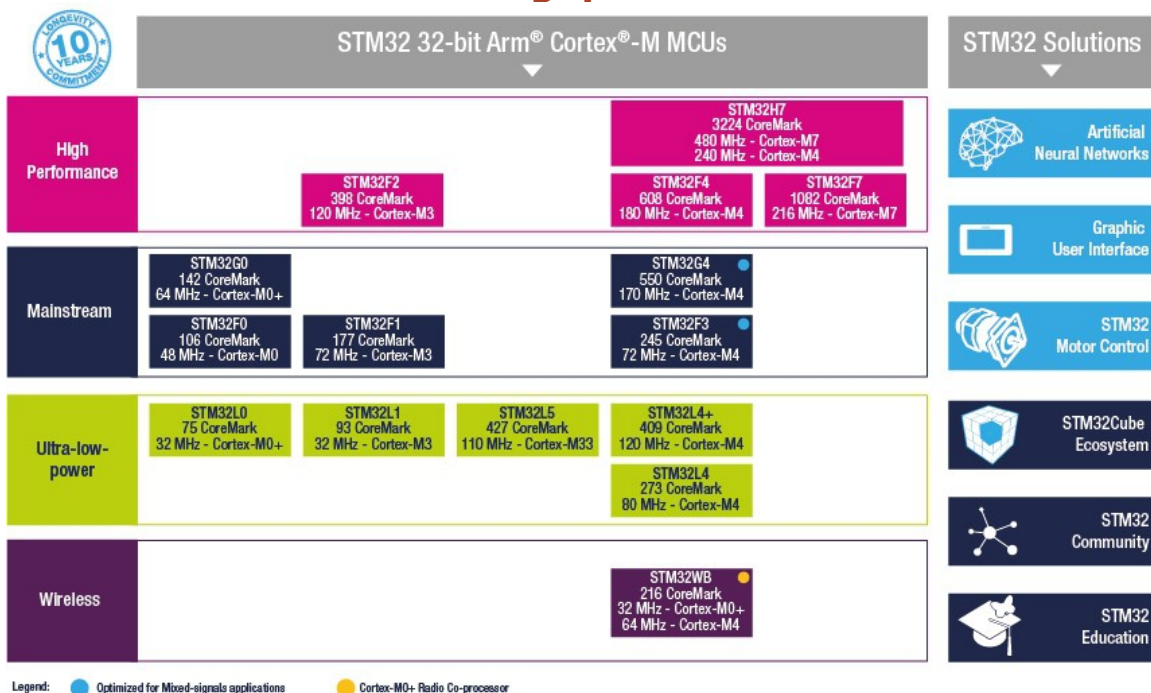


ELT-ESE-2C

ECS Regeltechniek practicum

Debugopdracht



Algemene informatie

Deze opdracht sluit aan bij de debug verdiepingsles, die in het regeltechniek practicum is opgenomen. Er wordt gewerkt met de volgende tools:

- De ontwikkelomgeving [Jetbrains CLion](#) + [CMake](#) + de [GNU ARM compiler](#).
- [Segger Ozone](#) debugger.
- [Segger RTT](#) real-time debug output.

