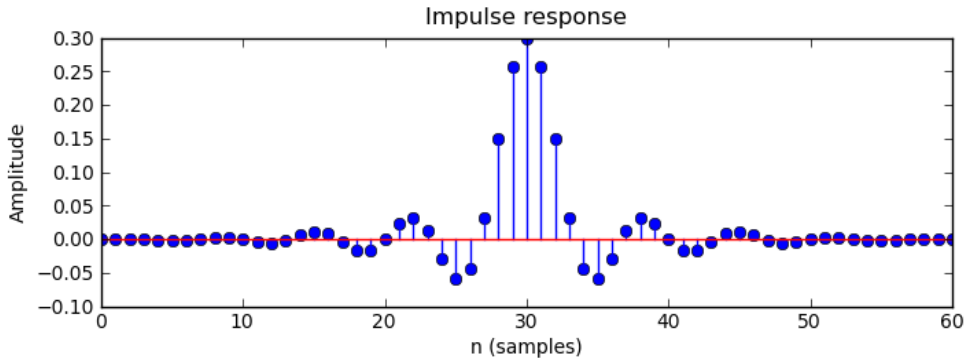
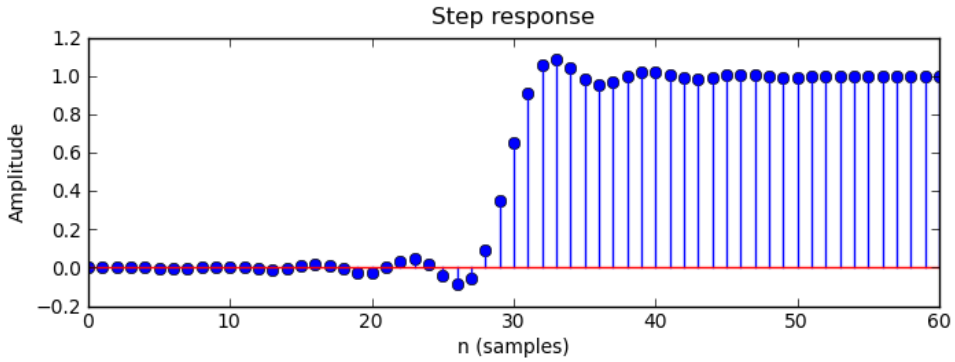
ESE DSPESEL

Digital Signal Processing practical work assignments

HAN Electrical Engineering/Embedded Systems

E.J Boks MSEE MA





Contents

1 General information

All practical assignments, software and accompanying information can be found as a Subversion repository at:

https://ese.han.nl/svn/dsbpracticum/trunk/2022

Students who started the practical work in the previous school year can load their version at:
https://ese.han.nl/opleiding/dsbpracticum/tags/2021

E/ESE students can log on using these qualifiers:

login : *dsbstudent*

password : *SignaalbewerkinG*

In addition, the general ESE log on for all ESE SVN practical work repositories can be used here :

login : *esestudent*

password : *EmbeddedSystemsEngineering!*

The practical work assignement can also be seen in your web browser. Go to DSB practicum to view everything with webSVN.

2 Prerequisites

2.1.1 Microsoft OneDrive

To participate successfully in the practicum, it is recommended to use a Microsoft OneDrive account. This can be requested free of charge (for 1 GBytes) at Microsoft OneDrive login.

After authentication in OneDrive, documents will be synchronized quickly:

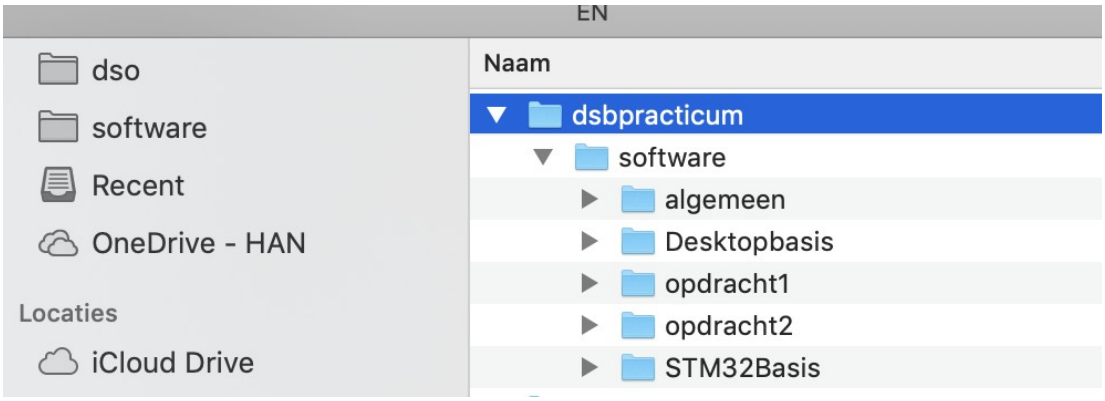


Image 1: The OneDrive Icon in the Finder (Mac OS X)

