**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.

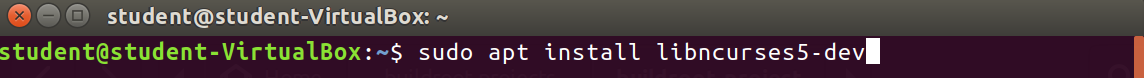
VirtualBox er en gratis programvare utviklet av Oracle Corporation, for å simulere virtuelle maskiner. Windows og diverse Linux-distribusjoner er blant operativsystemene som kan simuleres på disse maskinene.

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**

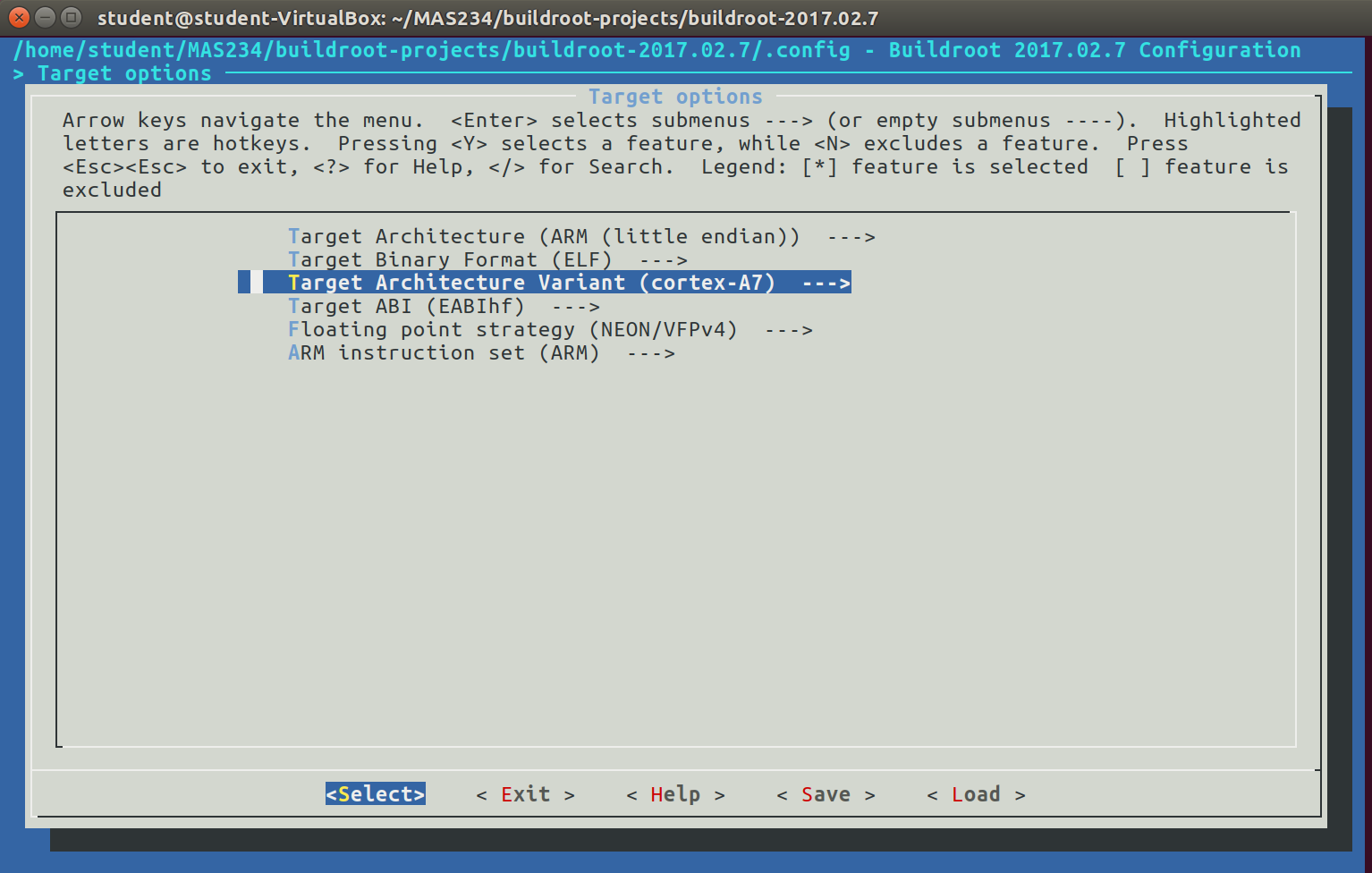
Buildroot er en gratis ”open source”-programvare for å enkelt bygge og konfigurere en komplett Linux-distribusjon for innebygde datasystemer.

