**Ubuntu og VirtualBox**

Ubuntu er et ”open source” operativsystem for datamaskiner, publisert av Canonical Ltd. Operativsystemet bygger på Linuxkjernen, og er per dags dato en av de mest populære Linux-distribusjonene. Det er gratis å bruke, som et resultat av at frivillige utviklere over hele verden bidrar til å utvikle systemet.

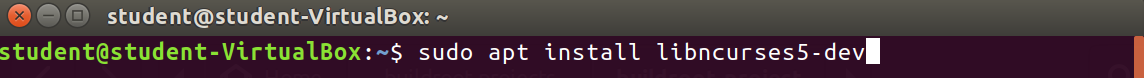
VirtualBox er en gratis programvare utviklet av Oracle Corporation, for å simulere virtuelle maskiner. Windows og diverse Linux-distribusjoner er blant operativsystemene som kan simuleres på disse maskinene.

I dette prosjektet har vi lastet ned og installert VirtualBox, for så å installere versjon 16.04.3 LTS av Ubuntu på en virtuell maskin. Vi installerte også VirtualBox Extension Pack for at den virtuelle maskinen skulle støtte enheter med USB 2.0 og USB 3.0.

**Buildroot**

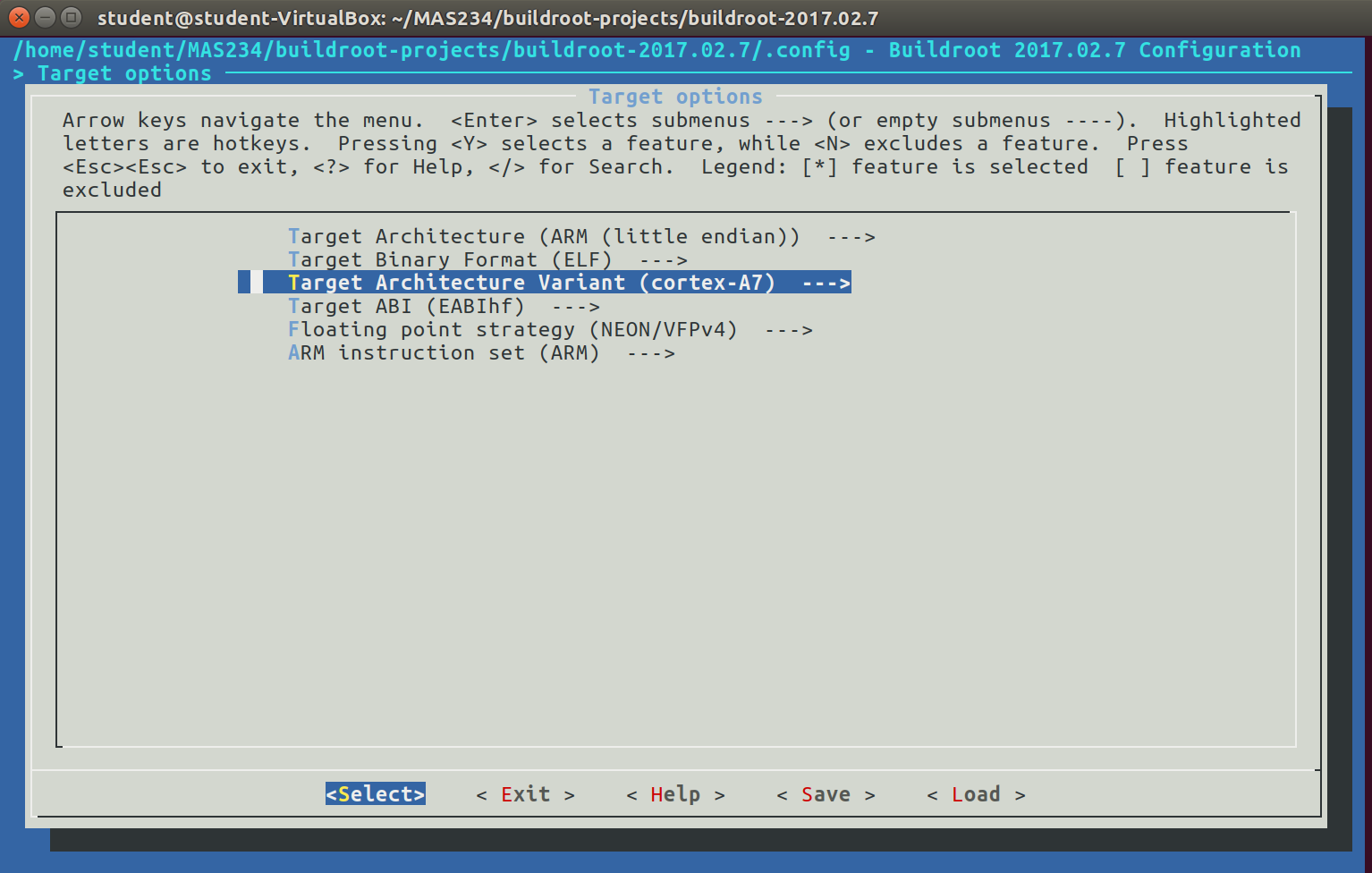
Buildroot er en gratis ”open source”-programvare for å enkelt bygge og konfigurere en komplett Linux-distribusjon for innebygde datasystemer.

