-ohjaukseen robottisovelluksella.

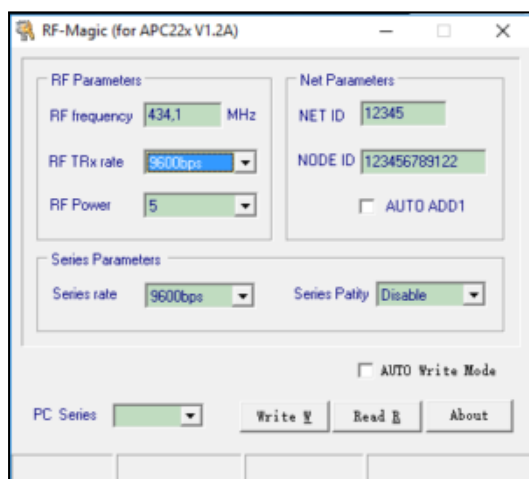
Voit kytkeä tällaisen moduulin mikro-ohjaimeen TTL-käyttöliittymän kautta. Voit myös kytkeä tietokoneesi toiseen APC220-moduuliin suoraan TTL-/USB-muuntimen kautta.



APC220-radiomodulit. Yhdessä on TTL-/USB-liitäntä tietokoneeseen kytkemistä varten ([kuva isompana](#))

Lähetin-vastaanotinlaitteiston asentaminen

TÄTÄ KOHTAA EI TEHDÄ!



Lähetin-vastaanottimet voidaan konfiguroida kahdella tavalla. Ne voidaan ohjelmoida suoraan Arduino-piirilevystä, tai voit ohjelmoida ne [RF-Magic](#)-ohjelmistolla. Vasemmalla olevassa kuvassa näkyy RF-Magic-ohjelman aloitusnäyttö.

RF-Magic-ohjelmassa on paljon toimintavirheitä, eikä se toimi oikein kaikissa tietokoneissa.

On suositeltavaa käyttää radion ohjelmointiin Arduinoa seuraavien ohjeiden mukaisesti.

RF-Magic-ohjelman aloitusnäyttö. ([kuva isompana](#))

Ohjainten asentaminen

Maa-aseman lähetin-vastaanotin yhdistetään tietokoneeseen USB–TTL-muuntimella. Tämä moduuli ei toimi oikein ilman ohjainta.

Voit ladata USB–TTL-moduulin uusimmat VCP (Virtual Com Port) -ohjaimet osoitteesta Silabs.com (CP210x USB–UART-sillan VCP-ohjaimet).

Napsauta linkkiä ja lataa käyttöympäristösi sopiva tiedosto.

Kun olet ladannut tiedoston, pura se ja aja CP210xVCPInstaller_x64.

On suositeltavaa asentaa ensin USB–TTL-muuntimen ohjaimet ja kytkeä sitten USB-muunnin ja siihen liitetty APC-moduuli. Kun moduuli on kytketty, Windows tunnistaa sen ja määrittää tarvittavat ohjaimet. Jos Windows ei tunnista laitetta tai ilmoittaa, ettei laite toimi kunnolla, irrota laite USB-portista ja kytke se uudelleen.

Lähetin-vastaanottimien ohjelmointi Arduinolla

Kytke Arduino-piirilevy tietokoneeseen ja lataa (ja pura) tietokoneeseen ohjelma `apc220cfg.ino`. Voit ladata sen [tästä](#).

Lataa ohjelma tietokoneeseen ennen kuin yrität kytkeä lähetin-vastaanottimen Arduino-piirilevyyn. Irrota USB-johto (ja paristo) Arduino-piirilevystä ja kytke lähetin-vastaanotin Arduinoon oikealla olevan kuvan mukaisesti.

Lähetin-vastaanotin kytketään Arduino-piirilevyn nastoihin, joissa on merkinnät GND, 8, 9, 10, 11, 12 ja 13.

```

APC version: 353
Jan 27 2013

commands:
* r Read apc220 radio config
* i go into echo mode: receive char, add 1 and return
* s no more echo - back to normal
* w Write apc radio config ...

"m" FFFFFFF 8 8 0 - number of letters indicates precise number of digits
FFFFFF: frequency: 434000 (434 MHz) range 418000-450000
R:   RF data rate - 1/2/3/4 equals 2400/4800/9600/19200bps
P:   Radio output power - 0... 9 9 equals 100m(100mW)
R:   UART baudrate - 0/1/2/3/4/5/6 equals 1200/2400/4800/9600/19200/38400/57600bps
C:   Byte Clock Parity - 0/1/2 equals NoClock(RN1)/EvenParity(RN1)/OddParity(RN1)

Write example: w 434000 3 9 3 0 1a...
434,000 MHz 9600 baud in air, 100m, 9600baud on UART, No Parity(RN1)
After 30 seconds with no keyboard input we will emit a char every two second

```



Lähetin-vastaanottimen kytkeminen Arduino Uno-piirilevyyn ([kuva isompana](#))

Lähetin-vastaanottimen määrittäminen ([kuva isompana](#))

Kytke Arduino-piirilevy USB-johdolla ja avaa Serial Monitor. Kirjoita ylhäällä olevalle komentoriville "m" ja paina Enter-näppäintä. Näyttöön tulee vasemmalla olevassa kuvassa näkyvä valikko. Voit avata kuvasta suuremman version napsauttamalla sitä.

Jos kirjoitat "r" ja paina Enter-näppäintä, ohjelma palauttaa lähetin-vastaanottimen nykyiset määritykset.

Frequencies

Frequencies (MHz)	Team	Frequencies (MHz)	Team
433,050		433,950	
433,100		434,000	
433,150		434,050	
433,200		434,100	
433,250		434,150	
433,300		434,200	
433,350		434,250	
433,400		434,300	
433,450		434,350	
433,500		434,400	
433,550		434,450	
433,600		434,500	
433,650		434,550	
433,700		434,600	
433,750		434,650	
433,800		434,700	
433,850		434,750	
433,900		434,800	

Voit määrittää radion uudelleen kirjoittamalla "w" ja viisi tarvittavaa parametria. Erottele parametrit välilyönnillä.

CanSatille on määritetty taajuus etukäteen. Jos näin ei ole, vasemmalla on luettelo taajuuksista. Voit avata kuvasta suuremman version napsauttamalla sitä. Määritä siirtonopeudeksi 9 600 bittiä sekunnissa (bps).

Huomaa, että molempiin lähetin-vastaanottiin on määritettävä samat asetukset, jotta voit käyttää niitä yhdessä.