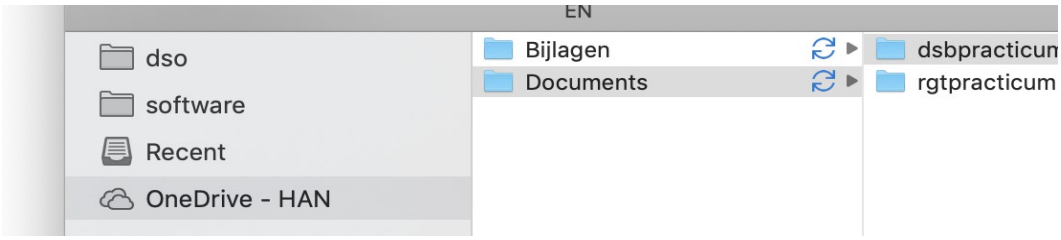


Image 2: Synchronization of documents is indicated by the blue arrows

2.1.2 Working on the assignments at school

The practical work will be done on the student laptop computer, where the student work will be kept in the OneDrive directory, preferably.

2.2 The development board

Finally, you will also need an DSB practical work board. Below the board is depicted that will be used:

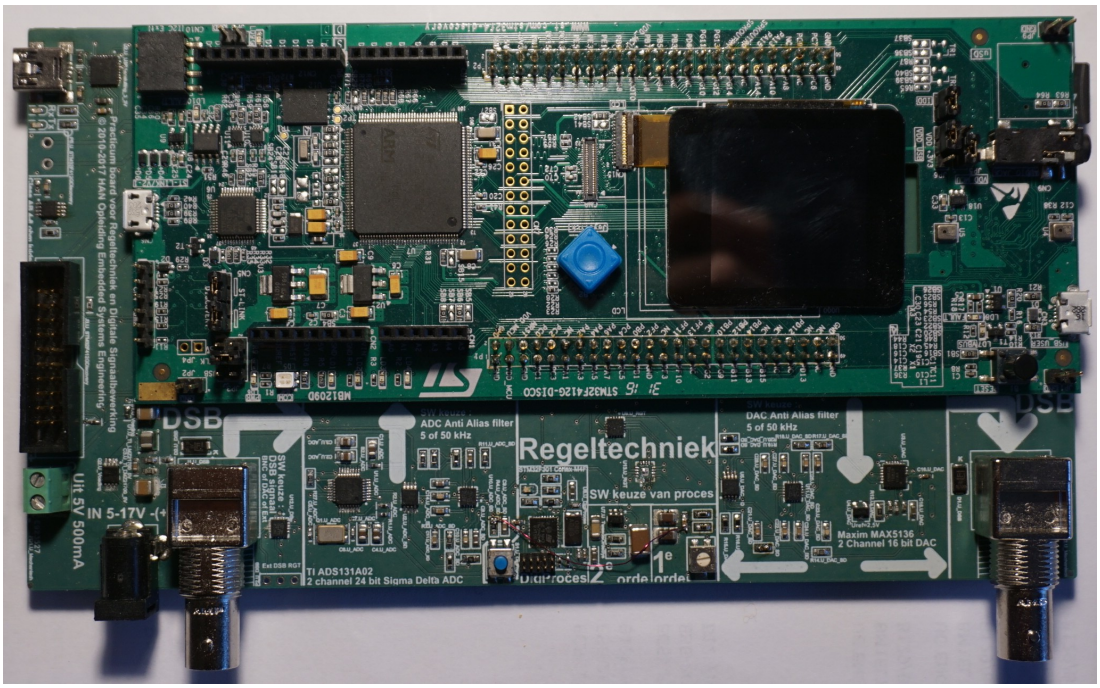


Image 3: The RGT + DSB Board, recognizable by the STM32F412 Discovery module:

The boards are present in room H036.
In addition, ARLE (Jelle Ellemans) manages extra boards that can be lent by students for work outside regular school hours.

3 Software

3.1 Software packages used in the practical work

During this course, the following software packages will be used. Options are presented for Windows and OS X. Linux has slightly different needs, please contact the teacher for these.