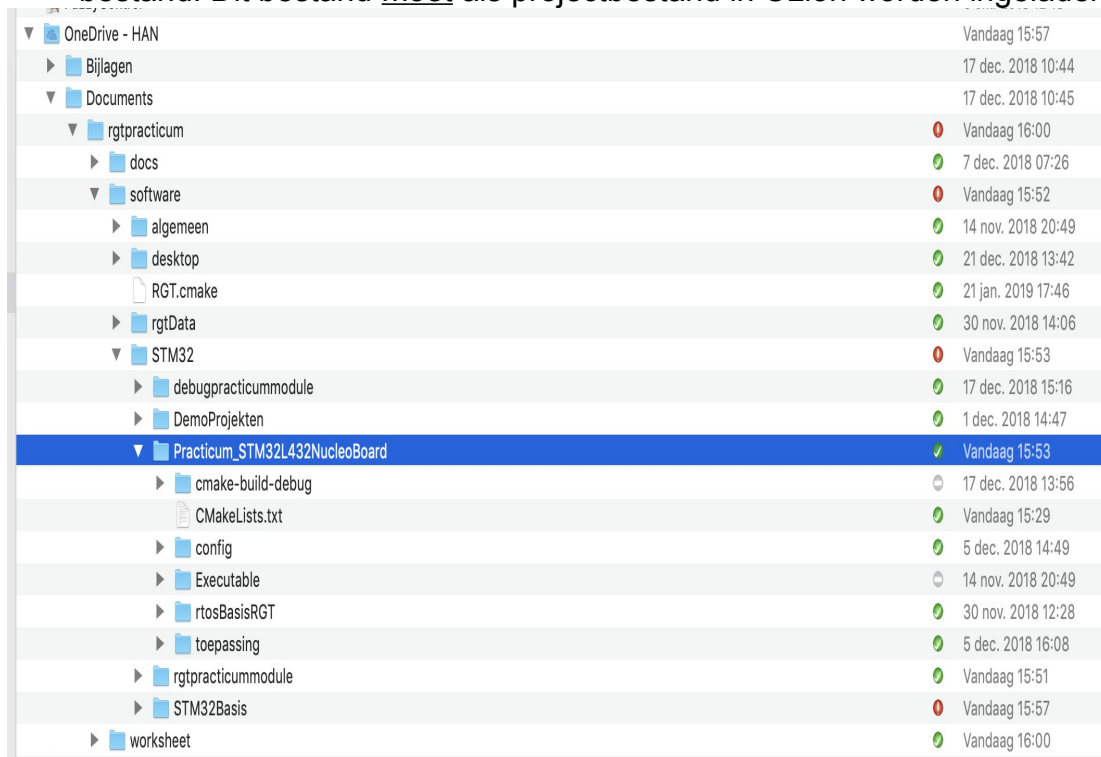
De debugopdracht behelst het schrijven van een driver voor de STM32 RTC BKUP registers.

Let op dat de practicumdirectory in het gehele pad GEEN spaties bevat – dit leidt tot problemen voor de compiler en make tools.