CanSat-taajuudet ([kuva isompana](#))

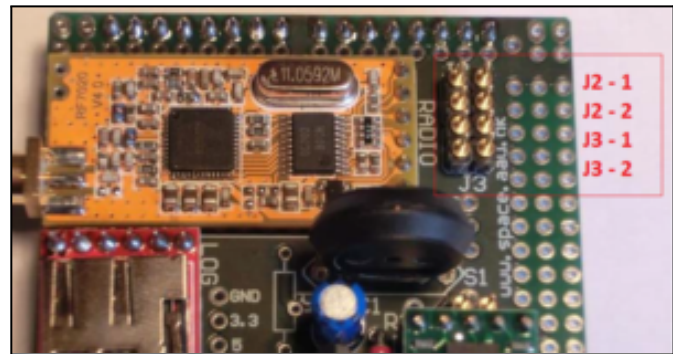
CanSat-laajennuspiirilevyn radio

CanSat-laajennuspiirilevyssä on oikeanpuoleisen kuvan mukaisesti kaksi hyppyjohdinsarjaa, J2 ja J3.

Kun J2- ja J3-hyppyjohtimia ei ole kytketty, voit ohjelmoida Arduinoa.

Kun asetat hyppyjohtimet kohtiin J2-1 ja J2-2, voit lähettää tietoja radiolla.

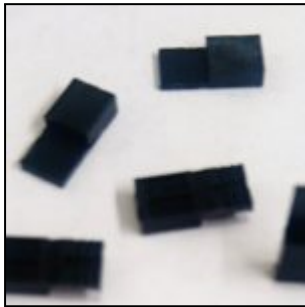
Kun asetat hyppyjohtimet kohtiin J3-1 ja J3-2, voit ottaa käyttöön tietojen tallentamisen SD-kortille (jos se on kytketty).



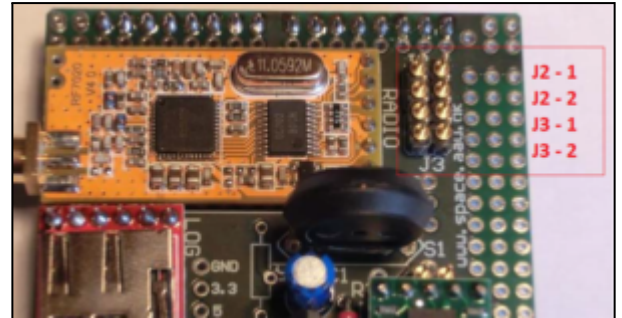
CanSat-laajennuspiirilevy, jossa on radio

Radion testaaminen

HUOM! TÄMÄ OHJE EI TOIMI (syy selvityksessä). TOIMI OHJEEN MUKAAN, MUTTA VALITSE ARDUINON SIJASTA TERMINAL-OHJELMA RADION TIETOJEN TARKASTELUUN.



Kun olet ohjelmoinut CanSat-sarjan, testaa, että radiolinkki toimii oikein. Aseta hyppyjohtimet J2:een, kytke vastaanotin tietokoneeseen USB–UART-liittimellä (USB-muunnin) ja avaa Arduino-ohjelmisto.



Muuta COM-portti Arduino Uno R3 (COMxx) -portista Silicon Labs CP210x USB to UART Bridge (COMxx) -porttiin

Arduino-ohjelmistossa valitsemalla Tools → Serial Port. Jos et muista, mikä on oikea COM-portti, tarkasta se Laitehallinnasta.

Tarkastele radiosta tulevia tietoja valitsemalla Serial Monitor. Tiedot tallennetaan Terminal by Br@y++ -ohjelmiston versiolla 1.9b.

Hyppyjohtimet

Tärkeä huomautus:

Jos Serial Monitorissa ei näy tietoja, syy voi olla, että USB-sarjaliitännän radiomoduulin käyttöönottonasta on kytketty pois käytöstä. Jos tämä on syy, tallennusohjelma [Terminal](#) toimii edelleen, mutta seuraavalla tavalla voit ottaa käyttöön tietojen siirron sillan läpi ja siirtää tietoja Arduino Serial Monitor -sovellukseen:

Irrota radiomoduuli USB–sarja-sillasta ja nosta/taita tai eristä radiomoduulin EN-nasta niin, ettei se kosketa sillan RTS-porttia. Älä irrota tai tuhoa nastaa, sillä saatat tarvita sitä myöhemmin. Kytke radiomoduuli uudelleen tietokoneeseen/siltaan ja avaa Serial Monitor uudelleen.



ettei se kosketa sillan RTS-porttia.

Taivuta radiomoduulin EN-nastaa niin,

Terminal-ohjelma

Terminal v1.9b by Br@y++

Terminal by Bray on ilmaisohjelma, joka voi lukea ja tallentaa tietoja sarjaportteista.

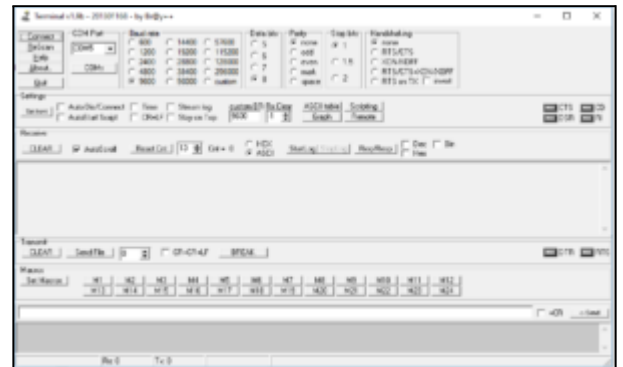
Voit ladata uusimman version [tästä](#).

Valitse oikea COM-portti ja valitse sitten ohjelman vasemmasta yläkulmasta Connect (Yhdistä).

Jos Arduino (tai radio) on kytketty, terminaaliin pitäisi tulla tietoja.

Jos haluat tallentaa tietoja, napsauta StartLog-painiketta.

Näyttöön tulee ponnahdusikkuna, jossa kysytään, mihin haluat tallentaa lokitiedoston. Valitse kansio ja tiedostonimi ja napsauta Open (Avaa) -painiketta.



Terminal-ohjelman aloitusnäyttö

(kuva isompana)

Sen jälkeen terminaali kirjoittaa kaikki sisään tulevat tiedot, kunnes painat StopLog-painiketta.

Kun lokitiedot on tallennettu, voit avata tekstitiedoston missä tahansa tekstieditorissa.