- Microsoft OneDrive, for synchronization of students elaboration (All).
- Microsoft Visual Studio as development environment (MS Windows).
- The XCode development environment (Mac OS X).
- Kitware CMake for generating the Visual Studio solutions (MS Windows).
- The wxWidgets toolkit (All).
- Jetbrains CLion for building software on Apple Mac OS X or Linux, and for performing embedded software development (All).
- ARM GNU EABI toolchain for compiling the embedded software (All).
- Tortoise SVN. For version management and downloading the practical items (MS Windows).
- The SVN port obtained through Macports (Mac OS X).
- Segger J-Link en Ozone . For programming and debugging the software in the embedded target (All).

It is imperative that students install these tool prior to or during the first week of the course.

Two documents exist for helping out students to install the software. These documents are enclosed in the bundle of documents that is delivered through SVN. They can also be found on the ESE DSPESE website website. Access to the download goes as follows:

- the download password for the embedded toolchain documentation is *STM32*
- het download password for the desktop IDE documentation is *wxWidgetsIsFun*

3.2 First time initialization

The student must download the practical work documents and software from the WWW or SVN Server into his OneDrive account.

When using ZIP from the WWW server : download and unpack the zip in the OneDrive documents directory.

When using the SVN checkout method:

- log in to MS OneDrive and make the OneDrive operational. First create a OneDrive account if necessary.

- Use Tortoise SVN to install the practicum documents from the declaration and place (s) in your own environment.
- For a ZIP download - download the indicated zip and install it in the OneDrive documents dir.
- For an SVN check-out way: Use the context menu with the right mouse button. SVN is in this menu.
- Select "SVN Checkout ...." and then:

