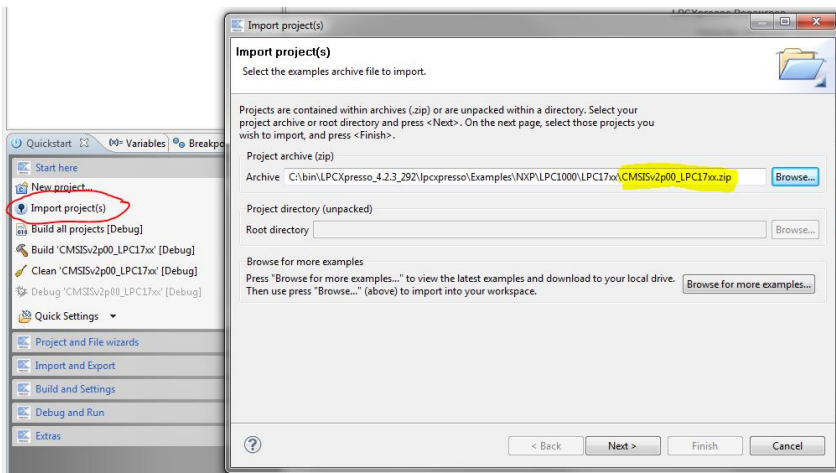


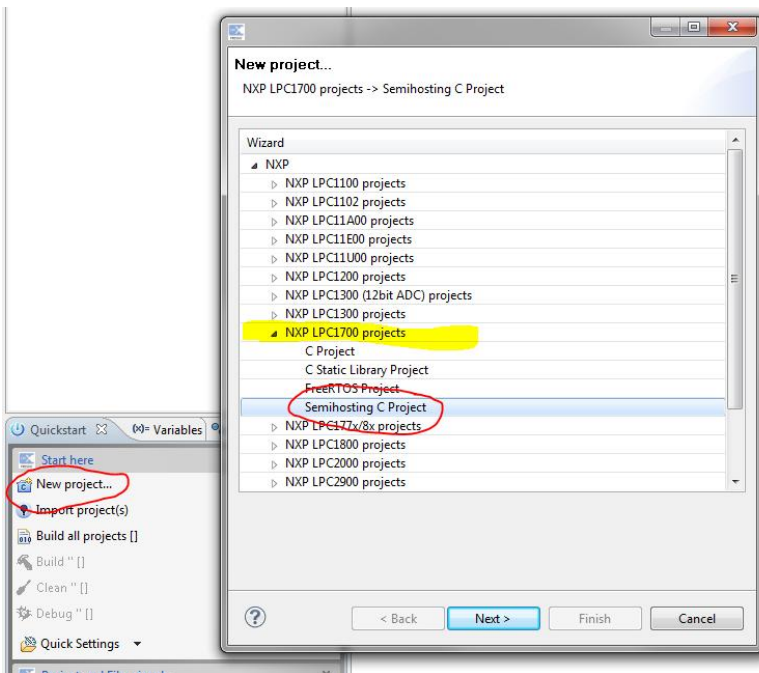
# Projektin luonti LPCXpressolla

Ensimmäiseksi työtilaan (workspace) pitää liittää käytössä olevalle prosessorille sopiva CMSIS-kirjasto. LPCXpresson projektinluontivelhot olettavat, että projektit käyttävät CMSIS-kirjastoa. Kirjaston puuttuminen estää projektin luomisen ja ohjelman kääntämisen, koska kirjasto sisältää mm. kellotaajuuden määrittelyjen vaihelukkojen asetukset sekä C-kielisen ohjelman ennen mainin käynnistämistä tehtävät alustukset.



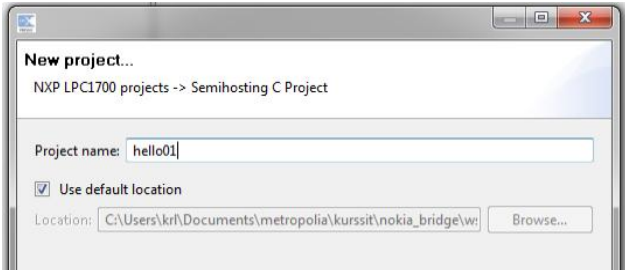
CMSIS-kirjasto liitetään valitsemalla *Import project* ja valitsemalla kirjaston sisältävä zip-tiedosto. Käyttämämme kirjasto löytyy lpcxpresson asennushakemiston alta *Examples\NXP\LPC1000\LPC17xx-*hakemistosta.