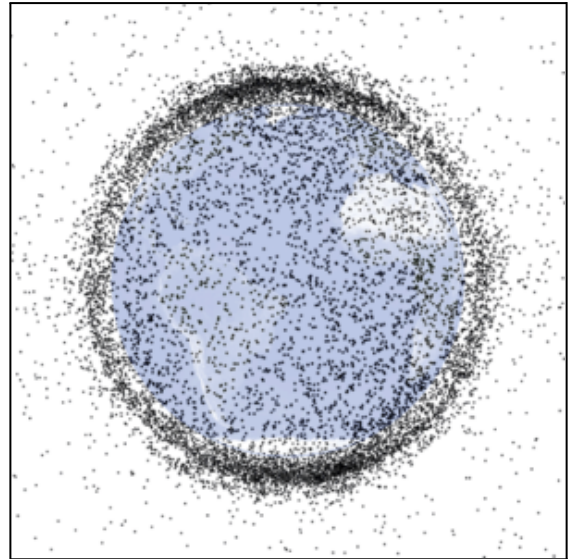
Jos tiedot ovat vain pilkuin eroteltuja numeroita, voit kopioida ne helposti Exceliin tai muihin laskentaohjelmiin.

Laskuvarjon suunnitleminen

Satelliitit eivät yleensä palaa maahan laskuvarjolla. Kun satelliittien käyttöikä on ohi, niiden kiertorataa muutetaan. Matalalla kiertävät satelliitit saattavat palaa ilmakehässä. Kauempana olevat satelliitit päätyvät kaukaiselle pysäköintikiertoradalle ja kiertävät planeettaamme ikuisesti.



*CanSat, johon on kiinnitetty laskuvarjo.
([kuva isompana](#))*



*Vanhat satelliitit ja muu avaruusromu
kiertävät maapalloa ([kuva isompana](#))*

Joskus avaruusalusten on kuitenkin palattava maapallolle näytteiden tai astronauttien kanssa. Yksi ratkaisu on käyttää laskuvarjoa.

Kun CanSat laukaistaan, siinä on oltava laite, joka hidastaa sitä, tai se murskautuu maahan. Laskuvarjolla voidaan myös varmistaa, että CanSat pysyy pystyasennossa. Tämä on erityisen tärkeää, koska se auttaa pitämään antennin oikeassa asennossa ja maksimoimaan telemetrian vastaanottomahdollisuudet. Seuraavilla sivuilla on vaiheittaiset ohjeet laskuvarjon suunnitteluun ja valmistamiseen.

Tarvittavat laskeutumisparametrit

Seuraavat arvot ovat alustavia ja voivat muuttua CanSat-kilpailun ja laukaisukampanjan edetessä. Niitä voidaan kuitenkin käyttää väliaikaisina ohjeina. Epävarmuustekijät laukaisukampanjassa voivat johtaa lopullisten laskeutumisnopeuksien muuttumiseen.

Vähimmäislaskeutumisnopeus: 8 m/s
Enimmäislaskeutumisnopeus: 11 m/s
Sallittu massa: 300–350 g
ilmanvastuskertoimet:

- Puolipallon muotoinen: 1,5
- Ristin muotoinen: 0,8
- Liitovarjo: vaihtelee muodon mukaan (voidaan määrittää testeillä)
- Tasainen, kuusikulmainen: 0,8

Laskuvarjon valmistaminen

Kun laskuvarjo on suunniteltu, voit aloittaa valmistusprosessin. Sen aikana on muistettava muutamia tärkeitä asioita. Laskuvarjon aukeaminen on melko raju tapahtuma, joten

laskuvarjo on valmistettava vahvasta kankaasta ja kuidusta. Yleensä leijakaupoista saa nailonnarua ja Ripstop-kangasta. Ne sopivat hyvin laskuvarjoihin.

Muista kangasta leikatessasi, että jotkin kankaat on käännettävä kaksin kerroin, että niitä voidaan ommella.

Lisää käteviä vinkkejä laskuvarjon valmistamiseen on [tässä](#).

Kun laskuvarjo on valmis, paras tapa tarkastaa sen toiminta on testata sitä.

Laskuvarjon suunnitteluun liittyvät laskelmat



Ristin muotoinen laskuvarjo.
Puolipallon muotoinen laskuvarjo.
([kuva isompana](#))



([kuva isompana](#))

Seuraavat laskelmat voidaan tehdä laskuvarjoa suunniteltaessa. Jotta voit tehdä nämä laskelmat, sinun on ehkä luettava seuraavat laskeutumisen fysiikkaa ja laskuvarjon suunnittelua koskevat osiot.

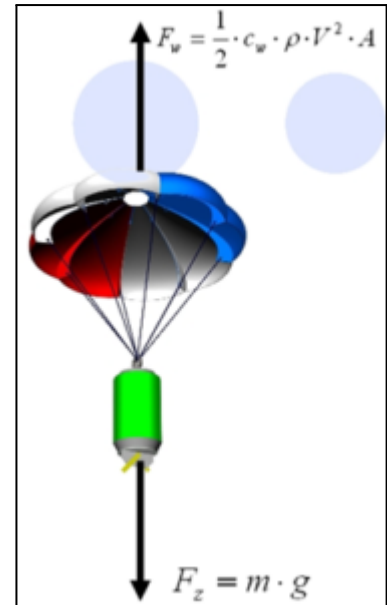
1. Laske tölkin törmäysnopeus ilman laskuvarjoa (kun se vapautetaan kilometrin korkeudelta). (Vastaus tähän kysymykseen tekee selväksi sen, miksi laskuvarjoa tarvitaan!)
2. Laske tarvittava laskuvarjon vähimmäispinta-ala, kun käytetään ristin muotoista laskuvarjoa. Minkä kokoisia laskuvarjon neliöiden pitäisi olla?
3. Tee samat laskelmat puolipallon muotoiselle laskuvarjolle. Mikä sen säde on?
4. Testaa laskuvarjon laskeutumisnopeus limsatölkillä, joka painaa saman verran kuin CanSat.
5. Kokeile erilaisia laskuvarjoratkaisuja. Ehkä laskuvarjo, jossa on reikiä tai useita pieniä laskuvarjoja? Molemmat ratkaisut parantavat CanSatin vakautta.

Laskeutumiseen liittyvä fysiikka

Ennen kuin laskuvarjon valmistaminen voidaan aloittaa, on selvitettävä laskuvarjon tarvittava koko. Tarkemmin ottaen on laskettava, kuinka suuri laskuvarjon pinta-ala on oltava, jotta se täyttää vaatimukset.

On loogista, että mitä suurempi laskuvarjo on, sitä hitaammin kappale laskeutuu. Myöhemmin tämä periaate todistetaan perusyhtälöillä.

Vaikka olisi hyödyllistä hidastaa CanSatin laskeutumista mahdollisimman paljon, olemme määrittäneet rajan, jotta CanSat laskeutuu mahdollisimman lähelle laukaisualuetta. Jos CanSat laskeutuu liian hitaasti, se voi leijua kilometrien päähän tuulen mukana, mikä ei ole sallittua eikä toivottavaa. Turvallisuussyistä myös enimmäislaskeutumisnopeus on määritetty.