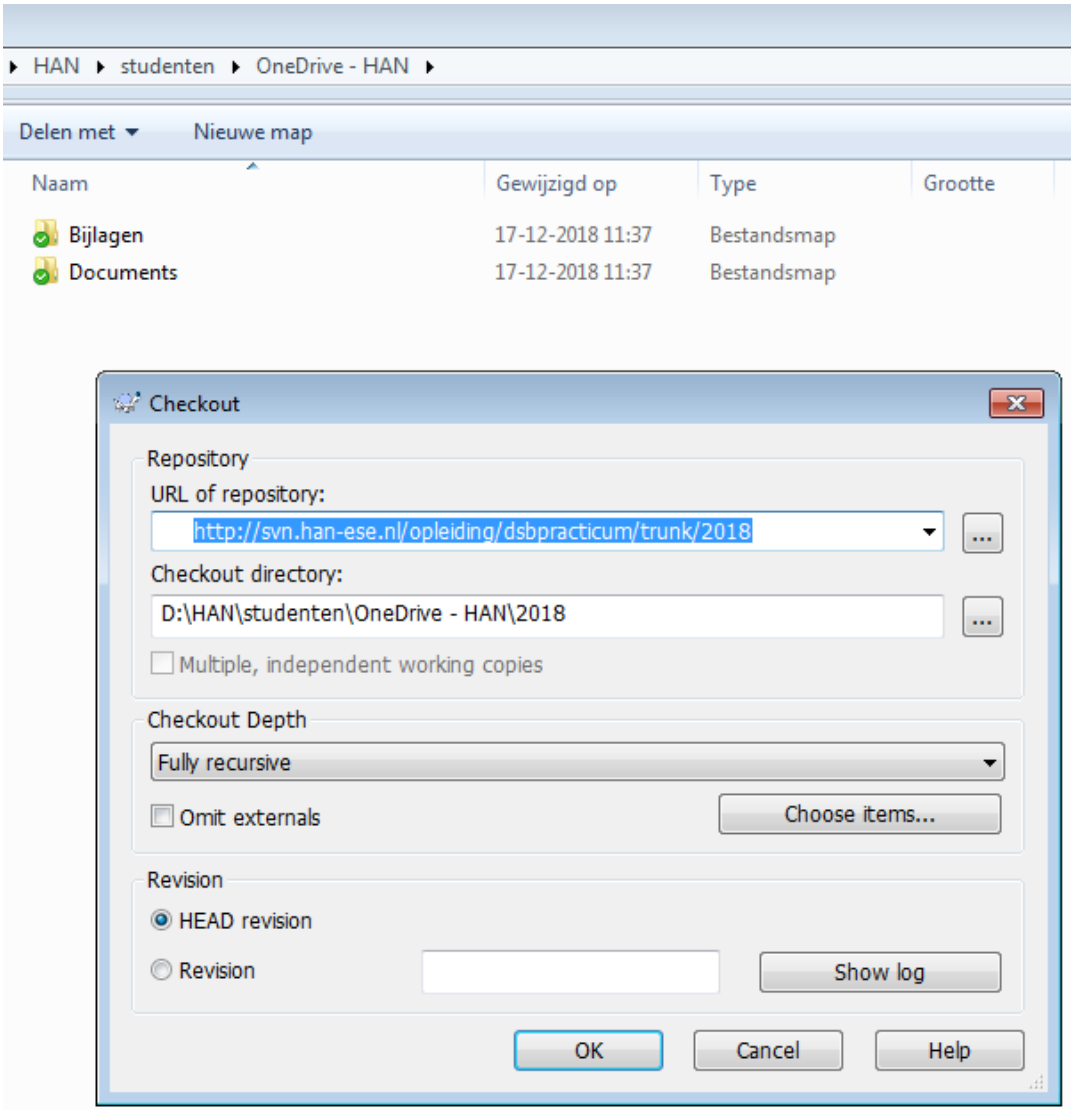


Image 4: Checking out from an SVN repository

A series of files will be copied from the server to the OneDrive directory.
The pieces of prepared software are in the "software" directory
The assignments and other documents can be found in the "docs" directory.

It is not possible to write things back to the repository, it is a Read Only repository for students. Your own changes are saved in the OneDrive directory.

In the event of a change in the assignment by the teacher, an update is necessary. This is indicated during the practicum and executed with the "SVN Update" command.

3.3 Language selection

The practical work can be performed in two languages:

- English
- Dutch

It is **imperative** to indicate which language will be using in the file “DSB.cmake”. This file can be found in the practical work directory:

Image 5: Location of the DSB.cmake file

Adapt the setting in this file to select either Dutch or English:

```
25
26  ## BELANGRIJK ..... IMPORTANT ........
27  ## Zet hier de gewenste taal :
28  ## Select the preferred language :
29  set(InterfaceTaal Nederlands)   # Voor Nederlands
30  #set(InterfaceTaal English)      # For English
31
32  if (NOT InterfaceTaal)
```

Image 6: Specifying the language selection

3.4 Prepared software for each assignment

The practical work assignments have been prepared in advance. They are constructed to provide a software framework that must be used when completing the assignments.

When loading assignment software, at first a project description must be generated. This depends on the software development environment that is used.

3.4.1 Microsoft Windows

- The desktop software assignments on Microsoft Windows will be performed using Microsoft Visual Studio. On this platform a MS Visual Studio Solution (*.sln) must be generated.

  Use the CMake GUI to generate s solution. Indicate in the GUI where the solution must be put (do not choose the software source directory):

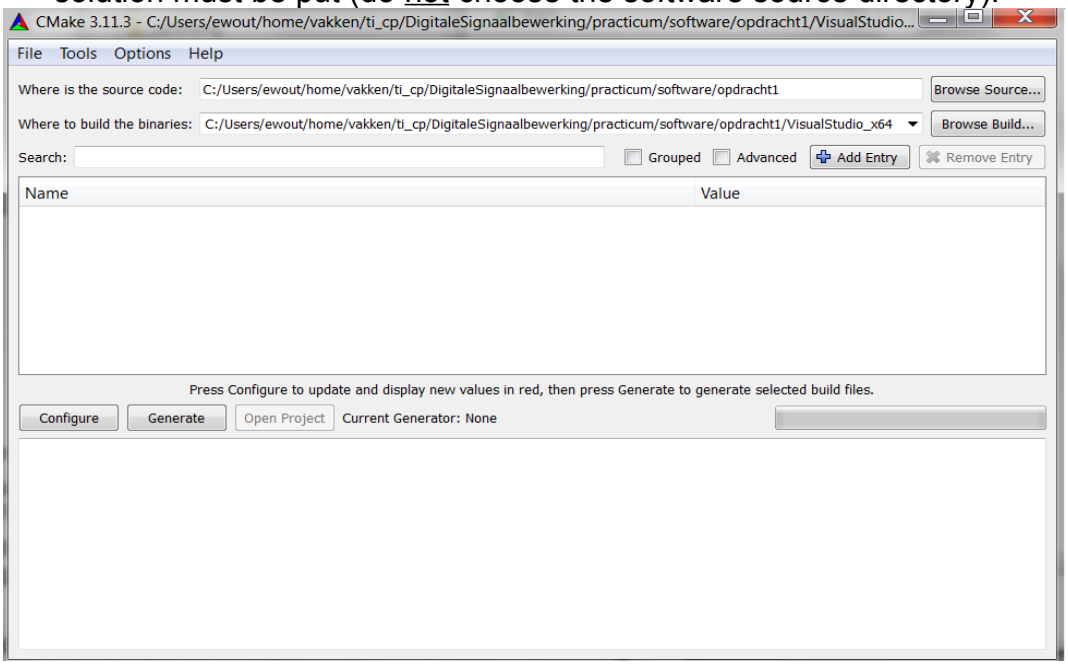


Image 7: Construction of Visual Studio Solution based on the CMake project description(1)

