



Installatie Desktop IDE en wxWidgets ToolKit bij DSPESEL (DSBprt)

voor

HAN Elektrotechniek/Embedded Systems Engineering

Inhoudsopgave

1	Software IDE installatie.....	2
1.1	Jetbrains CLion (Apple Mac OS X en Linux).....	3
1.1.1	Installatie van CLion.....	4
1.1.2	Installatie van wxWidgets.....	4
1.1.3	Gebruik van CLion.....	5
1.2	Microsoft Visual Studio (Microsoft Windows).....	6
1.2.1	Installatie van wxWidgets onder MS Windows.....	9
1.2.2	Installatie van CMake	11
1.2.3	Gebruik van CMake met Visual Studio.....	11

1 Software IDE installatie

Voor software ontwikkeling van desktop software bestaan meerdere mogelijkheden. HAN ESE ondersteunt de mogelijk om het DSPESEL practicum uit te voeren op de onderstaande platforms:

- Apple Mac OS X
- Microsoft Windows 7 en hoger
- Ubuntu Linux of FreeBSD



Het DSPESEL practicum wordt voor een belangrijk deel uitgevoerd op basis van de [wxWidgets toolkit](#). De keuze voor deze toolkit is gebaseerd op de volgende punten:

- wxWidgets is een volwassen cross-platform GI toolkit die ook commercieel goed inzetbaar is. De opzet van de toolkit is vergelijkbaar met Qt.
- WxWidgets is volledig open-source.
- WxWidgets produceert applicaties die niet te onderscheiden zijn van de platform standaard.
- Studenten moeten een verscheidenheid aan tools leren, en niet alleen andere gangbare platforms (zoals Qt) kennen.

Een vergelijking tussen wxWidgets en Qt is [hier](#) te vinden.

In de volgende paragrafen wordt uitgelegd hoe wxWidgets gebruikt wordt in het practicum. Op Unix platforms wordt gebruik gemaakt van JetBrains CLion, op Microsoft Windows wordt gebruikt gemaakt van MS Visual Studio.

Het DSPESEL practicum maakt gebruik van CMake voor de projectstructuur.

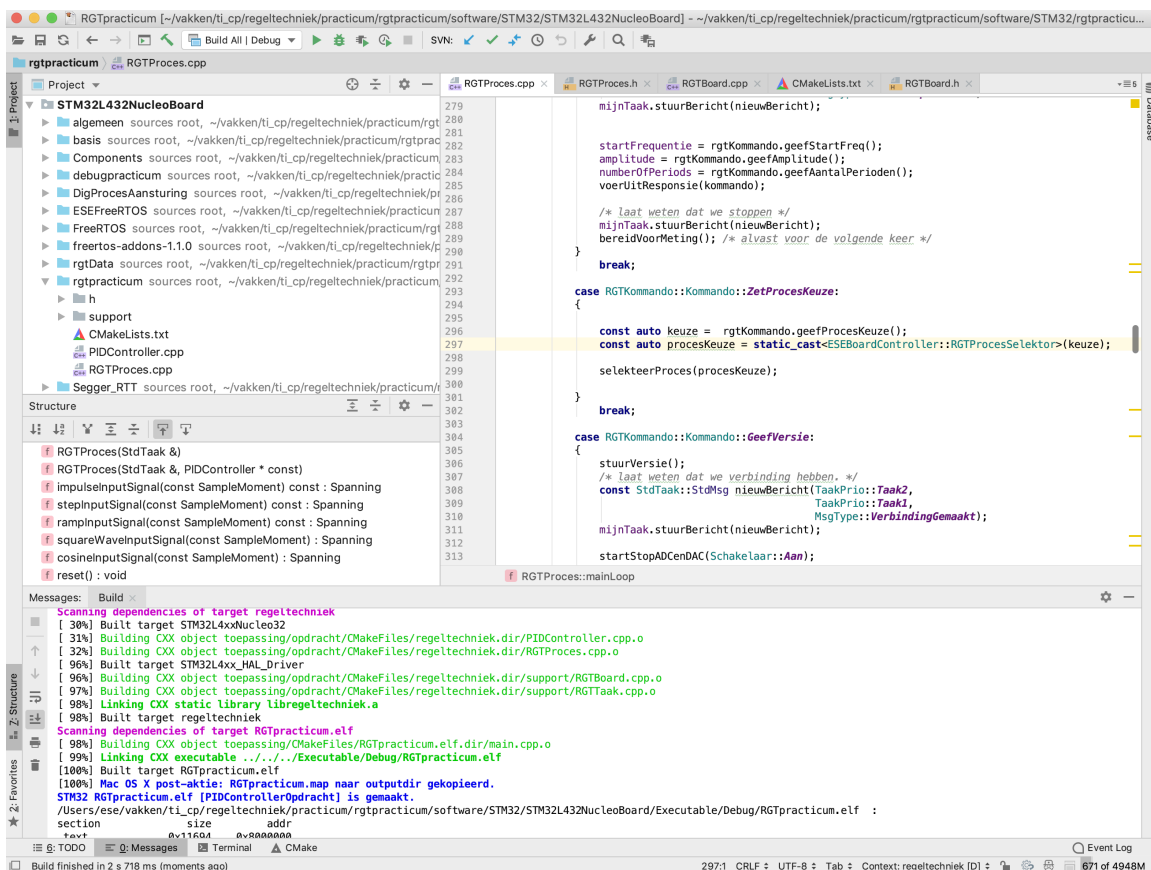


1.1 Jetbrains CLion (Apple Mac OS X en Linux)



CLion is een superieure ontwikkelomgeving (IDE) voor softwareprojecten die gebaseerd zijn op de [CMake](#) projectstructuur.