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

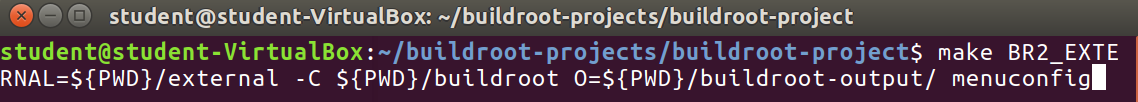
Vi lastet ned versjon 2017.02.07 av Buildroot på vår virtuelle Ubuntu-maskin. Via terminalvinduet fra buildrootprosjektet, prøvde vi å åpne konfigurasjonsmenyen med kommandoen ”make menuconfig”. Vi fikk beskjed om at ncurses-biblioteket manglet og måtte installeres. Dette installerte vi fra pakkebrønnen:

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.

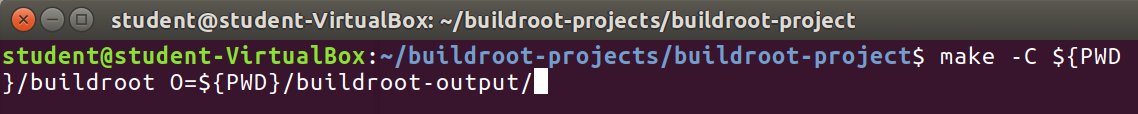


**Installasjon av delvis ferdig konfigurert oppsett**

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:

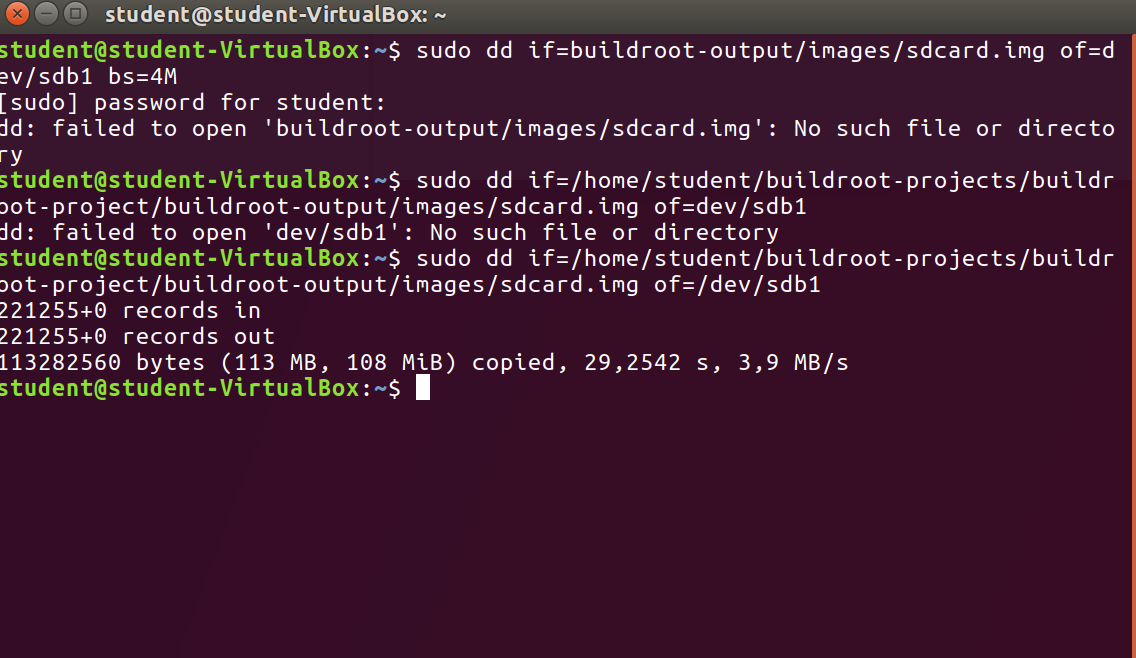


**SD-kort**

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**

Skrive litt om raspberry Pi og tilleggskortet

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 model B, som er den vi har, 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 opp Pi-hat kortet på Raspberryen.

Oppgave 2)

Raspberry Pi 3 blir forsynt av en 5,1V mikroUSB. Hvor mye det totale strømtrekket vårt er, vil variere ut i fra hva som er koblet til. Typisk forbruk for modell B, som er den vi har, er mellom 700-1000mA. Maximum står oppgitt til å være 1A. I tillegg kommer altså alt vi kobler til????

Den offisielle og anbefalte strømforsyningen fra og for Raspberry Pi gir ut stabile 2A.

Hvor mye har vi i USBen vår?

Terminering: Det ligger en intern termineringsmotstand på 120 Ohm på JP2. Ved å koble en ledning mellom disse to punktene, terminerer vi BUS-en. Uten terminering, klarte vi ikke å sende/motta CAN-meldinger.

Skrive om tilkobling av Raspberry med ethernet:

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/>