

# Prosjekt del 4

Raspberry Pi 3 / Buildroot og CAN-bus.

## Om prosjekt del 4

Nødvendig SW for siste del av prosjektet er Ubuntu Linux (f.eks. installert på en virtuell maskin) og QT Creator (kjørende på Ubuntu Linux).

## Dag 1 (og 2)

### Oppgave 0 - Installere Ubuntu Linux på Virtualbox

- Ubuntu-installasjonen fra første øving kan gjenbrukes.
- Maskinen som kjører Ubuntu-linux kalles "host" under lab-en. Merk at host-maskinen bør ha tilgang til en ethernet-port (hvis Virtualbox kjøres på en maskin med ethernet-tilgang, så kan vi benytte denne porten).
- Raspberry pi kalles "target".

### Oppgave 1 - Laste ned og konfigurere buildroot

Buildroot er et verktøy som benyttes for å konfigurere og bygge Linux-distribusjoner for innebygde datasystemer.

På lab-en skal vi benytte buildroot til å konfigurere en egen Linux-distribusjon for MAS234. Linux-distribusjonen skal fungere med CAN-bus på kombinasjonen Raspberry Pi 3 og Skpang CAN-kort.

Vi skal også legge inn en ekstra programvarepakke i buildroot-menyen, slik at vi får med vår egen programvare når distribusjonen genereres.

Buildroot har en stor samling programvarepakker og mengder av konfigurasjonsmuligheter.

Vi behøver heldigvis ikke benytte alt; og det finnes også ferdige oppsett for en god del maskinvare (slik som Raspberry Pi 3).

1. Last ned buildroot 2017.02.07 på Ubuntu Linux. Pakk ut og legg buildroot-katalogen inni en mappe som heter "buildroot-projects".

<https://buildroot.org/>

2. Åpne et terminalvindu og bruk cd (change directory) til å navigere frem til katalogen som ble pakket ut i forrige punkt. Kjør:

```
make menuconfig
```

Merk forøvrig at det **ikke** skal være nødvendig å kjøre noen av de buildroot-relaterte kommandoene som superbruker (unngå bruk av sudo).

Tips: Sannsynligvis mangler ncurses på maskinen. Dersom buildroot klager på dette, så må ncurses-devel installeres på Ubuntu før dere fortsetter. Pakken heter "libncurses5-dev" i Ubuntu-pakkebrønner.

3. Når `make menuconfig` i forrige steg fungerer; lukk konfiguratoren og kjør `make list-defconfigs`. Denne kommandoen lister opp eksisterende konfigurasjoner ("templates") for diverse maskinvare.

Kjør "make raspberrypi3\_defconfig" og bla litt rundt i menyene. Hvilken prosessor-arkitektur er valgt?

4. Nå KUNNE vi bygd en standard-konfigurasjon for Raspberry Pi 3. Imidlertid krever denne skjerm eller TTL-uart-kabel for terminal. Dette har vi foreløpig ikke, og det er heller ikke alltid tilgjengelig på et realistisk innebygget datasystem.
5. Dere får en kopi av et ferdig oppsett. (Dvs. i det minste halvferdig...). Se Fronter.

Buildroot-prosjektet er konfigurert slik at CAN-bus-relevante pakker er forhåndsvalgt, i tillegg til at det er satt opp en SSH-server.

Prosjektet inneholder en egen mas234-pakke med et qtmake-basert C++-program. Dette kan åpnes i Qt Creator for videreutvikling. Oppgaven i siste trinn blir å lage et program som kommuniserer på CAN-bus basert på dette halvferdige oppsettet.

6. Hent buildroot-project.tar.gz på Fronter og pakk ut. Les README-filen. En kopi av buildroot må pakkes ut eller klones inn i buildroot-mappen.
7. Stå i buildroot-project-mappen (hakkert utenfor buildroot-mappen). Kjør:

```
make BR2_EXTERNAL=${PWD}/external -C ${PWD}/buildroot O=/filbane/til/buildroot-project/buildroot-output/ menuconfig
```

Og sjekk oppsettet. Finnes det en can\_pingpong-pakke

her? Kan denne velges? Forsøk å legge inn nødvendige pakker for å oppfylle avhengighetene til can\_pingpong.

8. mas234-konfigurasjonsfilen kopieres inn i buildroot-output-folderen som .config. Merk at filer med navn som begynner på "." er skjulte filer. For å vise disse, bruk f.eks. `ls -la` fra terminalen. Dette overskriver eventuelle valg som ble gjort i forrige steg. Kjør kommandoen i punkt 7 på nytt.
9. Sjekk at can\_pingpong er valgt. Lukk menuconfig. Start kompilering:

```
make -C /filsti/til/buildroot-prosjekt/
buildroot-output/
```

10. Når buildroot har bygd ferdig (tar gjerne 10-30 minutter avhengig av mask), så skal det ligge et ferdig generert sd-kort-image i folderen buildroot-output/images/.

sdcard.img skal nå skrives til et microSD-kort ved å benytte en microSD-kort-leser og linux-programmet dd. Merk at dd lett skriver over harddisken på maskinen hvis det brukes feil.