CLion ziet er uit zoals hieronder weergegeven (Mac OS X) :



Jetbrains CLion

De voordelen van CLion zijn:

- Een superieure IDE om mee te werken met modere features als refactoring, zeer goede navigatie en volledige support voor versiebeheer met SVN, Git en anderen.
- Een veel betere ondersteuning van moderne C++ varianten (momenteel : C++14).
- Flexibele projectopzet gebaseerd op [CMake](#).
- Een superieure en zeer snelle debugger.

1.1.1 Installatie van CLion

Download [Jetbrains Clion](#) voor het gebruikte platform.

Run de installer. Indien fouten optreden, dan is waarschijnlijk iets niet goed geïnstalleerd bij de voorgaande stappen. CLion komt met een 30 dagen probeerperiode, daarna is een licentie vereist.

Optioneel : Meld je meteen aan bij Jetbrains om een studentlicentie te verkrijgen. Let op bij het verkrijgen van een licentie dat je een officieel student.han.nl email adres gebruikt (geen google/hotmail enz.) voor identificatie.

1.1.2 Installatie van wxWidgets

De installatie kan worden uitgevoerd moet worden uitgevoerd in diverse stappen.

Ga als volgt te werk:

1. [Alleen Apple Mac OS X]

Indien niet aanwezig, download en installeer Apple XCode via de App Store.



Installeer via [MacPorts](#) (of HomeBrew) de wxWidgets toolkit, en optioneel de LLVM/clang tools en de svn binaries. :

- [Macports wxWidgets link](#)
- [Macports LLVM Link](#)
- [MacPorts SVN link](#)

2. [FreeBSD/Linux]



Indien niet aanwezig, download en installeer de gcc en g++ en of llvm c++ compilers voor Ubuntu Linux. FreeBSD heeft standaard llvm clang C/C++ compilers geïnstalleerd.

In FreeBSD, installeer via Packages/Ports de wxgtk3 toolkit en optioneel subversion. In Ubuntu, doe iets soortgelijks via apt-get (niet helemaal duidelijk, verandert nogal vaak). Zie de [wx installatiegids](#) voor meer informatie.

1.1.3 Gebruik van CLion

CLion kan gebruikt worden voor de bouw van een (embedded) softwareproject als een geschikt CMake projectbestand (CMakeLists.txt) beschikbaar is, waarin zijn gedefinieerd:

- De gebruikte compiler toolchain.
- De target processor.
- Alle opties voor de assembler, compiler en linker in de toolchain.
- Alle bronbestanden die in de binary gecompileerd moeten.
- Alle paden voor headerbestanden.
- De gewenste vorm van de binary.
- Eventuele custom stappen die nodig zijn in het project.

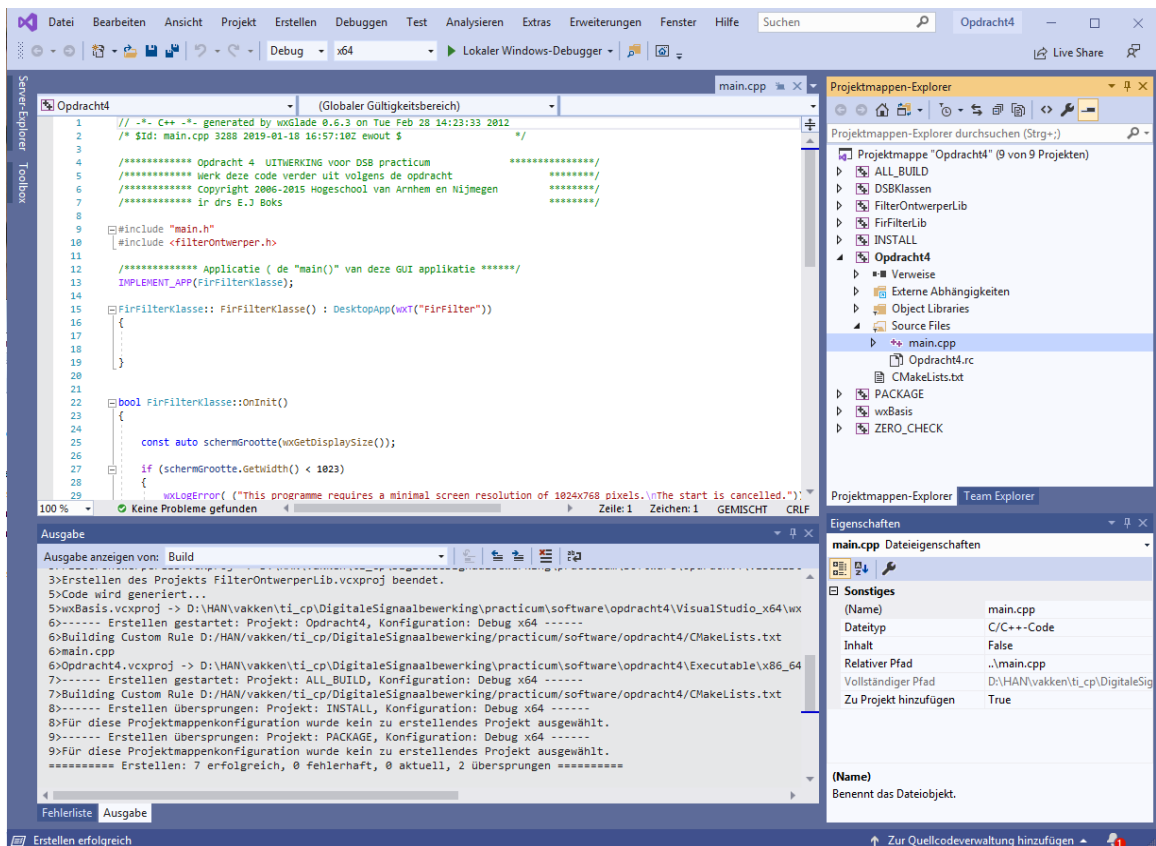
Op internet zijn diverse tutorials en veel informatie over CMake te vinden. In de practica wordt uitgelegd hoe elk practicumproject moet worden geconfigureerd voor Cmake, Clion en Ozone.