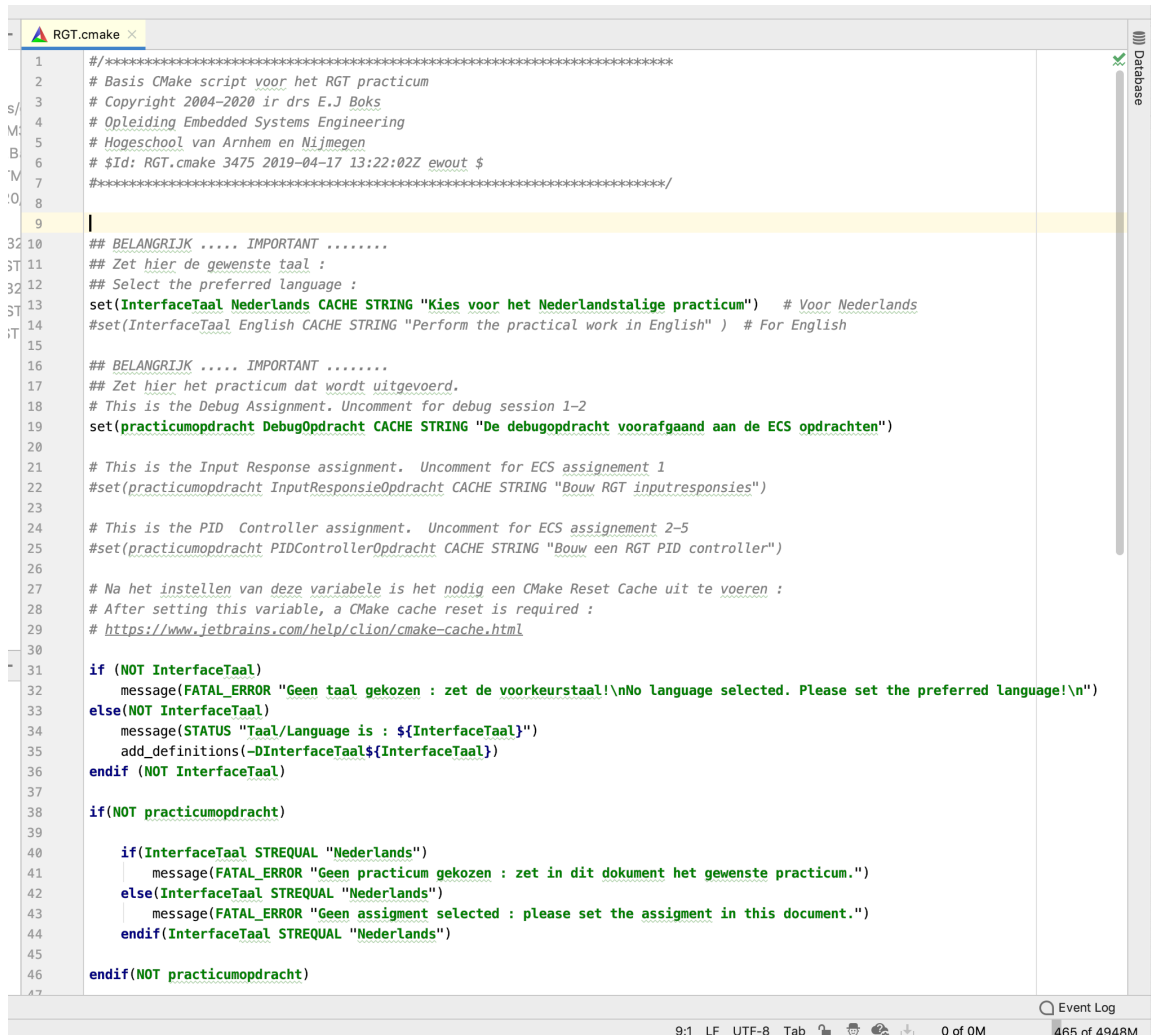
De opdracht

De opdracht vandaag en volgende week luidt:

1. Installeer alle benodigde software zoals door de docent aan het begin van het practicum is uitgelegd.
2. Bestuur een [korte introductie](#) over CMake.
3. [Bestudeer in het kort de eigenschappen van Clion](#), hoe je snel door de code kunt vliegen (CTRL/CMD + links muis), wat refactoring is en hoe SVN geïntegreerd is.
4. Bestudeer de Cortex-M en architectuur en [de STM32 familie](#). Onderzoek hoe de [CPU \(registers en geheugeninterface\)](#) en de [NVIC interrupt controller](#) zijn opgezet. Verzamel informatie over de verschillende energiemodes waarmee de cpu core werkt. Bestudeer de [TAP Debug controller](#) en beschrijf het gebruik er van. Vat voor je zelf samen wat je bevindingen zijn. Kennis van de Cortex-M architectuur is essentieel en zeer belangrijk voor elke ESE student and ingenieur!
5. Bestudeer uit de [STM32F412 Reference Manual](#) of [de STM32L432 Reference Manual](#) het RTC en BKUP peripheral. Na het laden van het project in Clion, ontdek ook hoe de HAL-driver voor de RTC werkt. In Clion, gebruik Shift ... Shift (snel) en typ HAL_RTC om een aantal HAL-functies te zien die werken met de HAL RTC-bibliotheek.
6. De aangeleverde projectdirectory "STM32" bevat de directories "*Practicum_STM32L432NucleoBoard*" en "*Practicum_STM32F412DiscoveryBoard*". Elk van deze directories is de hoofdpracticum directory. Deze directory bevat een CMakeLists.txt bestand. Dit bestand moet als projectbestand in CLion worden ingeladen.