Hyödynnämme laskuvarjon suunnittelussa yksinkertaista fysiikkaa. Arvioimme laskuvarjon pinta-ala yksinkertaistetun mallin avulla, minkä jälkeen voimme aloittaa varjon valmistamisen.

Laskeutumisen aikana CanSatiin vaikuttaa kaksi voimaa oikealla olevan kuvan mukaisesti. Painovoima vetää tölkkiä ja kiihdyttää sen nopeutta maata kohti, ja laskuvarjon ilmanvastus vaikuttaa CanSatiin päinvastaisesti ja hidastaa sen laskeutumista. Nämä voimat on esitetty oikealla olevassa kuvassa.

Kun CanSat on laukaistu, painovoima saa sen kiihtymään. Ilmanvastus vaihtelee CanSatin nopeuden mukaan. Kun nopeus on alhainen, ilmanvastus on pienempi kuin painovoima. Kun nopeus kasvaa, ilmanvastuskin lisääntyy, ja muutaman sekunnin kuluttua laskuvarjon ilmanvastus on yhtä suuri kuin painovoima. Sen jälkeen kiihtyvyys on nolla ja CanSat laskeutuu tasaisella nopeudella. Tämän vakionopeuden on oltava suurempi kuin vaatimuksissa määritetty vähimmäislaskeutumisnopeus. Seuraavia laskelmia varten käytämme tätä vähimmäisarvoa CanSatin vakionopeutena.

Painovoima F_g on yhtä suuri kuin: $F_g = m * g$

Tässä yhtälössä:

m : CanSatin massa

g : Putoamiskiihtyvyys, yhtä suuri kuin 9.81 m/s^2

Laskuvarjon ilmanvastus on yhtä suuri kuin:

$$F_D = 0.5 * C_D * \rho * A * V^2$$

Tässä yhtälössä:

A : Laskuvarjon kokonaispinta-ala (ei vain otsapinta-ala)

C_D : Laskuvarjon ilmanvastuskerroin. Tämä arvo vaihtelee laskuvarjon muodon mukaan.

ρ : Ilman paikallinen tiheys. Tämän oletetaan olevan vakio .

V : CanSatin laskeutumisnopeus

Kun kiihtyvyys on nolla, molemmat voimat ovat samankokoiset ja voimme kirjoittaa:

$$F_D = F_g$$

tai

$$m * g = 0,5 * C_D * \rho * A * V^2$$

Kun haluttu nopeus tiedetään, voimme helposti muokata yhtälöä ja laskea laskuvarjon tarvittavan pinta-alan.

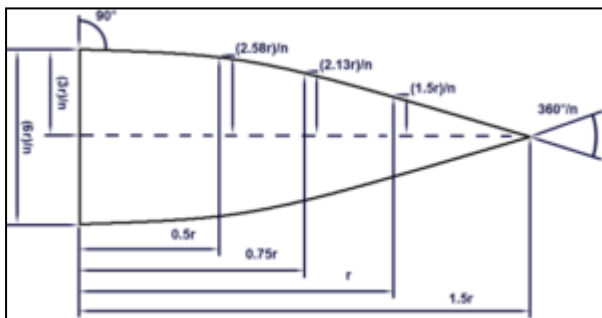
Puolipallon muotoinen laskuvarjo

Puolipallo on yleisin laskuvarjoissa käytettävä muoto.

Vaikka puolipallon muotoista laskuvarjoa ei ole vaikea tehdä, oikean muodon valmistaminen voi viedä aikaa. Oikealla oleva kuva auttaa. Kuvassa on käytetty kahta parametria:

n tarkoittaa tarvittavien osien määrää

r tarkoittaa laskuvarjon sädettä.



([kuva isompana](#))

*Puolipallon
muotoinen
laskuvarjo*



([kuva isompana](#))

Ristin muotoinen laskuvarjo

Voit valita puolipallon sijaan myös ristin muotoisen laskuvarjon. Tämän muodon etu on, että se on helppo valmistaa. Jos haluat tietää lisää ristin muotoisista laskuvarjoista, napsauta seuraavaa [linkkiä](#).



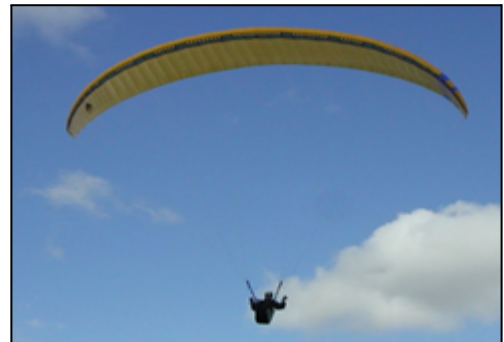
Ristin muotoinen laskuvarjo.

Liitovarjon muotoinen laskuvarjo

Liitovarjon muotoinen laskuvarjo toimii siiven tavoin. Muotonsa ansiosta sitä voidaan käyttää ohjaamiseen.

Liitovarjo on monimutkaisempi valmistaa kuin edellä mainitut muodot.

Joudut perehtymään laskuvarjon valmistamiseen, jos haluat käyttää tätä laskuvarjotyyppiä.

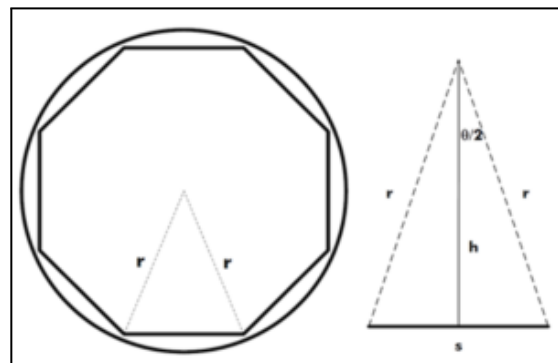


Liitovarjon muotoinen laskuvarjo.

Litteä laskuvarjo

Useimmat saatavilla olevat laskuvarjot on tehty tavallisista kaksiulotteisista ja litteistä geometrisistä kuvioista, kuten kuusi- ja kahdeksankulmioista.

Oikealla olevassa kuvassa on luonnos litteästä laskuvarjosta, joka koostuu kahdeksasta samankokoisesta kolmiosta.



*Kahdeksankulmainen laskuvarjo
([kuva isompana](#))*

Näin ollen laskuvarjon kokonaispinta-ala on

$$A = 8 * AT ,$$

Josta yhden kolmion pinta-ala on

$$AT = s * \frac{h}{2}.$$

Kun nämä kaksi yhtälöä yhdistetään, saadaan

$$A = \frac{8*s*h}{2}.$$

Voit lukea lisää litteän laskuvarjon pinta-alan laskemisesta [tästä](#).