It is of great practical importance that the VisualStudio building directory is not in the OneDrive structure. When placing the building directory in OneDrive, all (very large) Visual Studio db and temporary files are included in OneDrive, which increases the synchronization time of OneDrive and reduces the used online storage space disadvantageously.
So create the building directory in the student home directory and call it VisualStudio so that you can easily find it.

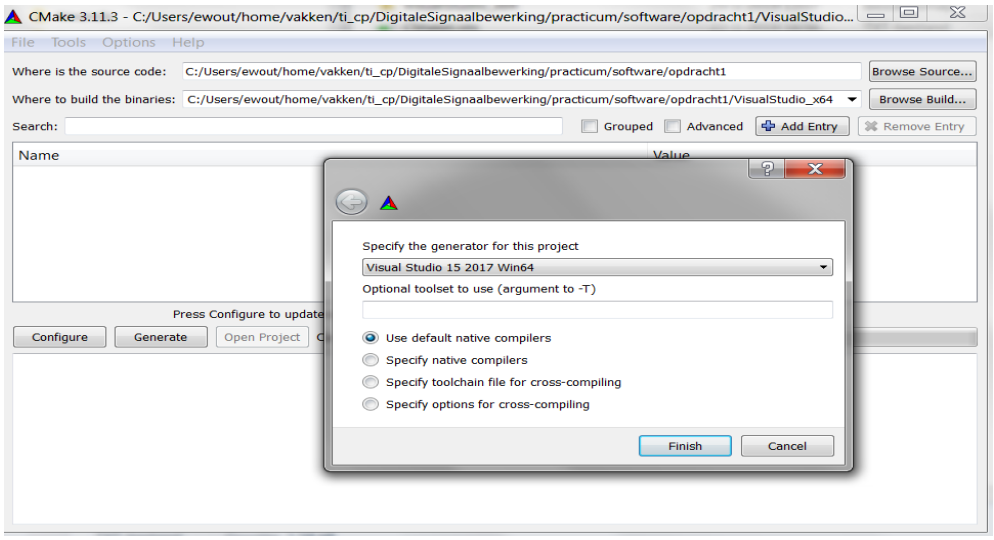


Image 8: Construction of Visual Studio Solution based on the CMake project description(2)

