

Open-CMSIS-Pack

Technical Project Meeting 2022-02-22

This meeting is recorded !



Agenda

- Welcome
- Updates from past week
- Generator Support in 'csolution' (demo)
- Review ToDo list
- Wrap Up

Updates from past week

- [csolution] Command to delete intermediate files ([#143](#))
- [csolution] Command to list boards ([#141](#))
- [CMSIS-Toolbox] tool config data ([#142](#))
 - Configuration file in installation folder instead of environment variables (CMSIS_[PACK|BUILD|COMPILER]_ROOT)
 - Merge tool: ([#104](#))
 - merge.tool =
 - mergetool.<tool>.path =
 - mergetool.<tool>.cmd =
 - BASE, LOCAL, REMOTE, MERGED
 - mergetool.keepBackup
- Conflicting files should be described in mutual exclusive way (e.g. linker files) [Spec Issue 89](#)
- Active/Default context [Spec Issue 86](#)
 - First listed Build+Target Type

'csolution' Generator Demo

- Fred provided a variant of STM32F4xx_DFP running STM32CubeMX 'headlessly'
 - Requires local STM32Cube.ioc file
 - '*list dependencies*' detects missing generator files '*.gpdsc'
 - '*list generators*' lists selected generators
 - '*run -g <generator>*' launches generator
 - Writes a <context>.generate.yml file prior to launch
 - '*list contexts*' lists all contexts of the solution
 - Context format: <project>.<build>+<target>
 - Proposal:
execute all commands in a 'context' instead of specifying a 'project'
- => We need a couple more days before releasing another devdrop

CMSIS Project Manager – prioritized ToDo list

1. Review of CMSIS-Toolbox by partners with feedback on potential issues (ongoing)
 - Discuss - Get rid of project name usage as much as possible [issue #81](#)
2. PLM of config files (proposal is included in specification)
 - Specify a preferred 3-way merge utility? E.g. <https://git-scm.com/docs/git-merge-file> (issues #104)
3. Multiple Component Instances ([issues #76](#))
4. RTE directory structure finalize (proposal is included in specification) ([issues #77](#))
5. Local copy of packs (ST requirement) – duplicate of #6?
6. Handling of packs (CMSIS-Pack-Root, Repositories, Local packs, etc.) ([issues #85](#))
 - Finalize definition cdefault.yml
7. CMSIS-Zone integration (resource management) ([issue #87](#))
 - support templating languages and use it with generators ([issue #78](#))
8. Execution Groups (or perhaps better execution phases) ([issue #88](#))
9. Generator support / extensions
- NEW** 10. Linker Scatter file handling
11. Layer interface definitions
12. Board conditions
13. Layers distributed in packs
14. Vendor specific additions (ST requirement) [devtools #83](#)
15. Multiple devices defined by a board

Wrap Up

- Reminder: CMSIS Annual Update Webinar
 - March 15th, 2022 @ 16.30 CET (15:30 UK)
 - Please confirm your attendance in advance by email to cmsis@arm.com.
 - No Open-CMSIS-Pack Technical Meeting!
- Action all:
 - Provide review feedback
 - Suggest topics for next week's agenda
- Next Open-CMSIS-Pack meeting: 1st Mar. 2022 @ 16:00 CET (15:00 UK)

Thank you



'csolution' updates

- Multi-Core identifier (missing Pname)

```
YAML: device: LPC55S69JEV98:cm33_core0
```

```
CPRJ: <target Dname="LPC55S69JEV98" Pname="cm33_core0" \>
```

- Linker Script files from project vs components

Linker scripts can be explicitly added into YAML files as normal file entries:

```
groups:  
- group: Sources  
  files:  
    - file: ./source/LPC55S69_cm33_core0_flash_s.scf
```

But they can also be indirectly provided by components, for example:

```
<component Cclass="Device" Cgroup="Startup" Cvendor="NXP"  
  <file attr="config" category="linkerScript" name="arm/LPC55S69_cm33_core0_flash.scf" />  
</component>
```

Shall the user-provided linker scripts have precedence over the component-provided ones?

- Generator support

New commands under implementation enable the use of generators according to the current specification:

https://open-cmsis-pack.github.io/Open-CMSIS-Pack-Spec/main/html/pdsc_generators_pg.html

```
list generators --solution <example.csolution.yml> [--context <project[.build-type][+target-type]>]
```

```
run --solution <example.csolution.yml> [--context <project[.build-type][+target-type]>] --generator <id>
```

Read generated info from GPDSC and check further component dependencies.

- Device discovering from selected board

Feature under implementation. When a **board** is specified the **device** specification can be omitted.

CMSIS Annual Update Meeting Virtual / Webinar

Date: March 15th, 2022

Time: 4.30 pm CET/11.30 am EDT/10.30 am CDT/8.30 am PDT

Duration: 1.5 hours including time for Q&A.

Agenda:

- CMSIS Overview and Market Adoption (www.arm.com/cmsis)
- Open-CMSIS-Pack: infrastructure to manage software components and improve code reuse (www.open-cmsis-pack.org)
 - Achievements since start of open governance project
 - Development tools and libraries for integration into VS Code and other IDEs
 - Specification updates and Roadmap
- Open-CMSIS-CDI: Common Device Interface for IoT and ML applications
 - Complements the Open-CMSIS-Pack project with common API interfaces
 - Starts with existing API definitions, documentation, and validation tests
 - Ramping up partner engagement
- CMSIS-DAP v2.1 – Update on Firmware for CoreSight Debug Access Protocol
 - Automated board discovery based on Open-CMSIS-Pack
 - Supports both browser-based and desktop debugging
 - Roadmap to Event Recorder support
- CMSIS-DSP/NN – Update on Software Library for DSP and Neuronal Networks
 - Introducing new compute kernels and Python support

Please confirm your attendance in advance by email to cmsis@arm.com.