1.2 Microsoft Visual Studio (Microsoft Windows)

Op het Microsoft Windows platform wordt gebruikt gemaakt van Microsoft Visual Studio.

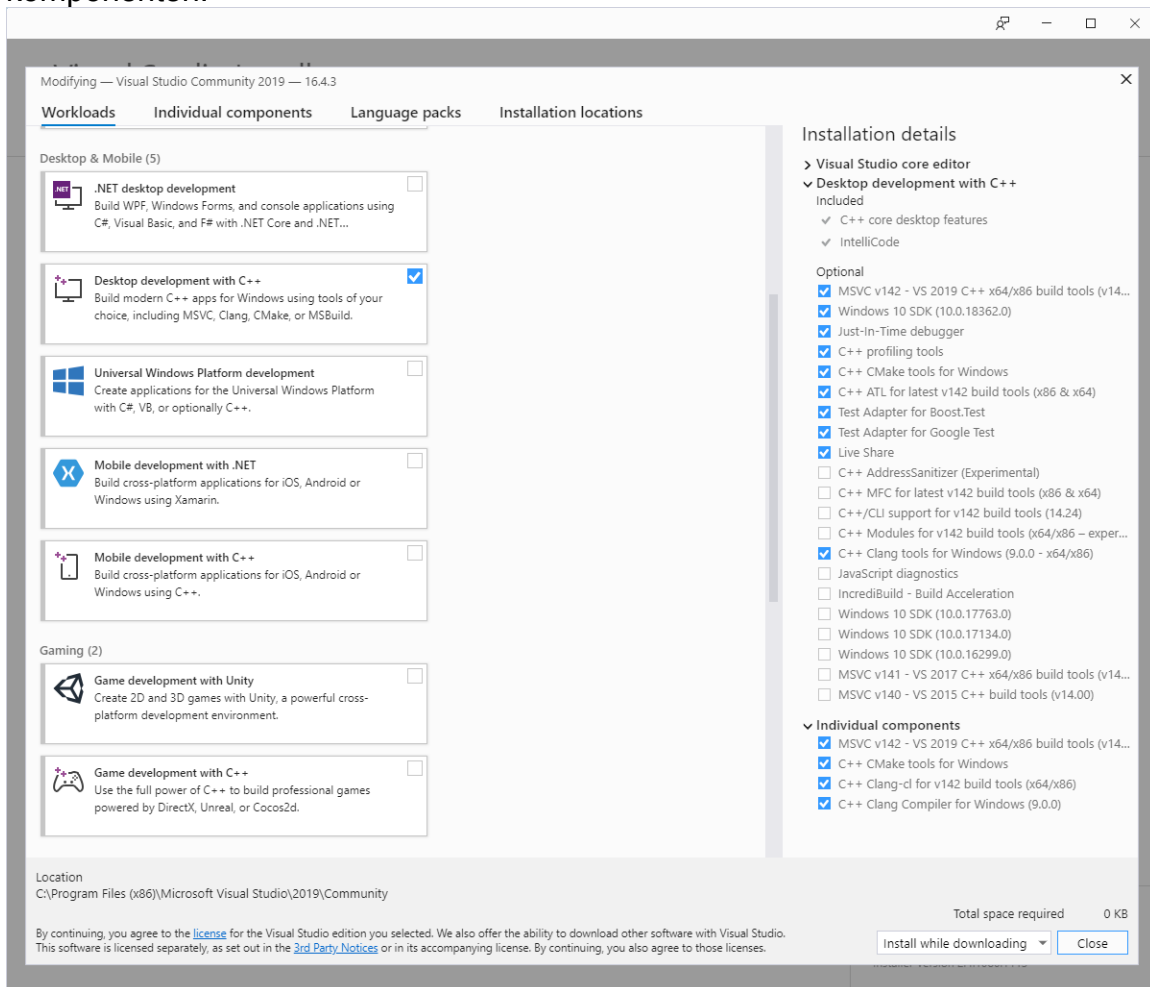


Visual Studio is een professionele en zeer prettige IDE die (uiteraard) zeer geschikt is voor het schrijven van applicaties voor Microsoft Windows. De reden dat op Windows voor Visual Studio gekozen is in plaats van CLion is het feit dat de Microsoft compiler voor Windows betere binaries aflevert dan de GCC compiler waar CLion op Windows mee werkt. Microsoft gaat met zijn tijd mee en integratie met tools die in de Unix wereld gangbaar zijn is steeds meer vanzelfsprekend, waardoor Visual Studio gemakkelijker te hanteren is naast andere tools.