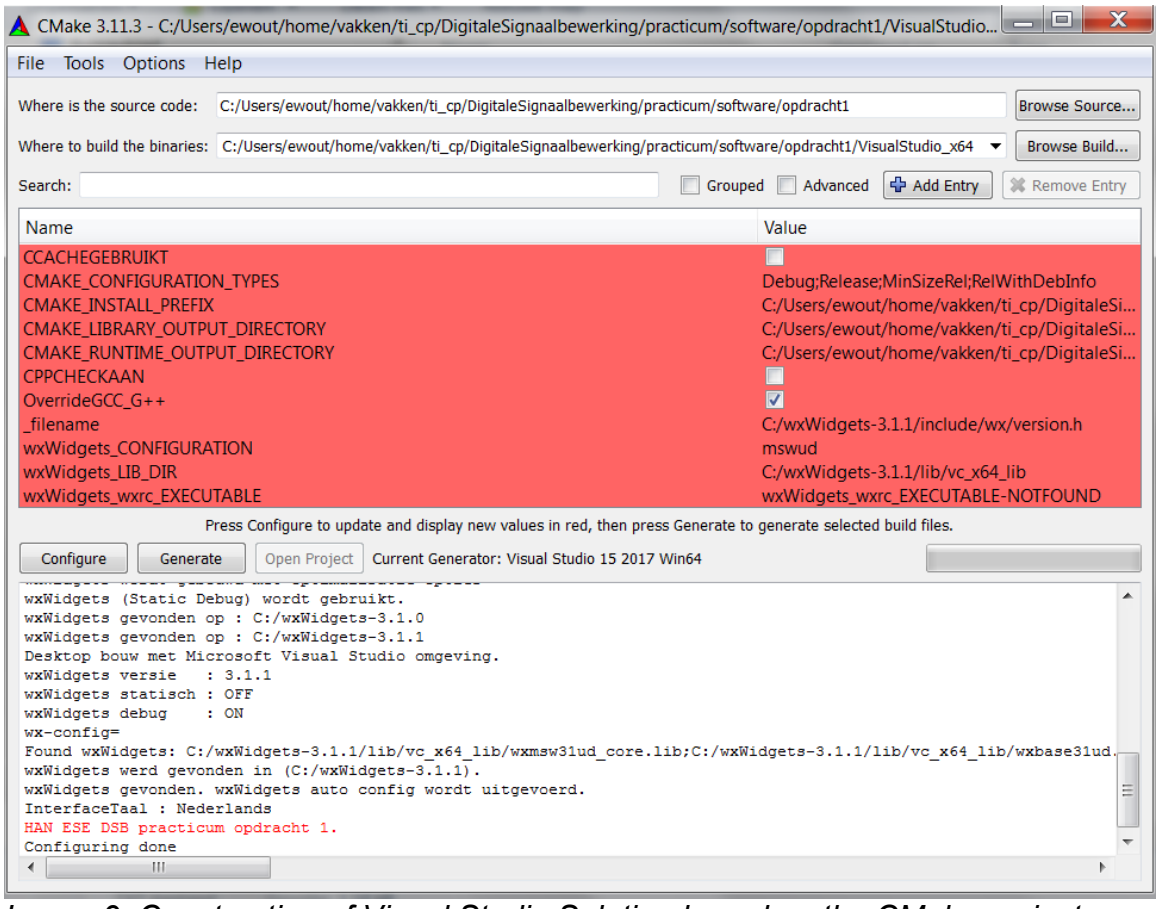


Image 9: Construction of Visual Studio Solution based on the CMake project description(3)

- After construction (“generate”) of the Solution, a solution exists. This solution can be downloaded into Visual Studio 2017.
- After loading the solution, select the sub project that must be executed first – pick the most obvious one.

In addition to the above described method, newer versions of Visual Studio (2017 and up) are capable of loading CMake based projects directly. See the Desktop IDE documentation for more details.

3.4.2 Apple Mac OS X and Linux

The desktop software assignments on Apple Mac OS X and Linux and the embedded assignments will be executed using CLion:

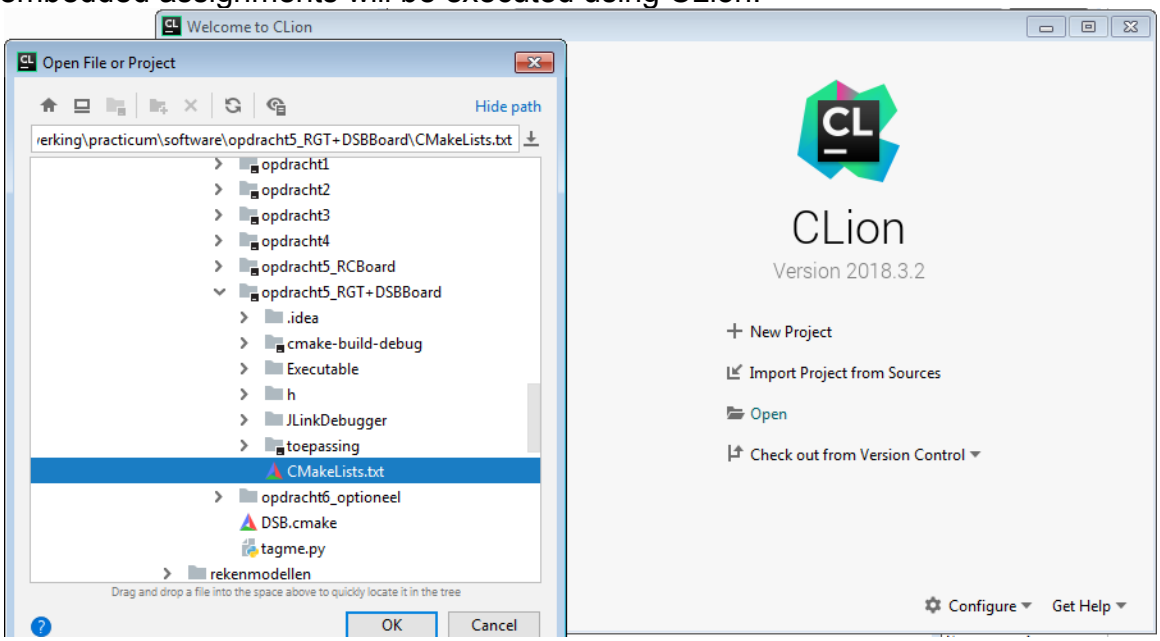


Image 10: CLion is built to load CMake files directly

Load the CMakeLists.txt file als a project into CLion. CLion wil automatically configure itself to build to project whether it is for a desktop or an embedded assignment.