Kirjaston liittäminen jälkeen projektin luominen tapahtuu luontivelhojen avulla. Valitse *New project* ja valitse avautuvasta ikkunasta sopiva luontivelho.

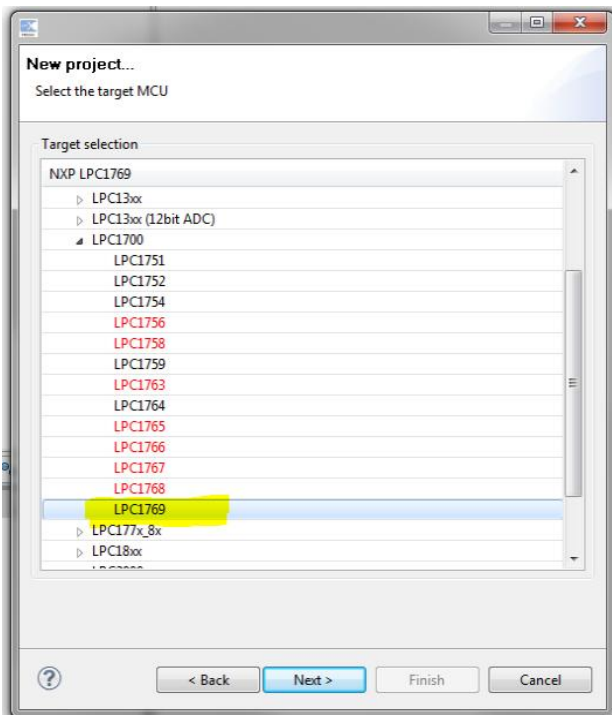


Semihosting-projektissa syöttö ja tulostus (printf/scanf jne.) tapahtuu debuggeriliitännän kautta isäntäkoneessa. Syöttö ja tulostuu debuggeerin kautta on hidasta, joten mitään aikakriittistä ei voi tehdä syötön ja tulostuksen yhteydessä. Asetuksen voi helposti vaihtaa myöhemmin, jos sovellus ei tarvitse standardi-kirjaston syöttöä ja tulostusta.

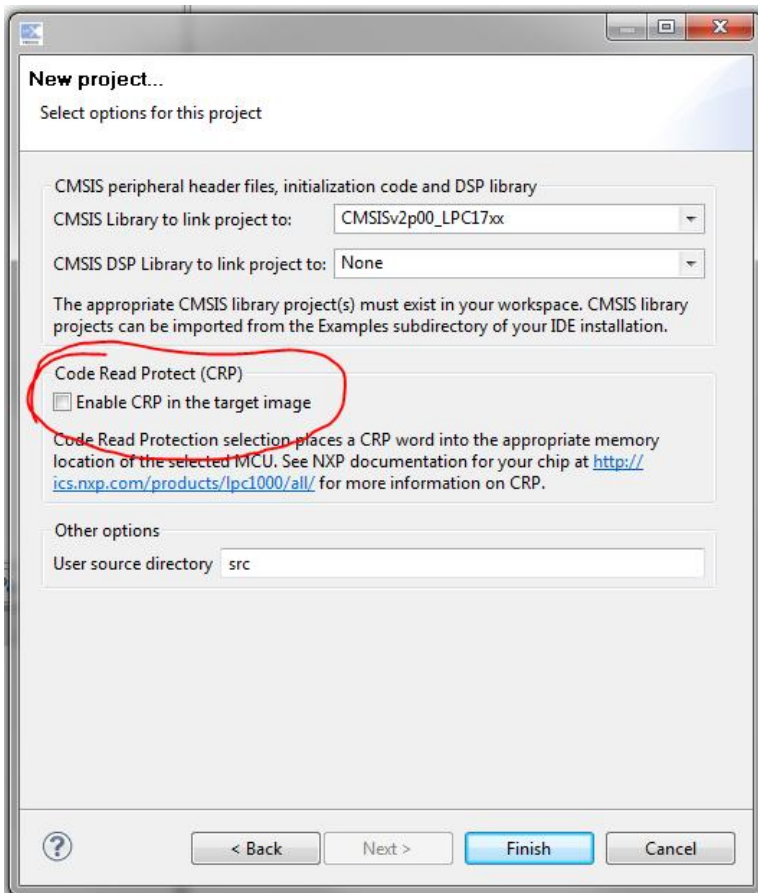
Projektille pitää tietenkin antaa myös nimi...