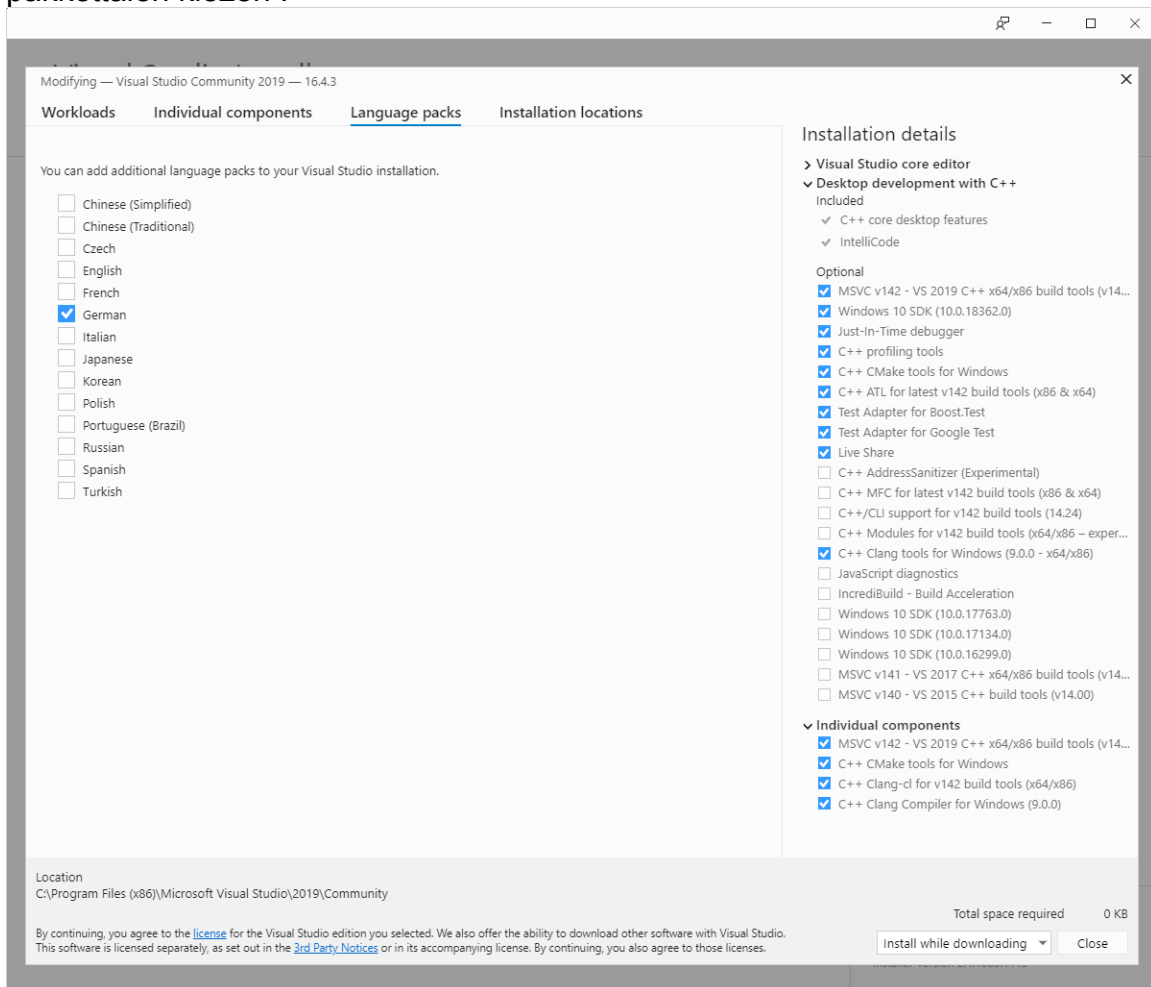
Microsoft Visual Studio Community 2019

Een installer voor Visual Studio heet [VS Community](#). De installatie volgt na het starten van deze installer door de keuze van de (voor ESE) gebruikte componenten:



Kies voor “Desktop development with C++” en vink ook CMake en CLang tools aan. Andere opties zijn minder relevant.

Zoals in bovenstaande afbeelding van Visual Studio te zien is, kan voor een andere taal dan Engels gekozen worden. Wie dit fijn vindt, kan uit een aantal pakketten kiezen .