3.5 Completion of the assignments

Each assignment contains missing pieces of software, which can be recognized by #error markers. It is is required to replace the error markers with the solution of the assignment.

```
void GemWaardeVenster::drawDataHandler(wxCommandEvent &event)
{
#error "Dit stuk software ontbreekt / This part of the software is missing !!"
/* Geachte student,
    Dit stuk ontbreekt. Werk dit uit om de opdracht uit te voeren.
     Honoured student,
     This part of the software is missing. Fill in your implementation to fulfill the task.


    gemVeranderd = false;
}
```

Image 11: Example of error markers during software development

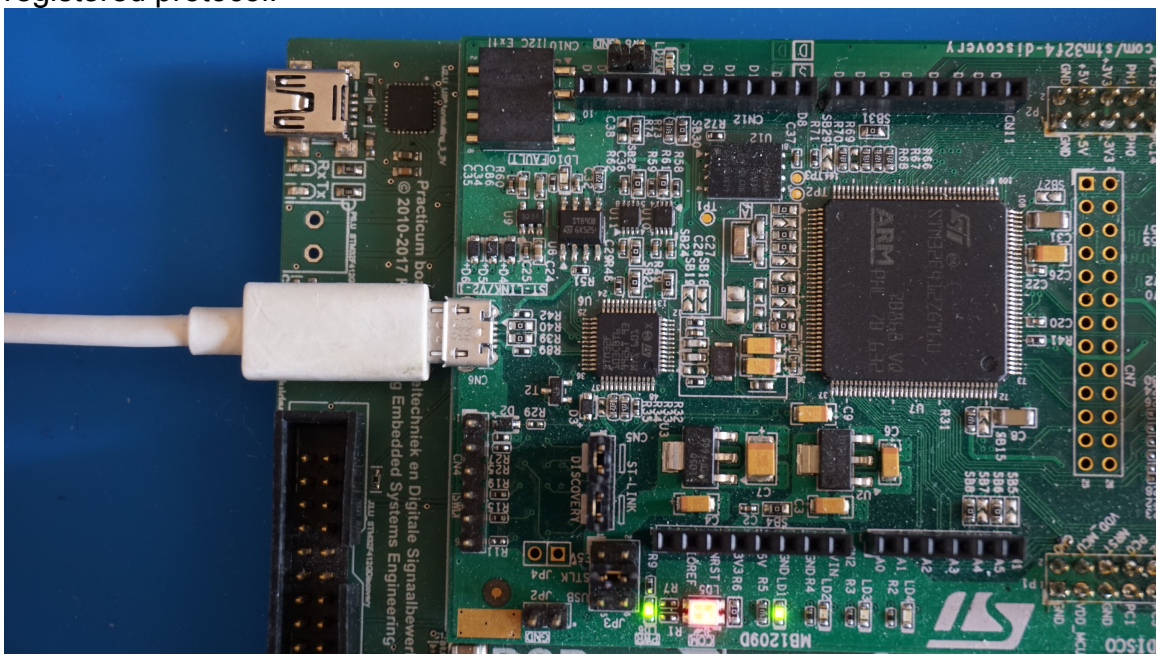
When matters are unclear, go to the teacher and ask him for clarification. Do not remain silent until the end of the practical work session!