Kun tiedät kokonaispinta-alan (A) ja ilmanvastuksen (C_D), voit helposti määrittää CanSatin laskeutumisnopeuden.

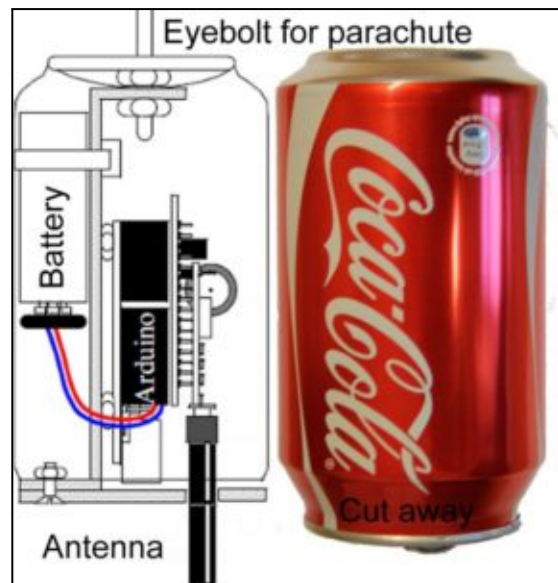
Jos et tiedä laskuvarjosi ilmanvastuskerrointa, voit tehdä CanSatilla pudotustestejä, kunnes tiedät rajanopeuden.

CanSatin rakenteen suunnittelu

Johdanto

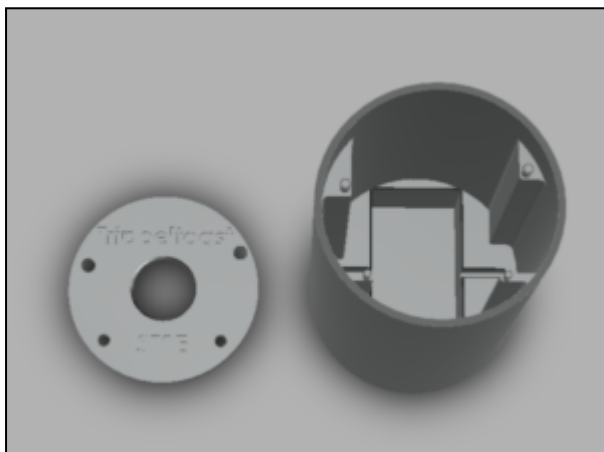
Ennen CanSatin laukaisemista sen sähköosat ja paristo on kiinnitettävä kiinnikkeeseen tai muulla tavalla, jotta ne kestävät laukaisun ja raketista irtoamisen aiheuttaman rasituksen.

Oikealla olevassa kuvassa on esimerkki piirros, jossa sähköosat on kiinnitetty tukevaan kiinnikkeeseen. Myös laskuvarjo on kiinnitettävä kiinnikkeeseen, tölkkiin. Tölkki on vain kuori, joka suojaa sähköosia suoralta iskulta.



ei

*CanSat kiinnikkeen kanssa.
([kuva isompana](#))*

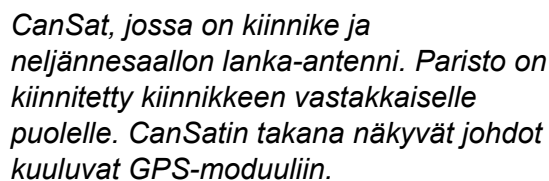


Toinen vaihtoehto on käyttää 3D-tulostettua koteloa, joka on suunniteltu pitämään sähköosat ja akun tiukasti paikoillaan. Norjalaisen yläkoulun (Bodin VGS:n) oppilaat [suunnittelivat kotelon](#) omaan CanSatiinsa. Se on vasemmalla olevassa kuvassa. Jos haluat suunnitella oman kotelon, kannattaa aloittaa tästä [mallitiedostosta](#). [Tästä](#) löydät lisää esimerkkejä 3D-tulostetuista mekaanisista rakenteista.

Kolmiulotteinen CanSat-kotelo, jonka ovat suunnitelleet Bodin VGS:n opiskelijat. ([kuva isompana](#))

CanSatin idea on, ettei se saa olla tavallista limsatölkkiä painavampi tai kookkaampi. (Alustavat) vaatimukset on lueteltu [tässä](#).

Antennin on oltava lanka-antenni, koska se on joustava. Näin se ei vahingoitu törmäyksessä, kun CanSat osuu maahan.



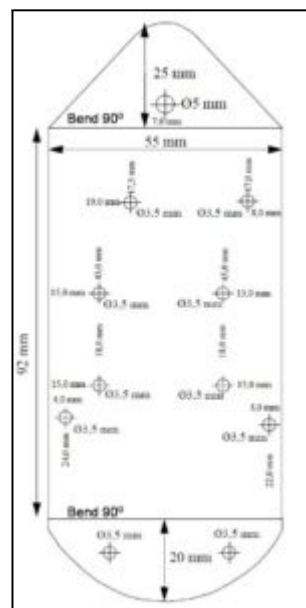
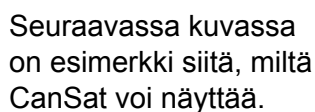
Kiinnike

Oikealla olevassa kuvassa on kiinnike mittoineen sekä reikien asemat ennen kiinnikkeen taivuttamista. Levy on suositeltavaa tehdä 0,5–1 mm paksusta alumiinista.

Levy taivutetaan yläosasta sisäänpäin ja alaosa ulospäin. Levyn taivutuskohdat näkyvät kuvassa.

Muista reiät nauhoille, joilla paristo kiinnitetään.

Jotta CanSat on helppo käynnistää ja sulkea, on suositeltavaa asentaa kytkin pariston ja CanSat-laitteen väliin. Muista kuitenkin, että kytkin voi olla heikko lenkki laukaisun aikana.



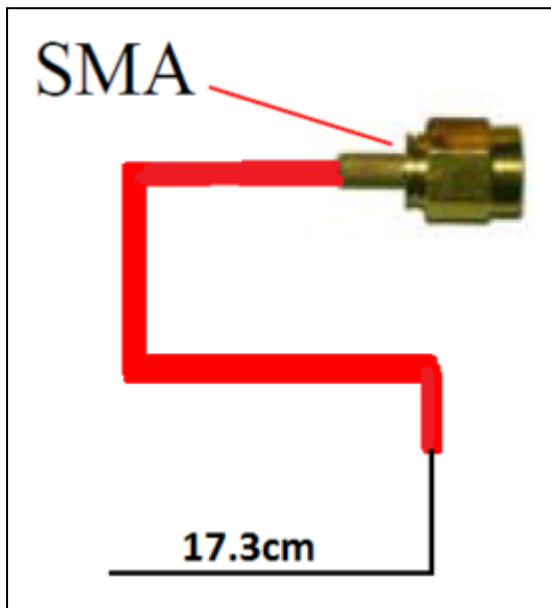
CanSat ja mekaaninen rakenne valmiina
tölkkiin laitettavaksi CanSat-kiinnikkeen
rakenne (kuva isompana)

CanSat-kiinnikkeen rakenne.
(kuva isompana)

Antennin rakenne

APC220-radion antenni on joustava 433 MHz:n antenni. Se on kestävä ja hyvä testaamiseen, mutta ei yleensä mahdu limsatölkkin sisään.