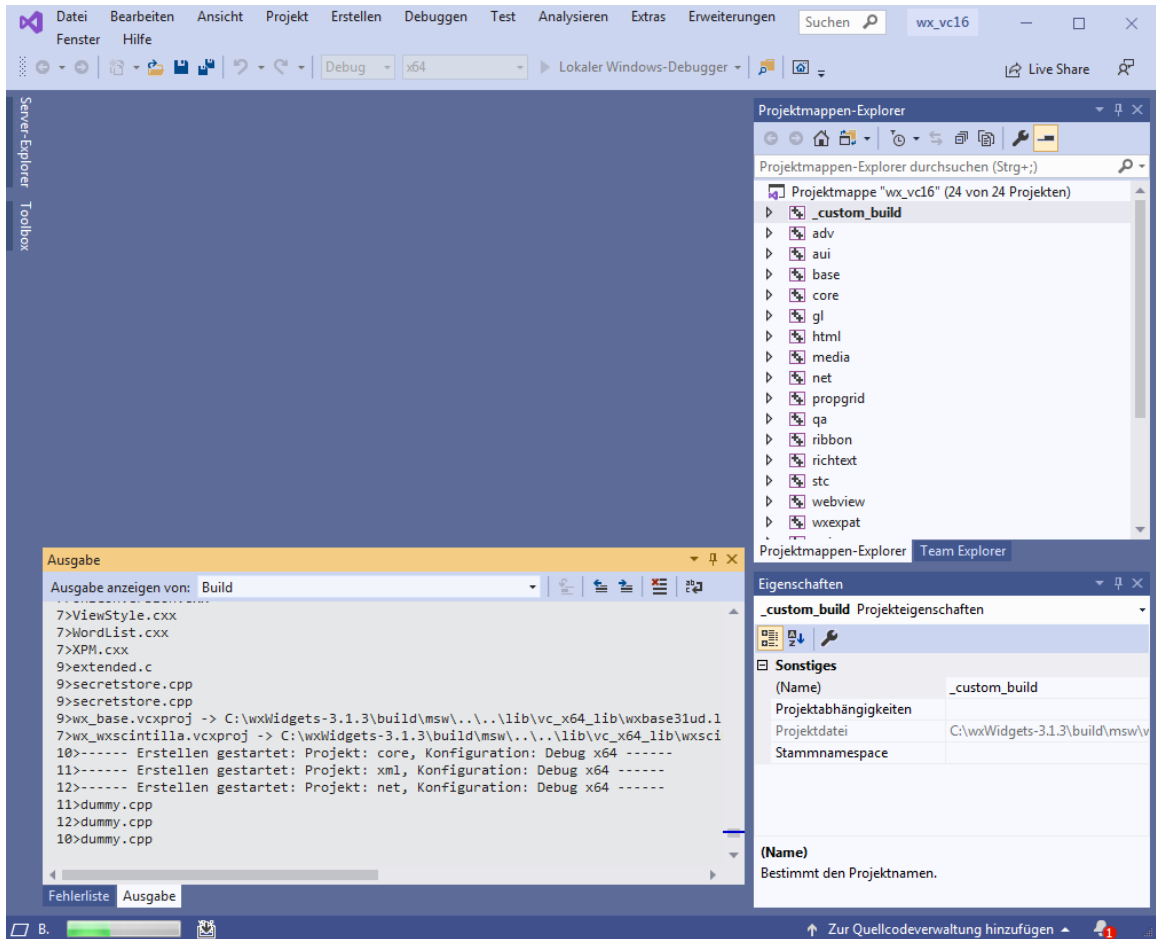
Na installatie van Visual Studio is het raadzaam deze te koppelen aan een Microsoft account (voor bijvoorbeeld OneDrive).

1.2.1 Installatie van wxWidgets onder MS Windows.

1. Download de wxWidgets win32 installer vanaf [de wxWidgets website](#) op de standaard lokatie (C:/wxWidgets_x.xx).
2. Start Visual Studio en navigeer naar de map waar een vc_15/vc_16 solution klaar staan ([C:/wxWidgets_x.xx/build/msw/wx_vc15](#) of 16):

Naam	Gewijzigd op	Type	Grootte
wx_vc9_wxscintilla.vcxproj	27-10-2019 18:41	VC++ Project	43 kB
wx_vc9_wxtiff.vcxproj	27-10-2019 18:41	VC++ Project	23 kB
wx_vc9_wxlib.vcxproj	27-10-2019 18:41	VC++ Project	20 kB
wx_vc9_xml.vcxproj	27-10-2019 18:41	VC++ Project	36 kB
wx_vc9_xrc.vcxproj	27-10-2019 18:41	VC++ Project	46 kB
wx_vc10	27-10-2019 18:41	Microsoft Visual S...	38 kB
wx_vc11	27-10-2019 18:41	Microsoft Visual S...	38 kB
wx_vc12	27-10-2019 18:41	Microsoft Visual S...	38 kB
wx_vc14	27-10-2019 18:41	Microsoft Visual S...	38 kB
wx_vc15	27-10-2019 18:41	Microsoft Visual S...	38 kB
wx_vc16	27-10-2019 18:41	Microsoft Visual S...	38 kB
wx_webview.vcxproj	27-10-2019 18:41	VC++ Project	39 kB
wx_webview.vcxproj.filters	27-10-2019 18:41	VisualStudio.vcxpr...	3 kB
wx_wxexpat.vcxproj	27-10-2019 18:41	VC++ Project	26 kB
wx_wxexpat.vcxproj.filters	27-10-2019 18:41	VisualStudio.vcxpr...	1 kB
wx_wxjpeg.vcxproj	27-10-2019 18:41	VC++ Project	28 kB
wx_wxjpeg.vcxproj.filters	27-10-2019 18:41	VisualStudio.vcxpr...	6 kB
wx_wxpng.vcxproj	27-10-2019 18:41	VC++ Project	27 kB
wx_wxpng.vcxproj.filters	27-10-2019 18:41	VisualStudio.vcxpr...	2 kB
wx_wxregex.vcxproj	27-10-2019 18:41	VC++ Project	29 kB
wx_wxregex.vcxproj.filters	27-10-2019 18:41	VisualStudio.vcxpr...	2 kB
wx_wxscintilla.vcxproj	27-10-2019 18:41	VC++ Project	42 kB
wx_wxscintilla.vcxproj.filters	27-10-2019 18:41	VisualStudio.vcxpr...	19 kB
wx_wxtiff.vcxproj	27-10-2019 18:41	VC++ Project	30 kB
wx_wxtiff.vcxproj.filters	27-10-2019 18:41	VisualStudio.vcxpr...	6 kB
wx_wxlib.vcxproj	27-10-2019 18:41	VC++ Project	27 kB
wx_wxlib.vcxproj.filters	27-10-2019 18:41	VisualStudio.vcxpr...	2 kB
wx_xml.vcxproj	27-10-2019 18:41	VC++ Project	38 kB
wx_xml.vcxproj.filters	27-10-2019 18:41	VisualStudio.vcxpr...	3 kB
wx_xrc.vcxproj	27-10-2019 18:41	VC++ Project	46 kB
wx_xrc.vcxproj.filters	27-10-2019 18:41	VisualStudio.vcxpr...	17 kB