Som en del av prosjektet skulle vi konfigurere en egen Linux-distribusjon som skulle fungere med en Raspberry Pi 3 og et Skpang CAN-kort

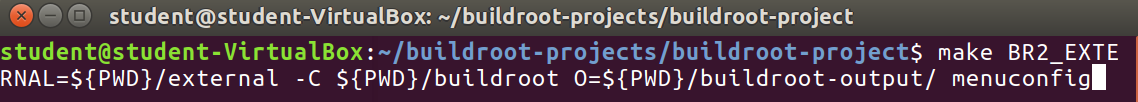
Vi lastet ned versjon 2017.02.07 av Buildroot på vår virtuelle Ubuntu-maskin. Via terminalvinduet fra buildrootprosjektet, prøvde vi å åpne konfigurasjonsmenyen med kommandoen ”make menuconfig”. Vi fikk beskjed om at ncurses-biblioteket manglet og måtte installeres. Dette installerte vi fra pakkebrønnen:

Når ncurses-biblioteket var installert, kontrollerte vi at vi fikk tilgang til konfigurasjonsmenyen til buildroot. Videre kunne vi kjøre kommandoen ”make list-defconfigs” i terminalen. Dette gav oss en liste med alle eksisterende konfigurasjoner for diverse maskinvare. Siden vi bruker en Raspberry Pi 3 i dette prosjektet, kjørte vi kommandoen for å konfigurere denne.

Etter dette åpnet vi konfigurasjonsmenyen igjen, og kontrollerte hvilke ”target options” vi hadde. Her kunne vi blant annet lese at prosessoren vår var en Cortex-A7 basert på ARM-arkitektur.

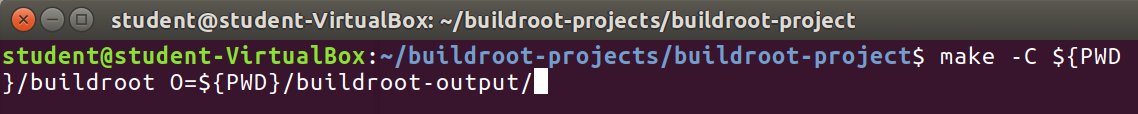


**Installasjon av delvis ferdig konfigurert oppsett**

Faglærer la ut en kopi av et delvis ferdig konfigurert oppsett til Raspberry Pi 3. Denne lastet vi ned fra Fronter og fulgte videre veiledning i oppgaven for installasjon. Vi åpnet buildroot konfigurasjonsmenyen fra prosjektmappa for å sjekke oppsettet.

For å installere can\_pingpong-pakken, kopierte vi mas234-konfigurasjonsfilen inn i buildroot-output-mappa. Vi navnga filen til ”.config”.