7. Zorg dat in dit RGT.cmake bestand taal gekozen is (Nederlands of Engels) en de Debugopdracht actief gemaakt is (zonder # er voor).



```
1  #/*****
2  # Basis CMake script voor het RGT practicum
3  # Copyright 2004-2020 ir drs E.J Boks
4  # Opleiding Embedded Systems Engineering
5  # Hogeschool van Arnhem en Nijmegen
6  # $Id: RGT.cmake 3475 2019-04-17 13:22:02Z ewout $
7  #*****/
8
9
10 ## BELANGRIJK ..... IMPORTANT .....
11 ## Zet hier de gewenste taal :
12 ## Select the preferred language :
13 set(InterfaceTaal Nederlands CACHE STRING "Kies voor het Nederlandstalige practicum") # Voor Nederlands
14 #set(InterfaceTaal English CACHE STRING "Perform the practical work in English" ) # For English
15
16 ## BELANGRIJK ..... IMPORTANT .....
17 ## Zet hier het practicum dat wordt uitgevoerd.
18 # This is the Debug Assignment. Uncomment for debug session 1-2
19 set(practicumopdracht DebugOpdracht CACHE STRING "De debugopdracht voorafgaand aan de ECS opdrachten")
20
21 # This is the Input Response assignment. Uncomment for ECS assignment 1
22 #set(practicumopdracht InputResponsieOpdracht CACHE STRING "Bouw RGT inputresponsies")
23
24 # This is the PID Controller assignment. Uncomment for ECS assignment 2-5
25 #set(practicumopdracht PIDControllerOpdracht CACHE STRING "Bouw een RGT PID controller")
26
27 # Na het instellen van deze variabele is het nodig een CMake Reset Cache uit te voeren :
28 # After setting this variable, a CMake cache reset is required :
29 # https://www.jetbrains.com/help/clion/cmake-cache.html
30
31 if (NOT InterfaceTaal)
32     message(FATAL_ERROR "Geen taal gekozen : zet de voorkeurstaal!\nNo language selected. Please set the preferred language!\n")
33 else(NOT InterfaceTaal)
34     message(STATUS "Taal/Language is : ${InterfaceTaal}")
35     add_definitions(-DInterfaceTaal${InterfaceTaal})
36 endif (NOT InterfaceTaal)
37
38 if(NOT practicumopdracht)
39
40     if(InterfaceTaal STREQUAL "Nederlands")
41         message(FATAL_ERROR "Geen practicum gekozen : zet in dit document het gewenste practicum.")
42     else(InterfaceTaal STREQUAL "Nederlands")
43         message(FATAL_ERROR "Geen assignment selected : please set the assignment in this document.")
44     endif(InterfaceTaal STREQUAL "Nederlands")
45
46 endif(NOT practicumopdracht)
```