3. Bouw de solution in 64 bit mode (dus niet in 32 bit mode !) in statische debug en optioneel statisch release mode. DLL mode hoeft niet :



Na afronding van het bouwen is alles klaar – laat de bouwmap met alle obj bestanden staan, dit is de wxWidgets toolkit voor gebruik in het practicum.

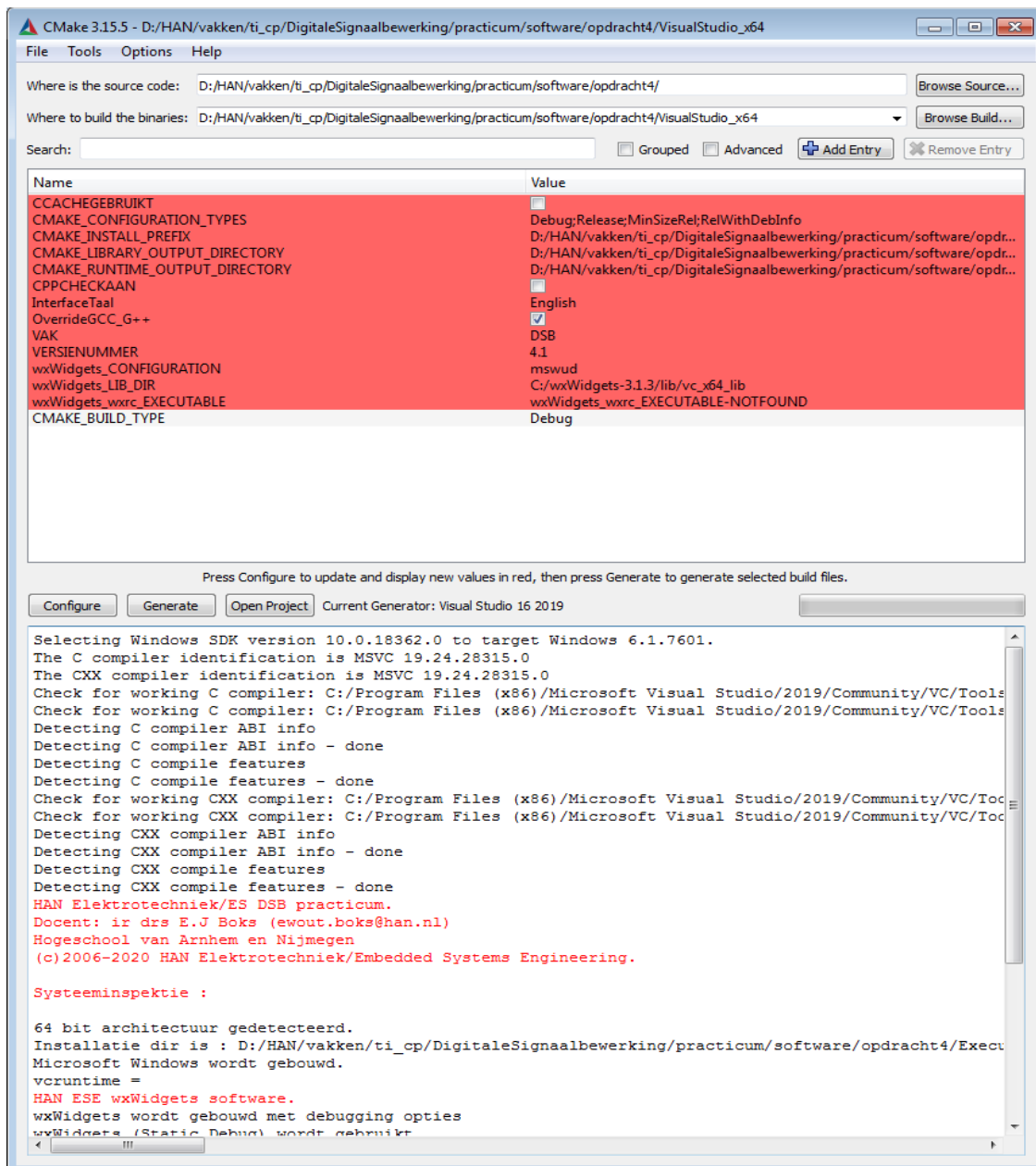
1.2.2 Installatie van CMake

Het is aan te raden om naast Visual Studio ook [CMake](#) te installeren, om zelfstandig Visual Studio Solution projecten te kunnen genereren uit CMake projektbestanden.