... ja valita käytettävä prosessori. Käyttämässämme kortissa on LPC1769. Prosessorin nimi näkyy listassa punaisella, koska sen flash-muisti on suurempi, kuin mitä LPCXpresson debuggeri tukee. LPCXpressossa on 128kB rajoitus debugattavan ohjelman koolle, mutta sen saavuttamiseksi pitää kirjoittaa jo melkoisen iso ohjelmisto.





**CRP pitää olla pois päältä.** CRP:n avulla piiri voidaan suojata siten että flash-muistin sisältöä ei voi muuttaa tai debuggerin käyttö voidaan estää. Osa CRP:n asetuksista on sellaisia, joita ei pysty käyttöönoton jälkeen peruuttamaan, joten CRP:llä laitteen voi saada täysin käyttökelvottomaksi.



## FreeRTOS projektin luominen

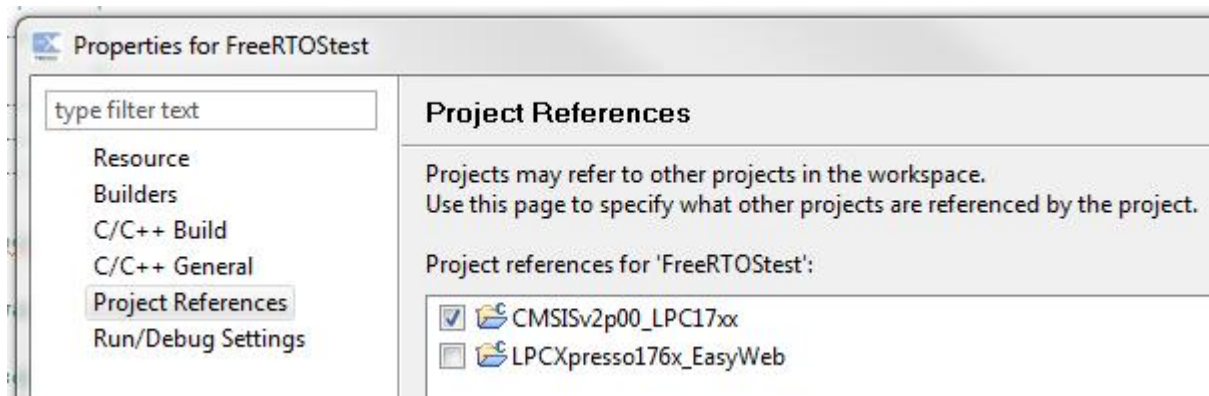
Periaatteessa FreeRTOS-projektin luominen on hyvin helppoa, koska eclipsen luontivelhosta voi suoraan valita FreeRTOS-projektin ja noudattaa ylläolevia ohjeita. FreeRTOS-projekti ei kuitenkaan automaattisesti käytä CMSIS:iä, joten laitteiston ohjauksen helpottamiseksi CMSIS kannattaa liittää mukaan.

Liittäminen on suoraviivaista, mutta vaatii jonkun verran hiiriakrobatiaa.

Ennen kuin CMSIS-kirjastoon voidaan luoda viittauksia, niin kirjasto pitää olla olemassa. Ensimmäiseksi pitää kääntää (build) CMSIS-kirjasto sekä *Debug* että *Release* konfiguraatioilla. Konfiguraation valinta tapahtuu  -napilla (paina nappia ja valitse Debug/Release). Valinnan jälkeen paina  -nappia (konfiguraation valintanapin vieressä).

## Viittaukset

Ensimmäiseksi kerrotaan että projekti viittaa toisiin projekteihin, jotta käännöksen yhteydessä myös riippuvat projektit käännetään mikäli niissä on tapahtunut muutoksia. Napsauta FreeRTOS-projektia hiiren oikealla napilla, valitse Properties. Ikkunasta valitse Project References ja ruksaa CMSIS-projekti mukaan.