Vi kontrollerte nå at can\_pingpong var valgt i konfigurasjonsmenyen, og startet kompileringen med kommandoen:

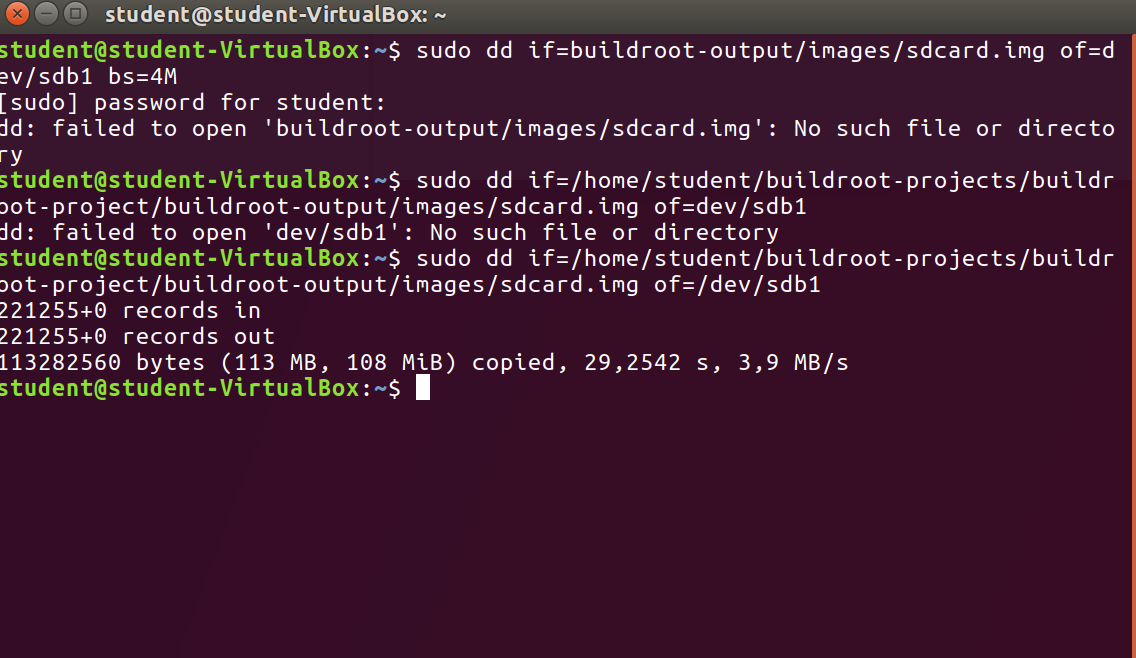


**SD-kort**

Kompileringen genererte et ferdig SD-kort image. Dette skulle vi skrive til et microSD-kort ved å benytte linux-programmet ”dd”.

dd står for ”data duplicator”, og er et verktøy med primærfunksjon å konvertere og kopiere filer. Dette gjøres direkte i terminalen med kommando-linjer.

Vi koblet til microSD-kortet med en USB-adapter, da den interne SD-kort leseren til datamaskinen ikke fungerte med Ubuntu. For å sikre oss mot at vi skrev SD-kort imaget til vår primærharddisk, kjørte vi kommandoen ”fdisk –l” i terminalen. Denne kommandoen listet opp tilgjengelige disker før og etter vi koblet til SD-kortet. Vi så da at minnekortet var navngitt /dev/sdb.

Ved å bruke ”dd”, skrev vi image-filen vår til SD-kortet. I figur ”????” kan det ses at vi skrev filen til /dev/sdb1. Dette er en partisjon på SD-kortet, og forårsaket feil da vi startet opp Raspberry Pien. Vi endret dette og skrev image-filen til hele SD-kortet da vi oppdaget feilen.