Bruk `fdisk -l` for å liste diskene som er montert på maskinen FØR sd-kortet er satt i kortleseren. Merk deg /dev/sda, /dev/sdb etc..

Plugg i minnekortleseren og minnekortet. Kjør `fdisk -l` på nytt. Disken som dukker opp nå er målet for kopieringen. Sjekk gjerne også at størrelsen `fdisk` indikerer er som forventet med tanke på hva slags kort som er satt inn.

Bruk dd til å kopiere, f.eks. når måldisken heter /dev/sde:

```
dd if=buildroot-output/images/sdcard.img of=/dev/sde bs=4M
```

Dette bør gå relativt raskt. Det anbefales å kjøre `umount` på måldisken før kopiering med dd. (F.eks. `umount /media/<brukernavn>/*` for å unmount alle automonterte diskene).

11. Løs ut disken og sett den inn i raspberry PI. Forsøk å boote.
12. <koble til med SSH, se oppgave 3>

13. For å bygge KUN can\_pingpong-pakken og ikke hele buildroot-oppsettet, kjør:

```
make -C /filsti/til/buildroot-prosjekt/  
buildroot-output/ can_pingpong-rebuild
```

14. For å kopiere krysskompilert program til raspberry pi, kjør:

```
scp /filsti/til/buildroot-prosjekt/  
buildroot-output/build/can_pingpong-3.14/  
can_pingpong root@192.168.234.234:/usr/  
bin/can_pingpong
```

## Oppgave 2 - Koble opp hardware

Hver gruppe får utdelt Raspberry Pi 3 og et "pi hat"-kort med IMU, GPS og CAN-bus-funksjonalitet for bruk under prosjektet. I tillegg har vi et microSD-kort pr. gruppe, som fungerer som "harddisk".

Dokumentasjon Raspberry Pi 3: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/README.md>

Dokumentasjon Skpang Can-kort: <http://skpang.co.uk/catalog/pican-with-gps-gyro-accelerometer-canbus-for-raspberry-pi-3-p-1521.html>

- Merk: Pi-hat-kortet SKAL kobles opp m/standoffs slik at GPIO-headeren ikke brekker under bruk.
- Husk å sjekke impedans på CAN-bus-en. Finnes det termineringsmotstand på SKpang-kortet?
- Det er behov for strømforsyning (micro-usb) og nettverkskabel for neste del.

## Oppgave 3 - Sende en CAN-melding fra terminal

1. Sett opp et nettverk med statisk IP på porten hvor Raspberry Pi er koblet til. Velg ip 192.168.234.1 på host-maskinen og nettmaske 255.255.255.0.
2. Raspberry Pi med MAS234-konfigurasjon har IP 192.168.234.234. Koble til denne med SSH fra

terminalen i Ubuntu:

```
ssh root@192.168.234.234
```

MERK: Det er IKKE anbefalt å logge inn som root over SSH. Spesielt ikke med et kort og dårlig passord som vi har her. Men så lenge vi kun kommuniserer med RPI på et eget, lokalt nettverk -- uten internett-tilkobling, så er det OK.

Passord: mas234

3. Kjør `ifconfig` for å sjekke hvilke nettverks-grensesnitt som er aktive. Hvis ikke `can0` er synlig; kjør:

```
modprobe spi_bcm2835
modprobe mcp251x
```

Dette laster kjernemodulene (drivere) for SPI og CAN-tranceiveren på SKPANG-kortet (MCP2515).

Deretter kjøres følgende for å konfigurere can-grensesnittet:

```
/sbin/ip link set can0 up type can bitrate 250000
```

4. Nå skal det være mulig å sende og motta CAN-meldinger på `can0`-grensesnittet. Sjekk gjerne at `can0` eksisterer vha. `ifconfig`.
5. Koble Peak-adapteret på CAN-bus-en. Start candump med:

```
candump can0
```

Candump printer alle can-meldinger som mottas til terminalen. Send noen meldinger fra PCAN-programmet, og verifiser at disse mottas.

Avslutt candump ved å trykke `ctrl+c`

6. Send et par CAN-meldinger fra Raspberry Pi:

```
cansend can0 234#23.40.00.00.FF.FF.FF.00
```

Sjekk at disse kommer frem.

## Oppgave 4 - Fritt spill

1. Koble opp minst to enheter på CAN-nettverket, f.eks. Teensy-en og Raspberry Pi-en.

2. To krav: Minst én node skal lese sensordata og sende disse til en annen node. Noden som mottar sensordata skal gjøre noe med disse, og sende tilbake prosesserte data til sensornoden eller en annen node.
3. Det er mulig å låne f.eks. en RC-servo og styre denne fra Teensy. Benytt i så fall eget (lab)-powersupply på servoen.
4. To eller flere grupper kan koble seg sammen på ett og samme CAN-nettverk.

## Oppgave 5 - Innlevering av utstyr

Gruppene kan benytte lånt utstyr frem til og med siste lab-dag (mandag 11. november). Alt utstyr må være innlevert før rapportens innleveringsfrist.