Viittauksen lisäämisen jälkeen tarvitaan muutama polku- ja symbolimäärittely.

## Symbolit

Properties → C/C++ General → Path and Symbols → Symbols

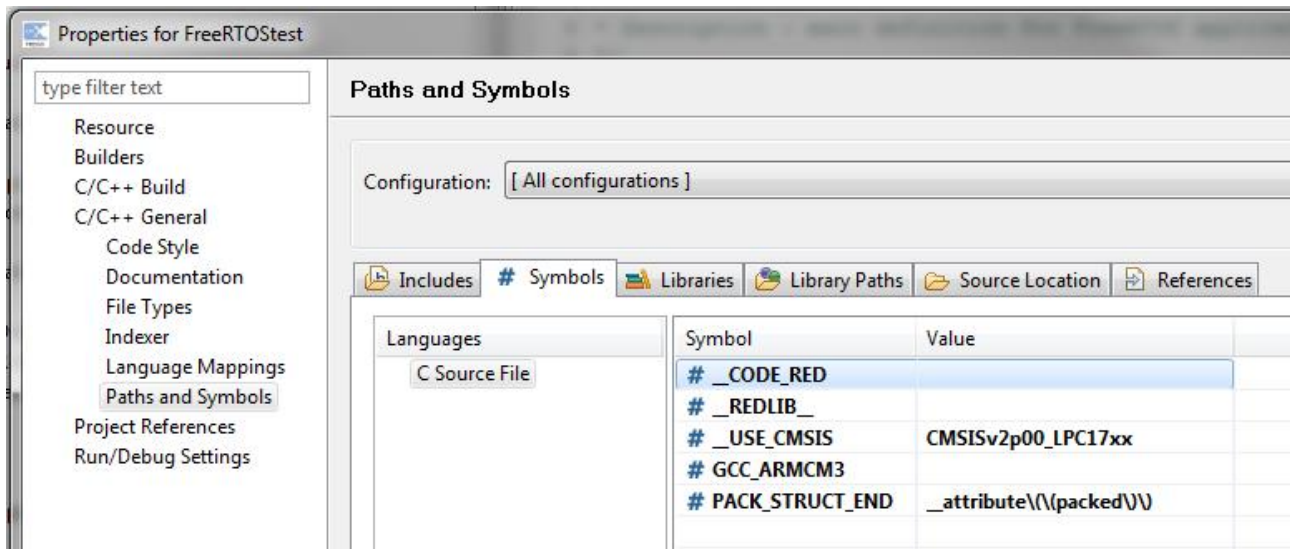
Configurations: [All configurations]

Sitten paina: ADD

Symbol: `__USE_CMSIS` (luo tämän niminen symboli)

Value: `CMSISv2p00_LPC17xx` (Tarkista oikea määrittely CMSIS-kirjaston nimestä. LPCXpresson 4 ja 5 versioissa on eri versio CMSIS-kirjastosta).

Muista valita "Add to all configurations", jotta määrittely menee sekä *Debug* että *Release* konfiguraatioihin.



## Polut (includes)

Properties → C/C++ General → Path and Symbols → Includes

Languages: Gnu C

Configurations: [All configurations]

Paina: ADD

Paina workspace-nappia ja valitse /CMSISv2p00\_LPC17xx/inc

## Kirjasto

Properties → C/C++ General → Path and Symbols → Libraries

Configurations: [All configurations]

Paina: ADD

**Kirjoita kirjaston nimi** (sama kuin CMSIS-projektin nimi) valintaruutuun ja paina OK. Älä kirjoita polkua tähän!

File: CMSISv2p00\_LPC17xx

## Kirjaston hakupolut

Kirjaston hakupolut on tehtävä kummallekin konfiguraatiolle erikseen.

Properties → C/C++ General → Path and Symbols → Library paths → Add

Releaselle: CMSISv2p00\_LPC17xx/Release (workspace path)

Debugille: CMSISv2p00\_LPC17xx/Debug (workspace path)

Lisää ohjelman alkuun seuraava include:

```
#include "LPC17xx.h"
```

Ohjeita LPCXpresson tukisivuilla: <http://support.code-red-tech.com/CodeRedWiki/LibraryProjects>