**Koble opp hardware**

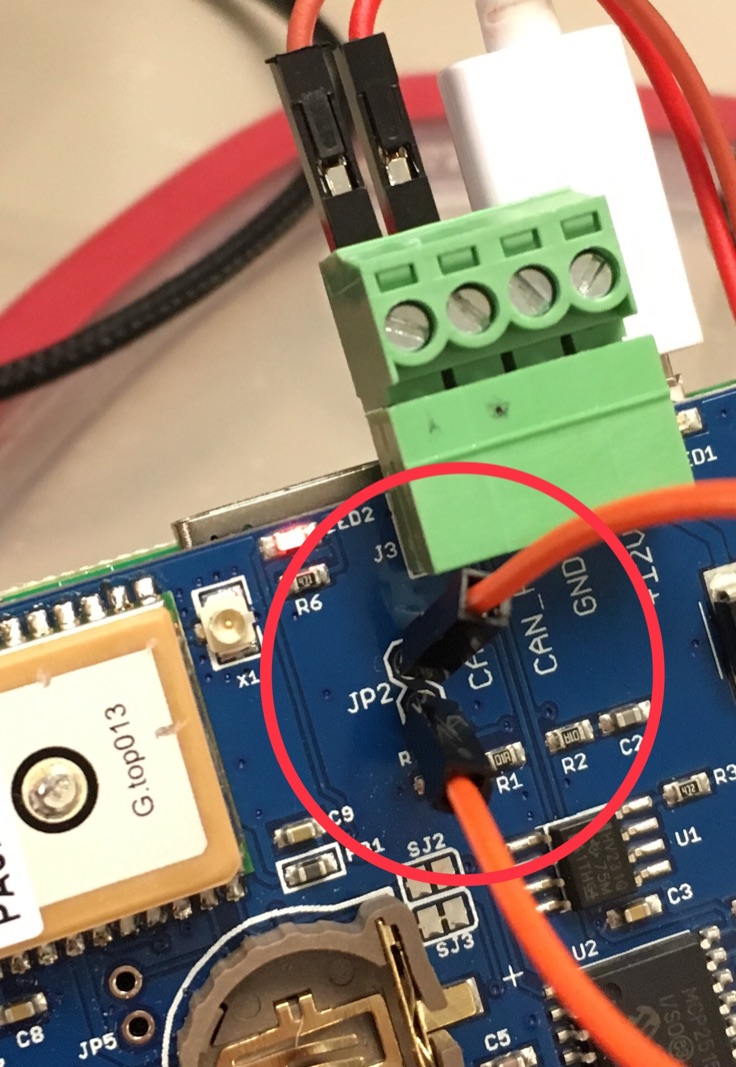
Raspberry Pi er en serie av små datamaskiner bygget på et enkelt kretskort. De er utviklet av Raspberry Pi Foundation, hovedsakelig for å promotere undervisning i grunnleggende databehandling på skoler. Produktet har blitt svært populært, og er nå også brukt innenfor robot- og mekatronikk-teknologi. Raspberry Pi 3 modell B bruker en 1.2 GHz 64-bit quadcore prosessor, har 1 GB ram og blant annet USB- og ethernettilkoblinger, trådløst nettverkskort og bluetooth.

PiCAN er et tilleggskort til Raspberryen som gir oss GPS, Gyro, Akselerometer og CAN-bus kommunikasjon.

I dette prosjektet har vi brukt en Rasperry Pi 3 modell B, samt et PiCan-tilleggskort koblet over det 40 pins store koblingspunktet til Pien.

I dokumentasjonen til Raspberry Pien fant vi spesifikasjoner på hvordan strømforsyning som kreves. Pien må bli forsynt med 5.1V over microUSB. Nøyaktig hvor mye strøm den trekker, varierer ut i fra hva man kobler til av utstyr. Typisk bruker modell B mellom 700-1000mA i seg selv. Fordi vi ikke hadde noe vanlig strømforsyning tilgjengelig, brukte vi en USB port fra en Macbook Pro. Apple oppgir alle USB3.0 til å kunne forsyne utstyr med opptil 900mA ved 5V.

Når Raspberry Pien var koblet opp med 5V strømforsyning og gav indikasjon på suksessfull oppstart med microSD-kortet, skulle vi koble til PiCAN-kortet. Vi koblet fra strømforsyningen, og monterte kortet forsiktig på det 40 pins svære koblingspunktet på Pien.



Vi koblet oss på CAN\_H og CAN\_L på PiCan-kortet gjennom PCAN-USB adapteren. For å kommunisere over CANbus, er man avhengig av å terminere nettverket med en motstand. I brukermanualen til PiCan-kortet står det beskrevet hvordan man kan benytte en intern 120 Ohms motstand til termineringen, ved å koble en ledning mellom punktene på JP2.