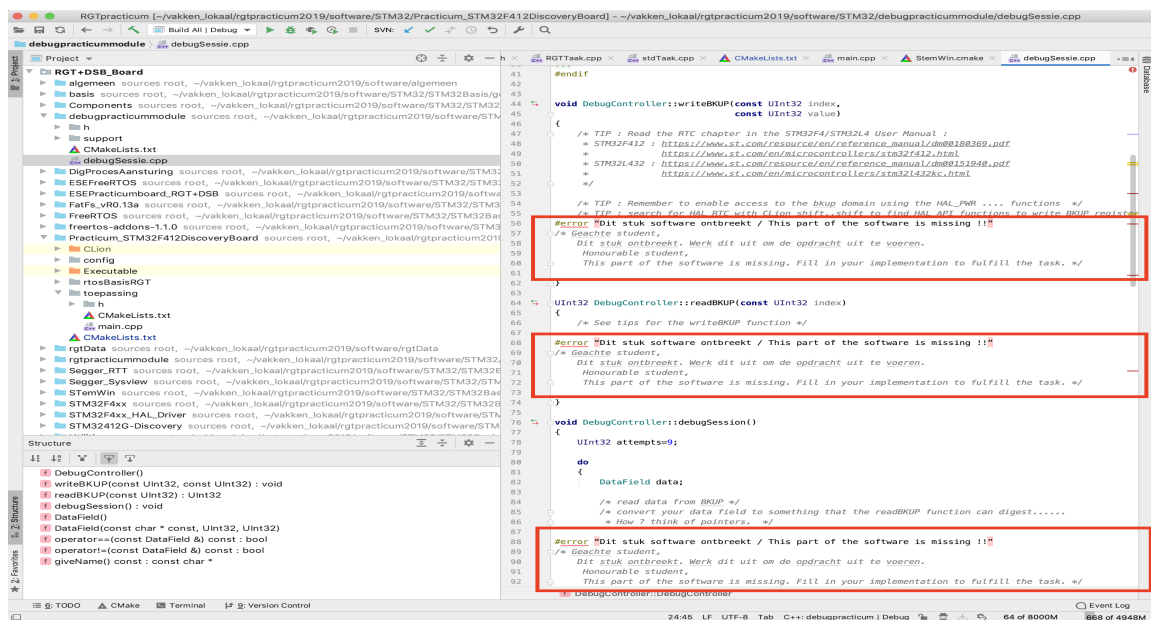
Voer daarna een [CMake Reset Cache](#) door om het project vorm te geven in CLion.

8. CLion laadt het project in en als alles goed gegaan is, verschijnt een "klaar" melding in de tab "CMake" :

10. Bekijk de structuur van het project en zorg er voor dat alle bestanden door CLion herkend worden (“klik op een de top of op een deelproject, rechter muisknop, selecteer dan : “*mark as project sources and headers*”) :

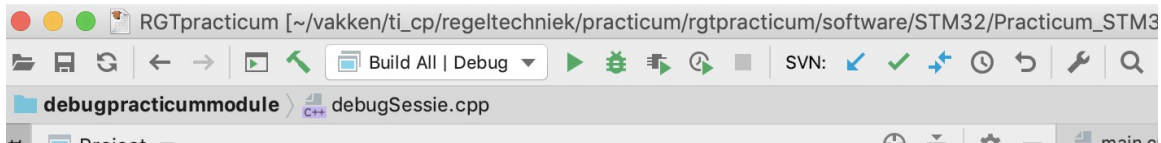
Dit geeft aan CLion aan dat het bestand voor de context analyse van de software door CLion in de context database moet worden opgenomen. CLion weet ALLES van de software die wordt gebouwd, maar moet natuurlijk wel weten welke bestanden meegenomen moeten worden in de analyse.

11. Alle door de student uit te voeren code is door **#error** markers aangegeven. Door de software te compileren komen deze plekken automatisch te voorschijn.

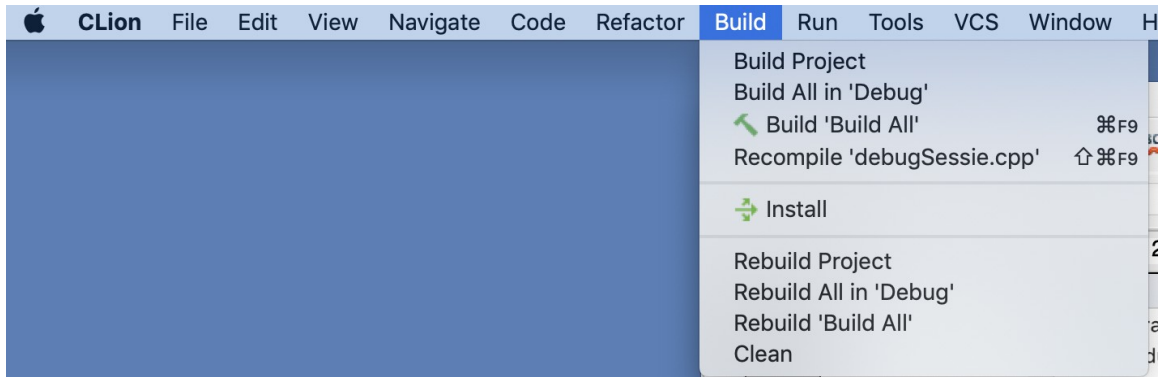


12. Vul in het project de met **#error** gemarkeerde stukken van de driver in voor het gebruiken van de STM32 BKUP registers. De driver moet er voor zorgen dat gegevens van en naar de BKUP registers kunnen worden geschreven.