Yksinkertaiset lanka-antennit ovat hyvä vaihtoehto, ja ne voidaan kiinnittää SMA-liittimeen tai juottaa suoraan lähettimen ulostuloon. Tavallisesti nämä antennit ovat neljännesaallon antenneja. Paksuus ei ole tärkeä tekijä – tärkeintä on joustavuus ja kestävyys.



APC220-radiomoduulien joustava antenni.

Antennin pituus voidaan laskea seuraavasta yhtälöstä, kun taajuus (f) ja valon nopeus (c) tiedetään:

Jos taajuus f on 434 MHz, neljännesaallon pituus lasketaan yhtälöllä:

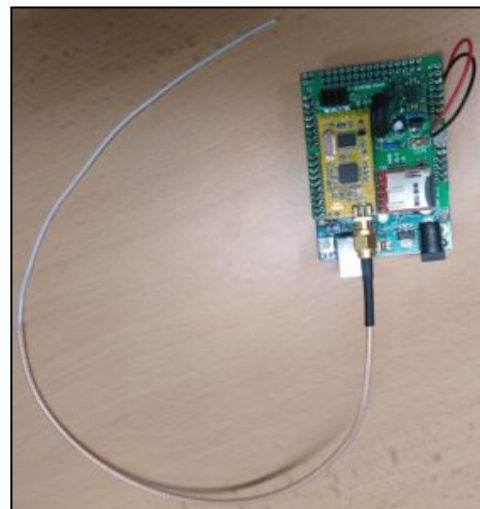
$$L = \frac{c}{4 \cdot f} = \frac{3 \cdot 10^8 \text{ m/s}}{4 \cdot 434 \cdot 10^6 \text{ Hz}} = 0.173 \text{ m}$$

Tämän laskutoimituksen mukaan antennin on oltava 17,3 cm pitkä.

Neljännesaallon lanka-antenni ([kuva isompana](#))

Voimme rakentaa antennin koaksiaalikaapelista. Poista kaapelin muovikuorta ja metallista suojajohdinta 17,3 senttimetrin matkalta, mutta jätä kaapeliin sisäjohtin ja dielektrinen eriste. Varmista, että kaapelin johdinosa yltää tölkkiin saakka ennen kuin kuorit sen.

Oikealla CanSat neljännesaallon lanka-antennilla ([kuva isompana](#))



Toissijainen tehtävä

Esimerkkejä CanSatin toissijaisesta tehtävästä ja antureista:

- toteuta turvavyöjärjestelmä laskeutumista varten
- kiihtyvyyssanturi
- tietojen kirjaaminen piirilevyyn asennetulle MicroSD-kortille
- GPS
- edistynyt telemetria (kaksisuuntainen tiedonsiirto)
- magnetometri asennon mittaamiseen
- kaasuanturit (CO2 tai muu kaasu)
- valokuva- tai videokamera (tai kaksi kolmiulotteisia kuvia varten).

CanSat-perussarjaan kuuluu jo analoginen 3-akselinen kiihtyvyyssanturi ja integroitu dataloggeri (openlogger).

Käytämme esimerkkinä CanSatiin ja Arduino-ohjelmointiin sopivasta GPS-moduulista GY-GPS6MV2-moduulia (U-blox NEO-6M-0-001), jota käsitellään [tässä](#).

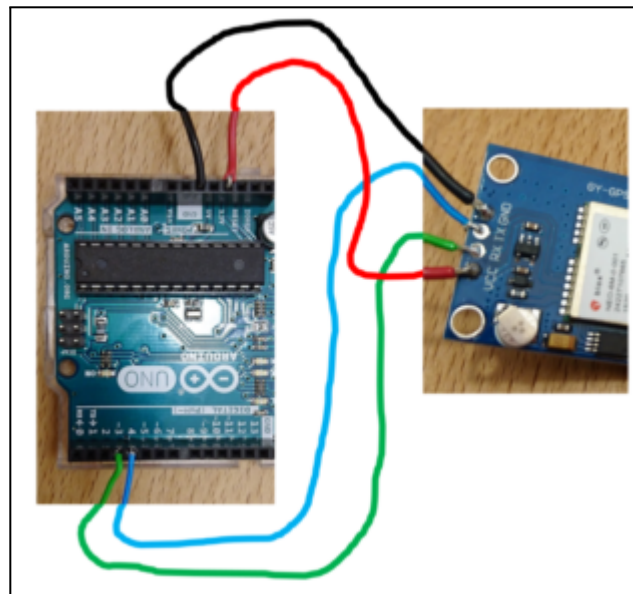
GPS-moduuli

Markkinoilla on monia pieniä CanSatiin ja Arduino-ohjelmointiin sopivia GPS-moduuleja. Selitämme tässä, kuinka U-blox NEO-6M-0-001 -GPS-moduulia voidaan käyttää melko helposti Arduino Unon ja CanSat-sarjan kanssa.

Jotta U-blox NEO-6M-0-001 -moduulia olisi helpompi käyttää, voit tilata antennilla varustetun GY-GPS6MV2-laajennuskortin.

GPS-moduulin käyttäminen Arduino Unon kanssa

Arduino Unon ja GPS-moduulin välille tarvittavat liitännät ovat maadoitus, syöttöjännite (3,3 V), Tx ja Rx.