**Konfigurasjon av ethernettilkobling**

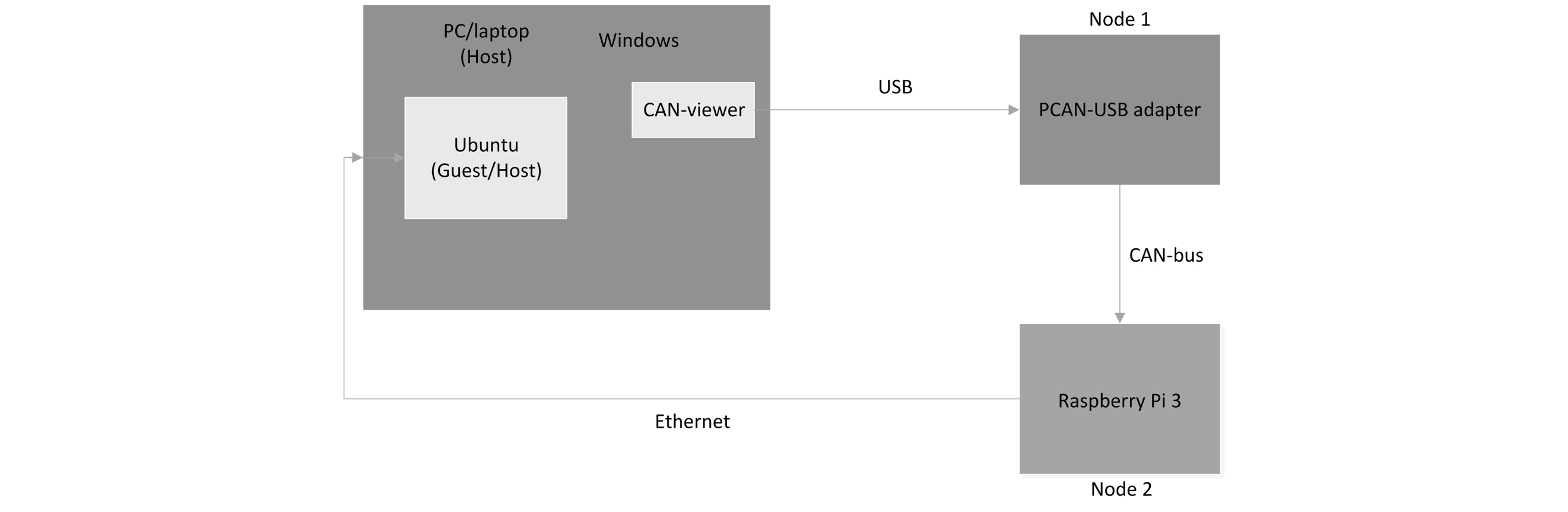
Vi skulle kommunisere gjennom Raspberry Pi med to metoder:

1. En virtuell Ubuntu-maskin over ethernetkabel.
2. PcanView i Windows over et CAN-nettverk, via en USB-adapter.

Figur ???? viser hvordan vi koblet opp systemet vårt i denne delen av prosjektet.

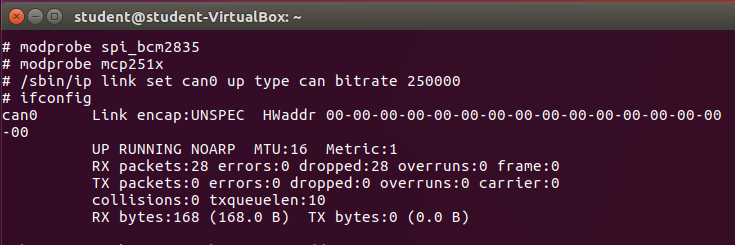
For å kommunisere med Raspberryen over ethernet, måtte vi gjøre noen endringer i nettverksinnstillingene til VirtualBox. Vi lagde en ny ”bridged adapter” og noterte oss MAC-adressen. Under nettverksforbindelsene i Ubuntu, fant vi riktig ethernettilkobling ved å sammenligne adressen mot MAC-adressen til adapteren. IP-adressen til maskinen vår satt vi til å være 192.168.234.1 under IPv4. Vi vet fra før at IP-adressen til Pien er satt til 192.168.234.234.

SSH er en protokoll for sikker tilgang til en datamaskin fra en annen. Dette lar en blant annet overføre filer og kjøre kommandolinjer over internett. Med kommandoen ”ssh [root@192.168.234.234](mailto:root@192.168.234.234)”, fikk vi tilgang til Raspberry Pi på root-nivå. I Linuxdistribusjoner vil det si at vi har tilgang til hele operativsystemet og kan gjøre alle slags endringer. Dette kalles også for ”superbruker”.