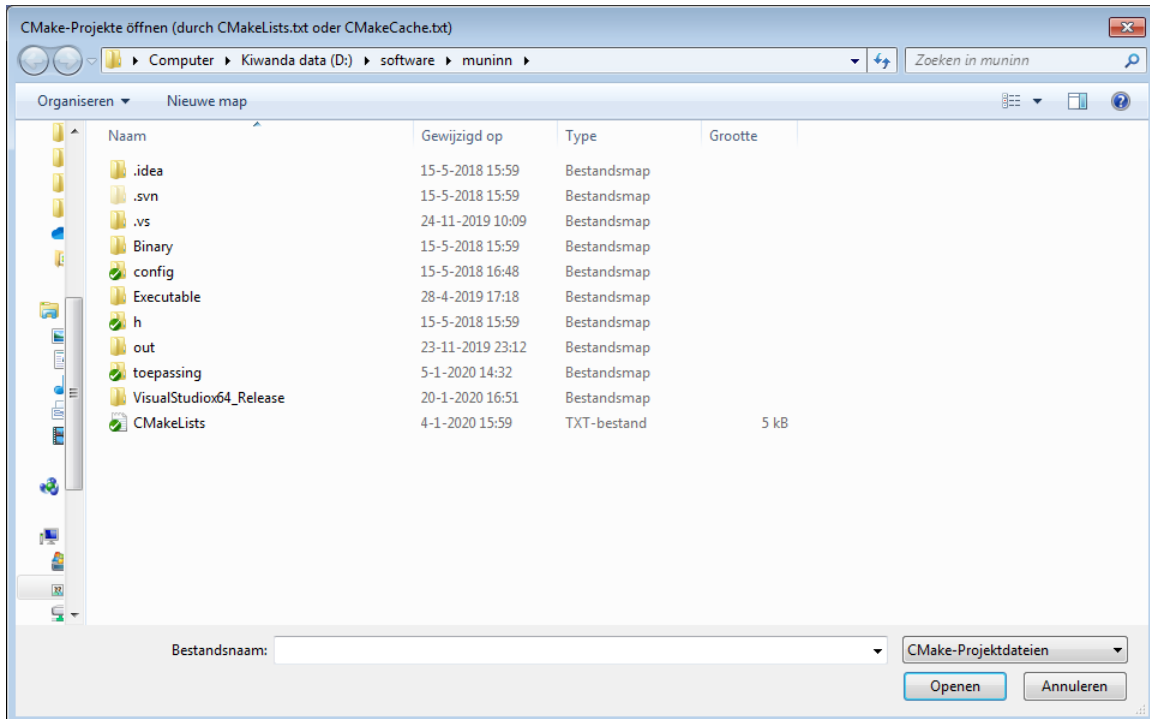
1.2.3 Gebruik van CMake met Visual Studio

Het is op twee manieren mogelijk om de practicumprojecten te gebruiken in Visual Studio:

- De traditionele manier. Met de CMake generator wordt een VS Solution geconstrueerd, welke daarna in Visual Studio wordt ingeladen :