*U-blox NEO-6M-0-001 -GPS-moduuli kytkettynä Arduino Unoon kahdesta sarjaportista (Rx D4:ssä ja Tx D3:ssä).
([kuva isompana](#))*

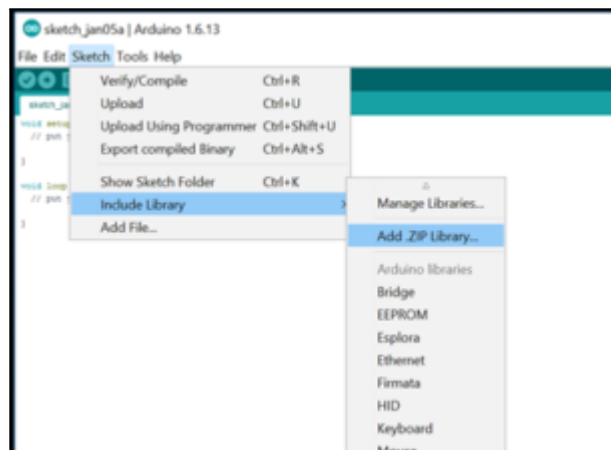
GPS module	Arduino Uno
Vcc	3.3 V
RX	digital port 3 (Tx)
TX	digital port 4 (Rx)
GND	GND

Kuten vasemmalla olevassa taulukossa lukee, moduulin Tx on kytkettävä Arduinon Rx-porttiin ja GPS-moduulin Rx-portti on kytkettävä Arduinon Tx-porttiin. Koska käytämme Rx/Tx-portteja radio- ja USB-tiedonsiirtoon ja tietojen kirjaamiseen, käytämme niiden sijaan ohjelmistopohjaisia sarja portteja. Käytämme Arduino Unossa

digitaalista porttia 3 ohjelmistopohjaisena Tx-sarjaporttina ja digitaalista porttia 4 ohjelmistopohjaisena Rx-sarjaporttina. Katso oikealla oleva kuva.

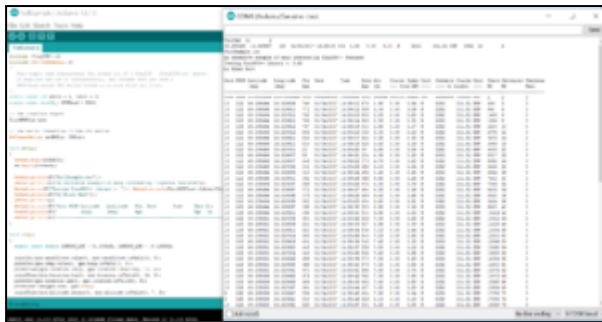
Ota seuraavien ohjeiden avulla GPS käyttöön niin, että se lähettää Serial Monitoriin tiedot koordinaateista, korkeudesta ja satelliittien määrästä:

1. Lataa ja tallenna pakattu TinyGpsPlus-kansio tietokoneesi Arduino-kirjastoon. Katso oikealla oleva kuva. Lataa tarvittava kirjasto seuraavalla linkillä: [TinyGPSPlus](#).
2. Kopioi seuraavan tekstitiedoston teksti: [tiny_gps_1](#) tyhjään Arduino-luonnokseen ja käännä luonnos, jotta kirjasto on varmasti lisätty oikein.
3. Kytke Arduino Uno USB-johdolla tietokoneeseen ja lataa luonnos
4. Arduino Unoon.
5. Irrota Arduino Uno tietokoneesta.
6. Kytke GPS-moduuli Arduino Unoon.
7. Kytke Arduino Uno uudelleen tietokoneeseen. Jos kytkennät on tehty oikein, GPS-moduulin punaisen valon pitäisi vilkkua.
8. Avaa Serial Monitor ja seuraa GPS-moduulin tietoja. Esimerkki tiedoista on alaoikealla olevassa kuvassa. GPS:n ja Arduinon on oltava käynnissä joitakin minuutteja ennen kuin GPS luo yhteyden riittävän moneen satelliittiin (yleensä neljään), jotta se voi ilmoittaa GPS-sijainnin.



*Pakatus kirjaston lisääminen Arduinoon
(kuva isompana)*

Jos se toimii oikein: Katso parhaillaan käyttämäsi Arduino-ohjelman rivejä. Siellä on useita kommenttirivejä. Kyseisillä riveillä on tulostusmahdollisuuksia, jotka voidaan helposti lisätä Serial Monitor -tulosteeseen.



Esimerkki Serial Monitorin tulosteesta, jossa käytetään ohjelman kokoversiota. ([kuva isompana](#))

Voit ladata tästä monimutkaisemman version, jossa on muun muassa tarkistussumma: [gps_full_version](#). Vasemmalla olevassa kuvassa on esimerkki Serial Monitorin tulosteesta, jossa käytetään ohjelman kokoversiota.

```
LAT=69.295516
LONG=16.029972
ALT=13.30
Number of satellites in use    9

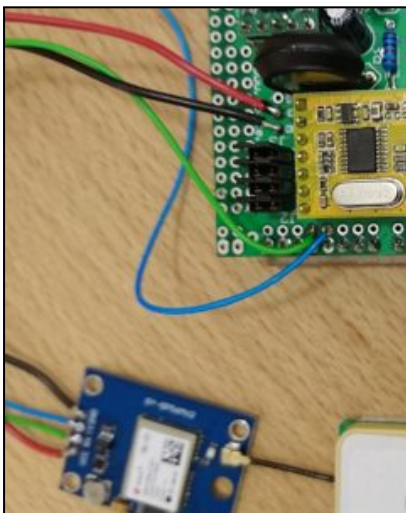
LAT=69.295509
LONG=16.029975
ALT=12.30
Number of satellites in use    9

LAT=69.295509
LONG=16.029977
ALT=11.40
Number of satellites in use    9
```

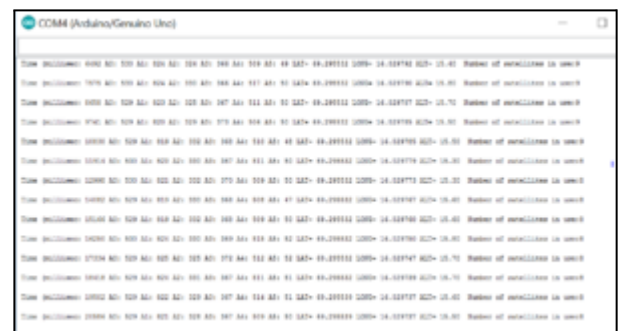
Serial Monitor -tuloste
GPS-moduulista ([kuva isompana](#))

Jos haluat lisätietoja TinyGpsPlus-ohjelmasta ja muista Arduino-ohjelmista, joita käytettiin tämän osion inspiraationa ja lähteenä, napsauta tätä [linkkiä](#).

CanSat-sarja ja GPS-moduuli.



Jos haluat tulosteen sekä Arduino-sarjan antureista että GPS-moduulista, Arduino-ohjelmat on yhdistettävä ja GPS voidaan juottaa laajennuspiirilevyyn vasemmalla olevan kuvan mukaisesti. Voit ladata melko yksinkertaisen ohjelman, joka yhdistää GPS-moduulin ja CanSat-sarjan anturit, [tästä](#). Ohjelma lähettää Serial Monitoriin oikealla olevan kuvan mukaisen tulosteen. Voit avata kuvasta suuremman version napsauttamalla sitä.



CanSat-sarjan antureiden ja
GPS-moduulin yhdistelmä. Tuloste
Serial Monitoriin ([kuva isompana](#))

/* This program is intended for use as a test program for the CanSat course at NAROM, Andøya Rocket Range 2013.

