

nRF52DK and blink

Nordic semiconductor nRF52 Development Kit (nRF52DK)

BLINKY

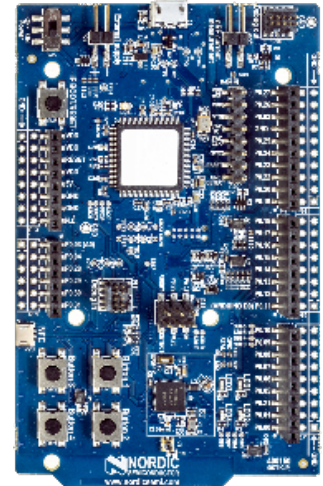
Anta at du har eit nRF52dk i handa og ingenting anna. Vi skal nå hjelpe deg å få det til å blinke.

1. Hardware

1. Du treng ein [Micro USB](#) til å plugge i nRF52.
2. Du treng ein datamaskin for å plugge andre enden av microUSBen i. Denne gjennomgangen antar at du finn ein datamaskin som køyrar windows eller GNU/Linux.
3. Vi anbefaler at du har både datamus og tastatur til datamaskinen din, da dette vil gjere resten av gjennomgangen enklare å følge.

2. Software

1. Du må nå installere følgende programmer frå internett. Vi legg med linker, men viss det slutter å fungere, skal du kunne finne nettsidene ved hjelp av ein søkemotor på internett.
 - a. [Segger embedded studio](#)
 - b. [Nordic Command Line Tools](#) (AUR)
 - c. [nRF5 SDK](#)
 - i. Dette er ei .zip mappe, unzip der du føler det er praktisk
 1. Windows: Du bør lagre den så langt nede i mappehierarchyet som mogleg, fordi windows [slit med lange filstinamn](#).
 2. GNU/Linux: Sørg for at brukeren din har rettigheter til denne mappa.



Source: [Nordicsemi.com](https://www.nordicsemi.com)

3. Nordic Command Line Tool

- Windows
 - Opne Command Prompt som administrator.
 - GNU/Linux
 - Bruk terminalen, og viss kommandoane ikkje fungerer, vurder å bruke sudo.
1. Køyr kommandoen "nrfjprog"
 2. Denne kommandoen velger om du skal ha eit [SoftDevice](#) på nRF52, og i såfall kva for eit du vil ha.
 3. For å fjerne alle SoftDevices på nRF52, bruk kommandoen "nrfjprog --eraseall" eller "nrfjprog -e".
 4. For å legge til eit [SoftDevice](#) (dette steget kan hoppast over til du veit kva SoftDevice du treng):
 - a. Naviger ved bruk av cd til nRF5_SDK_[versjon]/components/softdevice. Ex: "cd nRF5_SDK_[versjon]/components/softdevice"
 - b. Her velger du kva [SoftDevice](#) du vil bruke, dei har namn som f.eks. s112. Du vil vite kva SoftDevice du treng når du treng eit (Dette blir også nevnt seinare her).
 - c. Naviger vidare til den mappa du vil, og til "hex" mappa i den. Her finn du ei fil ved endelse .hex. F.eks. "s112_nrf52_6.1.1_softdevice.hex"
 - d. Køyr kommandoen "nrfjprog --program [softdevice].hex". Ex: "nrfjprog --program s112_nrf52_6.1.1_softdevice.hex"
 - i. NB hugs å bruke heile filstien til SoftDevice, med mindre du er i mappa.

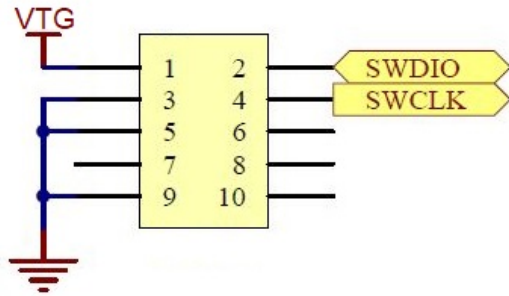
4. Segger Embedded Studio

1. Opne Segger Embedded Studio. Dette fungerer forskjellig i Windows og på Linux, følg tutorials på internett for korleis.
2. Først skal vi opne eit prosjekt: Øverst er det ein meny. Velg følgende:
 - a. File
 - b. Open Solution...
 - c. Naviger til "..\nRF5_SDK_[versjon]/examples/peripheral/blinkypca100xx/blank/ses/blinkypca100xx.emProject"
 - i. pca100xx: Sjekk kva som står på den kvite lappen på den store chippen ca midt på nRF52dk. F.eks. PCA10040. Bytt ut pca100xx med det du finn.
 - ii. blank: Dette er din [SoftDevice](#). "blank" betyr at det ikkje er ein [SoftDevice](#) på nRF52 enda, noko du også kan oppnå ved å bruke "nrfjprog --eraseall". Forskjellige eksemplar har forskjellige [SoftDevice](#), og du må ha bruke "nrfjprog --program" (sjå <https://www.nordicsemi.com/Software-and-tools/Development-Tools/nRF-Command-Line-Tools/Download> over) for å programmere ein [SoftDevice](#) som samsvarer med den du bruker i prosjektmappa di for at koden din skal fungere.
 - iii. ses står for Segger Embedded Studio. Viss du vil bruke andre IDEer, må du velge ei anna mappe her.
 - d. Klikk "Open"
3. Deretter skal vi bygge prosjekter. I menyen øverst, velg:

- a. Build
 - b. Build Blinky_pca100xx
 - c. Trykk "Continue" i lisensviduet som kjem.
 - d. Nederst i vinduet til Segger, er det eit Output-vindu. Her Ser du om koden bygga suksessfullt.
4. Nå skal vi koble til nRF52dk og laste koden over på det. I menyen øverst, velg:
 - a. Target
 - b. Connect J-Link
 - c. Target
 - d. Download blinky_pca100xx
5. Gratulerer, nå skal din nRF52dk blinke.
 - a. Viss dette ikkje er tilfellet, sjekk at du har fulgt stega over nye, gjerne prøv gjere dei igjen. Kom gjerne innom Omega Verksted for å spørre om hjelp, sammen skal vi få det til.

Debugging of external target

The devkit can also be used to program and debug a nRF52 device on another board. This is done via the *Debug out* connector. The following connections are required **on the target board**, mapped to the Debug out connector on the DK as follows:



When the external board is powered, the interface MCU will detect the supply voltage of the board and program/debug the target chip on the external board instead of the on-board nRF52832.

For more info see the [Nordic Infocenter page](#).