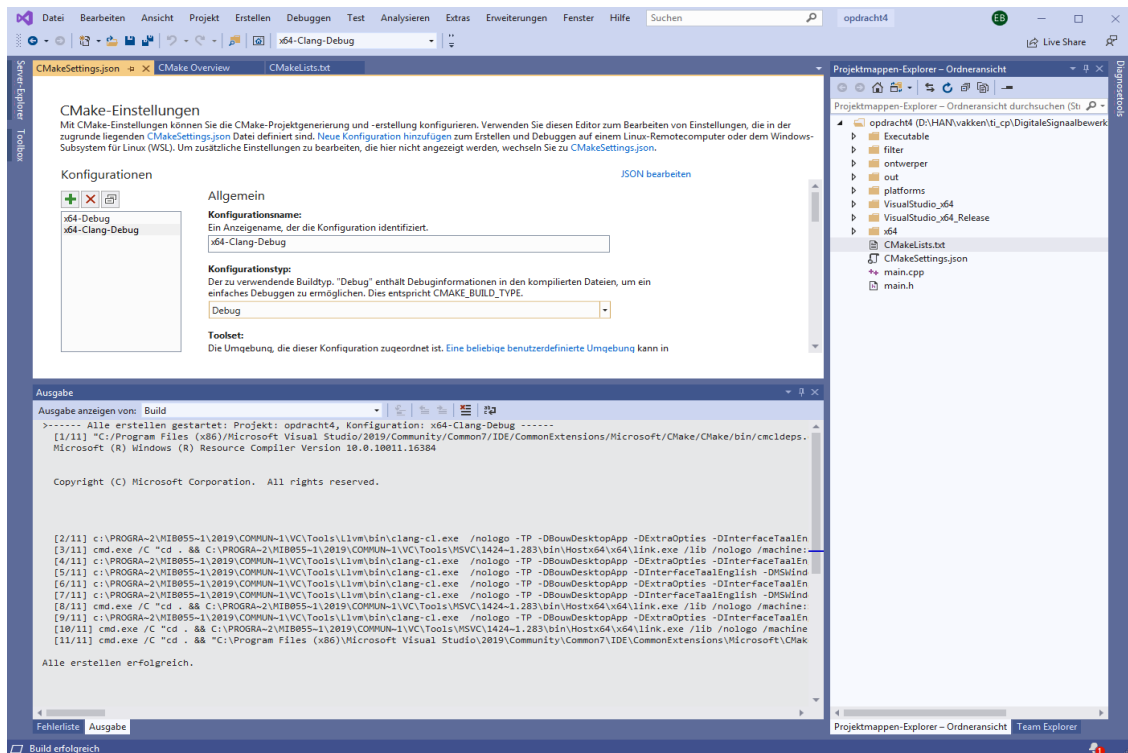


- De “nieuwe” manier. Microsoft heeft sinds VS2017 een koppeling geïntroduceerd waarmee CMake projecten direct in Visual Studio kunnen worden ingeladen. Ga bij Bestand/Open naar de opening optie “CMake”, waar na de bestandsdialoog verschijnt waar mee een CMakeLists.txt bestand kan worden gekozen:

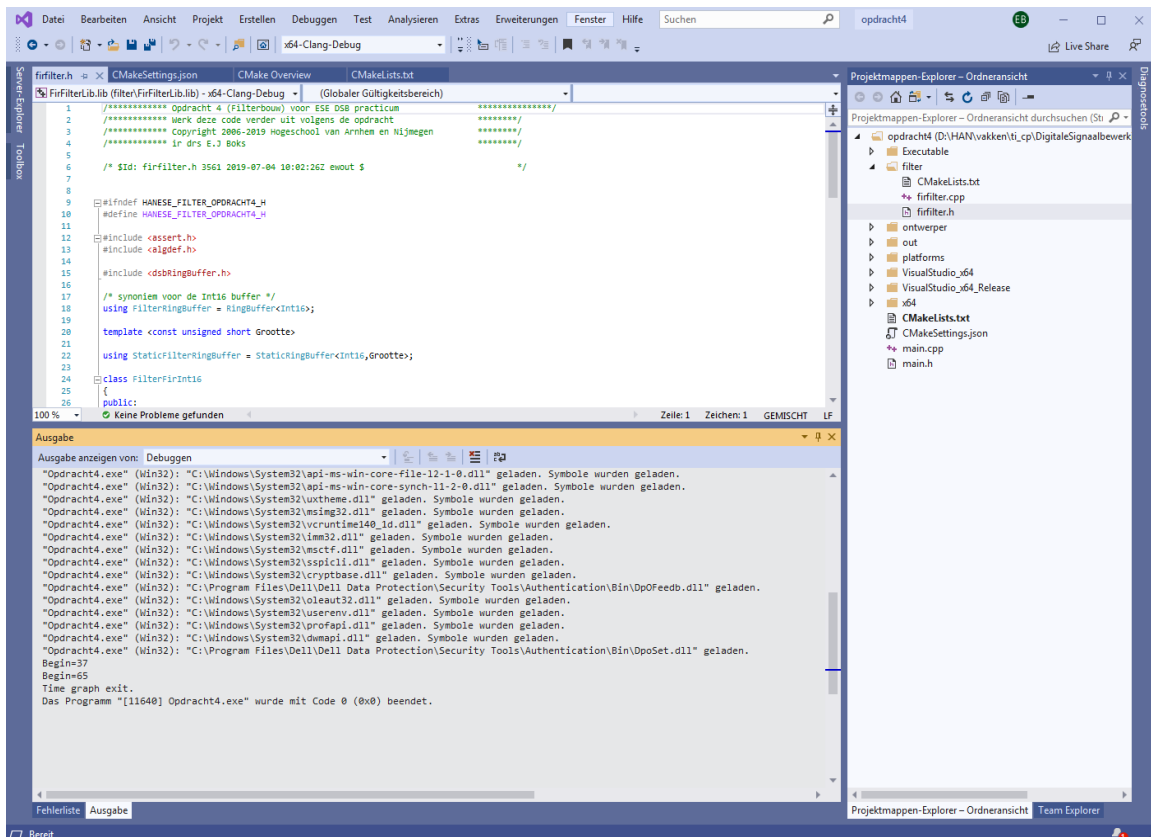


Visual Studio zal hier na het project inladen en op de standaard bouwmethode instellen.

Er kunnen naast deze standaard methode ook andere bouwconfiguraties worden gekozen. Een configuratie op basis van de [LLVM clang compiler suite](#) (de standaard compiler voor Apple Mac OS X en FreeBSD) is nu ook mogelijk ; de wonderen zijn de wereld niet uit :



Hierna kan het project worden gebouwd en uitgevoerd op basis van clang !



Voor vragen en opmerkingen, neem contact op! Ik zal proberen problemen of onduidelijkheden zo snel mogelijk op te lossen.

Arnhem, 30.01.2020

ir drs E.J Boks (ewout.boks@han.nl)