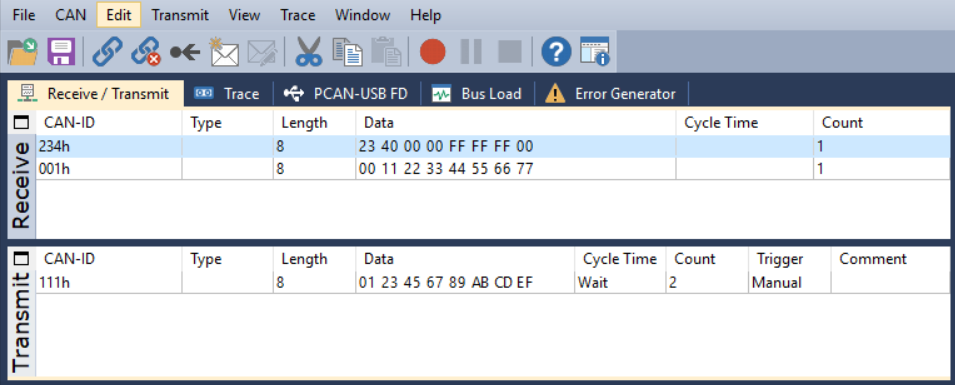
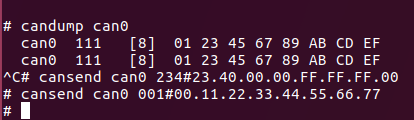
+

Elrjngerjgn

**Kommunikasjon over CAN-bus**

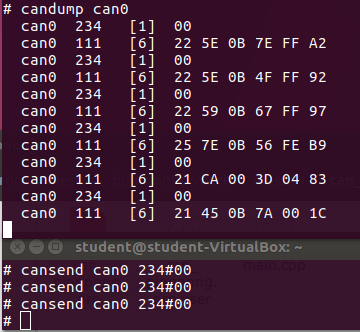
Ved å kjøre kommandoer for å laste inn driverne for SPI og CAN-transceiveren på PiCAN- kortet vårt, samt en kommando for å konfigurere CAN-grensesnittet, gjorde vi det mulig å kommunisere over CAN-bus. For å kontrollere at dette ble utført riktig, kjørte vi ”ifconfig” i terminalen for å se hvilke nettverksgrensesnitt som eksisterte. Can0 kom opp som aktiv. (Se figur)

Kommandolinjen ”candump can0” printer alle CAN-meldinger som sendes på nettverket. For å teste kommunikasjonen mellom PCAN-adapteren og Raspberryen, forsøkte vi å sende meldinger fra PcanView til terminalen i Raspberry. Figur (???) og figur (???) viser to meldinger med ID 0x111 og likt innhold.



For å ta i bruk CAN-nettverket til et praktisk eksempel, ønsket vi å bruke Raspberryen til å forespørre informasjon fra Teensyen og den tilkoblede IMUen. Når vi sendte ut en CAN-bus melding med ID 0x234 fra terminalen til Raspberryen, mottok og behandlet Teensyen denne. Teensyen vil så lese av gyrodataen fra IMUen og sende denne dataen i en CAN-melding med ID 0x111.

Figur ??? viser mottatt CAN-melding med ID 0x111 og 6 bytes, hvorav de to første representerer gyroverdi for X-retning, de to neste for Y-retning, og de to siste for Z-retning.



Kilder:

1. Buildroot: <https://en.wikipedia.org/wiki/Buildroot>
2. Cortex-A7: <https://en.wikipedia.org/wiki/ARM_Cortex-A7>

Kilder:

1. Extension Pack: <https://www.virtualbox.org/wiki/Downloads>
2. Ubuntu nedlastning: <https://www.ubuntu.com/download/desktop>
3. VirtualBox nedlastning: <https://www.virtualbox.org>

Kilder:

1. User guide med info om termineringen: <http://skpang.co.uk/catalog/images/raspberrypi/pican/PICANGPSACC_V1.pdf>
2. Info om apple angående maks forsyningsstrøm fra USB3:

<https://support.apple.com/en-us/HT204377>

1. Raspberry Pi dokumentasjon: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/>
2. Raspberry Pi generell info:

<https://en.wikipedia.org/wiki/Raspberry_Pi>

1. Forklaring på hva SSH er:

<https://help.ubuntu.com/community/SSH>

1. Forklaring på hva root vil si:

<http://www.datamaskin.biz/Systems/basic-computer-skills/200343.html#.WgtnxLaDrOQ>

Diskusjon:

1. Ikke lurt å koble seg på root-nivå over SSH, spesielt når man har et svakt passord (mas234)