13. Bouw de software (in debug mode release) met het “hamer symbool” of door het menu build/Build project te kiezen:



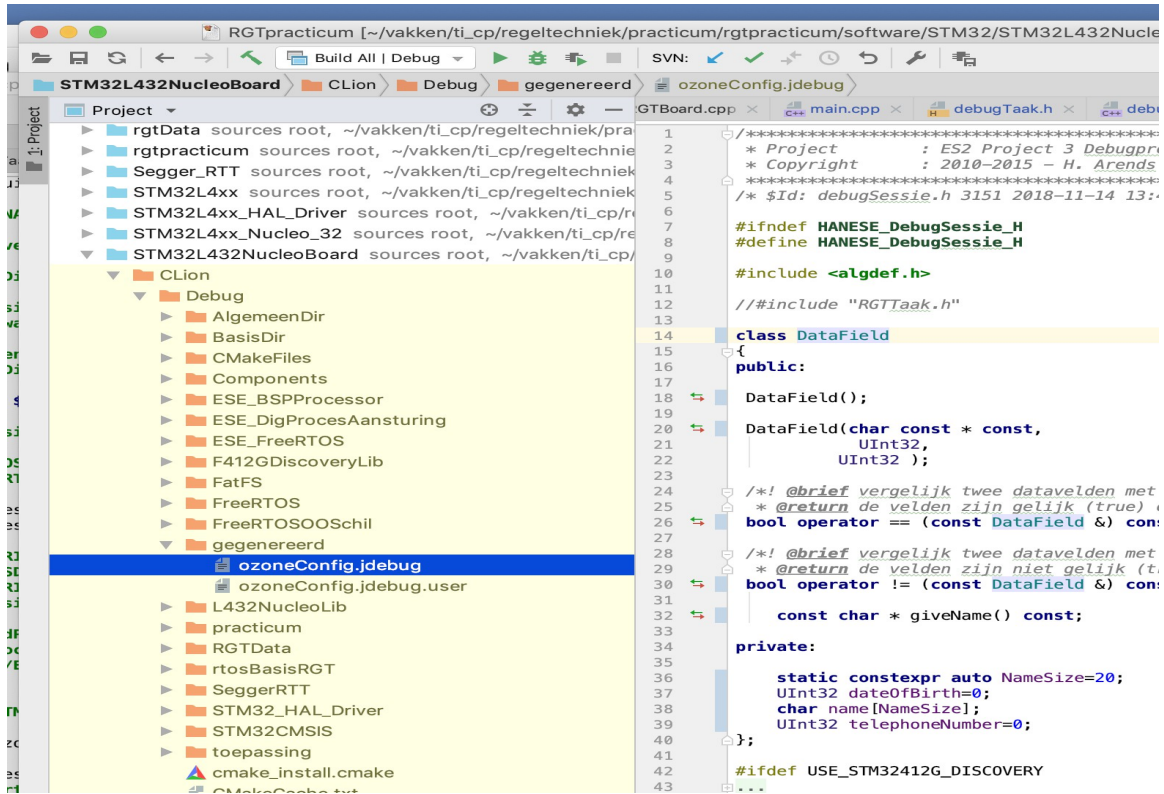
of :



14. Het bouwen van de driver kost wat tijd en vereist behoorlijk wat compilerruns. Wees geduldig, vraag je collega studenten en denk na!
15. Als de binary klaar, dan wordt dit afgebeeld in de compiler output:

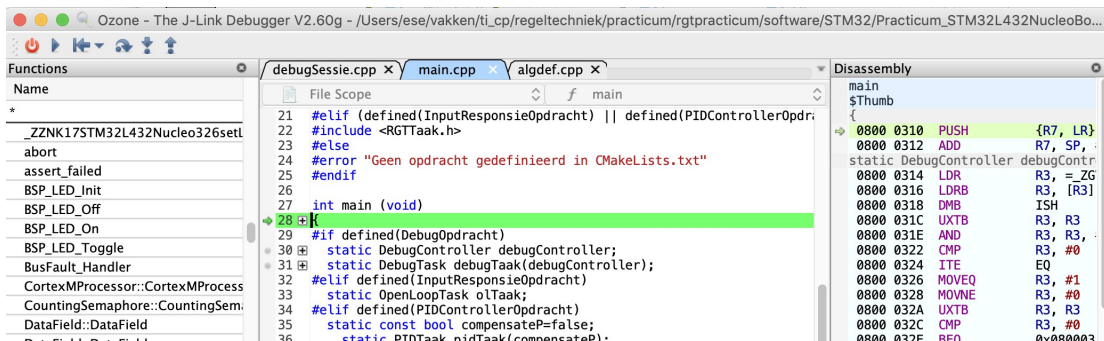
```
[ 97%] Built target STM32L4xxNUCLEO32
[ 97%] Built target debugpracticum
Scanning dependencies of target RGTpracticum.elf
[ 98%] Building CXX object toepassing/CMakeFiles/RGTpracticum.elf.dir/main.cpp.o
[ 99%] Linking CXX executable ../../Executable/Debug/RGTpracticum.elf
[ 99%] Built target RGTpracticum.elf
[100%] Mac OS X post-actie: RGTpracticum.map naar outputdir gekopieerd.
STM32 RGTpracticum.elf [DebugOpdracht] is gemaakt.
/Users/ese/vakken/ti_cp/regeltechniek/practicum/rgtpracticum/software/STM32/STM32L432NucleoBoard/Executable/Debug/RGTpracticum.elf :
section      size      addr
.text        0xd7ec   0x8000000
.ARM.exidx   0x408     0x800d7ec
.rodata      0x12ec   0x800dbf4
.data        0x70    0x2000000
.bss         0x4838   0x2000070
._usrstack   0x100     0x20048a8
.comment     0x7f      0x0
```


16. Het is nu tijd om de gemaakte binary in de target te laden. We gebruiken hier voor de door Segger aangeleverde [Ozone debugger](#). We kunnen Ozone gebruiken, omdat de [ST-Link debug interface vervangen is door een Segger Jlink interface](#). Open in de Ozone Debugger het voorbereide debugger bestand ozoneConfig.jdebug dat in de Clion gegenereerde directory te vinden is (meestal cmake-build-debug/gegenereerd or Clion/Debug/gegenereerd) :



Open dit in Ozone, en maak daarna verbinding met de target (USB kabel moet wel geplaatst zijn !).

17. Als het project goed ingeladen is, verschijnt de software in het middenvenster:



Voor meer informatie over Ozone's mogelijkheden, [bekijk deze video](#).

18. De debugger wordt gestart door de linkerboven groene knop te starten.
19. Dan zal de software in de target worden geladen en debugger blijft met de uitvoering van de software in de eerste regel van main() staan:
20. Test de driver door met “Single step” en met breakpoints door de code heen te stappen. Als er een fout optreedt, analyseer deze dan met behulp van variabele en register inspectie ga terug naar CLion en corrigeer de fout ... compileer ga terug naar Ozone en laad de binary opnieuw in enzovoort tot controller doet wat er opgedragen is..
21. Start de RTT monitor op vanuit het Segger menu in de Windows startknop en monitor de RTT output. Klopt ht inderdaad wat er gebeurt?

Inleveren

Toon opdracht 16-20 werkend aan de docent tijdens het practicum of tijdens de volgende practicumssessie om voor deze opdracht een voldoende te krijgen.