3.6 Communication between PC and DSB board

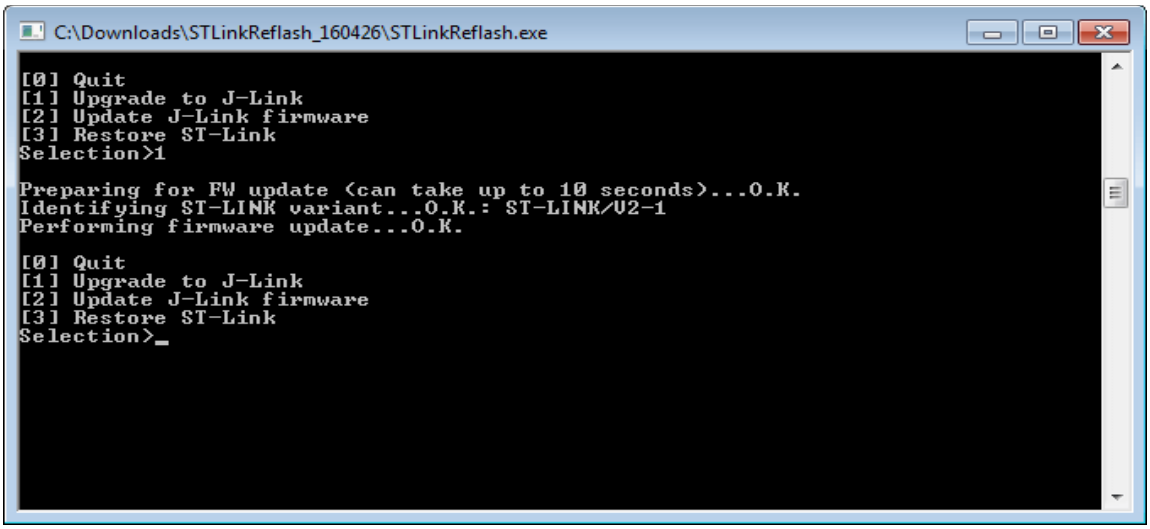
The development board for assignment 5 employs the Segger J-Link protocol for debugging purposes. Segger J-Link offers greater speeds, reliability and cross-platform application over the standard ST-Link protocol usually found on the boards after purchase. You do need a Windows machine to do this – there aren't OS X and Linux versions of the tool.

When the board is connected, make certain that the debugging front-end has been switched over to Segger J-Link.

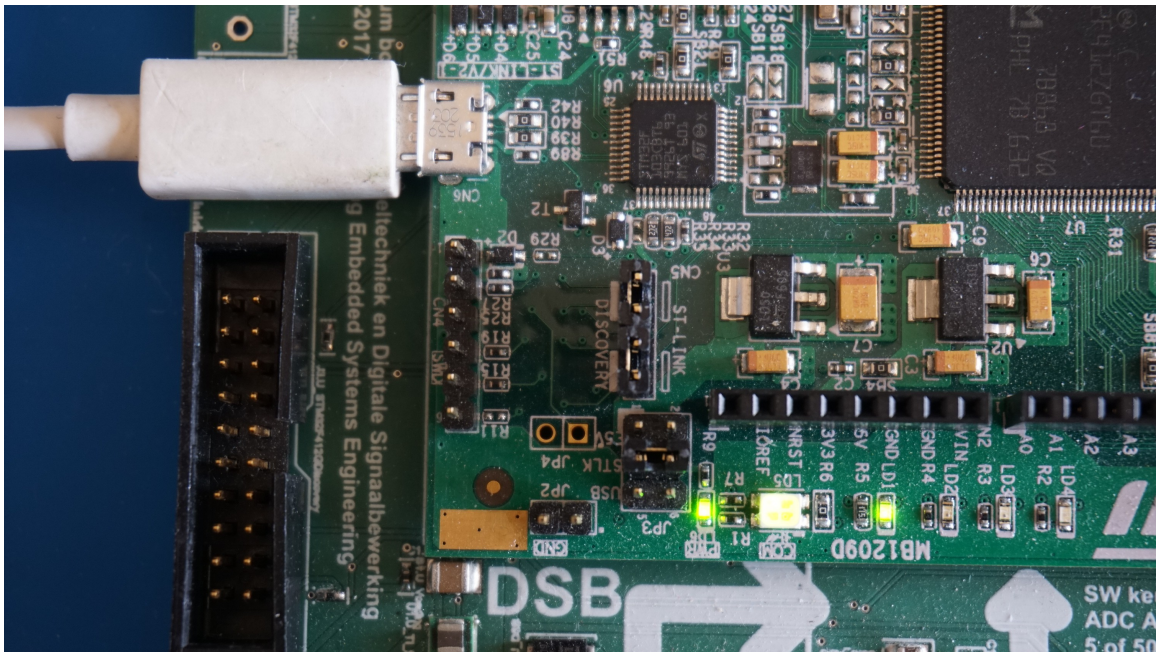
The debugger front-end will show a red LED (centre bottom) when ST-Link is the registered protocol:



In order to execute the assigments it is mandatory to replace the ST-Link with the Segger J-Link. The procedure is detailed on Segger's website and quite easy:



After the update, the debugger LED will show a green colour:



4 Delivery of proof of assignment completion

By default, delivery is done in this way:

- Demonstration of working software must be performed during the practical sessions to the teacher.
- Provide a short and simple report for each couple of 2 students. Give the report the name: ***<nickname>_<assignment name>* .pdf**:
  * .doc [x] files and * .rar files will not be accepted! Hand in the report through email.
- Each assignment has a time period assigned. This period is not binding but indicative for a good progress during the course. In case you experience a lack of time when an assignment cannot be completed, you must discuss the case with the teacher.