** Beerware, this code is free to use, but if you use it and happen to meet me, you can buy me a beer

** by Thomas Gansmoe, Narom AS (www.narom.no / www.rocketrangle.no) */

```
#define BITRATE 9600      //Bitrate for the serial interface (Radio and logger)
#define Vcc 5.0           //Supply voltage
```

```
// **** [ Sensor calibration ] ****
```

```
// NTC-sensor variables
```

```
#define NTC_Grad -20      //The gradient value for the NTC sensor
```

```
#define NTC_Off 75       //The offset value for the NTC sensor
```

```
// Accelerometer-sensor settings
```

```
#define AccSensMode 0     //Accelerometer sensitivity mode. 0 = +- 3g, 1 = +- 9g (Not implemented in the code yet)
```

```
#define Acc3 0.440       //Acc sensitivity mode 0 = 440 mV/g (from datasheet)
```

```
#define Acc9 0.118       //Acc sensitivity mode 1 = 118 mV/g (from datasheet)
```

```
#define Acc0 1.65        //Acc 0g voltage (from datasheet)
```

```
#define AccxOff 0.0      //Offset values - Calibrated by finding voltage at zero G.
```

```
#define AccyOff 0.0
```

```
#define AcczOff 0.0
```

```
// LM335 Temp sensor
```

```
#define TmpOffset 0.0     //Volts at zero degree Celsius
```

```
#define TmpSens 100      //Sensitivity. Deg.Celsius /volt
```

```
/* **** */
```

```
int LOOPTIME = 500;      //Interval time in milliseconds
```

```
//Configure sensor output format
```

```
int OutFormatTmp = 0;    //Sets the output format. (0 = Volt, 1 = Deg.C)
```

```
int OutFormatNTC = 0;    //Sets the output format. (0 = Volt, 1 = Deg.C)
```

```
int OutFormatPressure = 0; //Sets the output format. (0 = Volt, 1 = hPa)
```

```
int OutFormatAcc = 0;    //Sets the output format. (0 = Volt, 1 = G)
```

```
unsigned long time;
```

```
long int counter=0, number=0;
```

```
float G = 9.81;
```

```
void setup() {
```

```
    Serial.begin(BITRATE);
```

```
}
```

```
float SecFromStart(){
```

```
    time = millis();
```

```
int sec = time/1000;
int des = (time%1000)/10;
float result = (float)sec+((float)des)/100;
return result;
}

void printdata(){
  Serial.print("Counter: ");
  Serial.print(counter);
  Serial.print(" | Time[s]: ");
  Serial.print(SecFromStart());

  PrintTmp(OutFormatTmp);
  PrintNTC(OutFormatNTC);
  PrintPressure(OutFormatPressure);
  PrintAcc(OutFormatAcc);

  Serial.println();
}

float BitToVolt(int n){ //Function to convert raw ADC-data (0-255) to volt
  int raw = analogRead(n);
  float volt = (float)raw*5.000/1023;
  return volt;
}

// The LM335 temp sensor
void PrintTmp(int Format){
  analogReference(INTERNAL);
  analogRead(5);
  delay(10);
  //float Tmp, volt = BitToVolt(5); //Reads Analog5
  float Tmp, volt = analogRead(5)*1.1/1023;
  analogReference(DEFAULT);
  analogRead(5);
  delay(10);
  Serial.print(" | Temp: ");
  switch (Format){
    case 0:
      Serial.print(volt,3);
      Serial.print(" V");
      break;
    case 1:
      Serial.print((volt - TmpOffset)*TmpSens);
      Serial.print(" Deg.C");
      break;
  }
}

// The NTC sensor
void PrintNTC(int OutFormatNTC){
```

```
float volt = BitToVolt(0);
Serial.print(" | NTC: ");
switch (OutFormatNTC){
  case 0:
    Serial.print(volt);
    Serial.print(" V");
    break;
  case 1:
    Serial.print(NTC_Off + NTC_Grad * volt);
    Serial.print(" Deg.C");
    break;
}
```

```
// The Pressure sensor (MPX4115)
void PrintPressure(int OutFormatPressure){
  float volt = BitToVolt(1), hPa;
  Serial.print(" | Pressure: ");
  hPa = 10*(volt/(0.009*Vcc)+(0.095/0.009));

  switch (OutFormatPressure){
    case 0:
      Serial.print(volt);
      Serial.print(" V");
      break;
    case 1:
      Serial.print(hPa);
      Serial.print(" hPa");
      break;
  }
}
```

```
// The Accelerometer sensor (MMA7361L)
float AccCalc(float volt){
  float mv2, gradient;
  if (AccSensMode == 0){ gradient = Acc3;}
  else {gradient = Acc9;}
  mv2 = (volt-Acc0)/gradient;
  return mv2;
}
```

```
void PrintAcc(int OutFormatAcc){
  float voltx = BitToVolt(2)+AccxOff, volty = BitToVolt(3)+AccyOff, voltz =
  BitToVolt(4)+AcczOff, Gx, Gy, Gz;
  Serial.print(" | Acceleration [x,y,z]: ");
  Gx = AccCalc(voltx);
  Gy = AccCalc(volty);
  Gz = AccCalc(voltz);

  switch (OutFormatAcc){
    case 0:
```

```
    Serial.print(voltx);
    Serial.print(" V, ");
    Serial.print(volty);
    Serial.print(" V, ");
    Serial.print(voltz);
    Serial.print(" V, ");
    break;
case 1:
    Serial.print(Gx);
    Serial.print(" G, ");
    Serial.print(Gy);
    Serial.print(" G, ");
    Serial.print(Gz);
    Serial.print(" G ");
    break;
}
}

void ReceiveSerial(){
    while (Serial.available()){
        delay(10);
        char RxChar = (char)Serial.read();
        switch(RxChar){
            case '1':
                LOOPTIME=1000;
                break;
            case '2':
                LOOPTIME=500;
                break;
            case '5':
                LOOPTIME=200;
                break;
            case 'R':
                counter = 0;
                break;
            case 'V':
                OutFormatNTC = 0;
                OutFormatAcc = 0;
                OutFormatPressure = 0;
                OutFormatTmp = 0;
                break;
            case 'S':
                OutFormatNTC = 1;
                OutFormatAcc = 1;
                OutFormatPressure = 1;
                OutFormatTmp = 1;
                break;
        }
    }
}
```

```
void loop() {  
  long int loop_start = millis(), loop_end;  
  printdata();  
  ReceiveSerial();  
  counter++;  
  loop_end = millis();  
  if (LOOPTIME > (loop_end - loop_start)) { //Sets the delay to acquire right loop time  
    delay(LOOPTIME - (millis() - loop_start));  
  }  
}
```