Building Own Application for R-Car D3 Custom Integrity

Generate Makefiles with CMake

For general information about CGI Studio Build System and CMake properties available to configure Candera applications refer to CGI Studio Build System.

To configure and generate makefiles for compiling the Player, follow below steps:

1. Point CMake to source directory
   - Folder containing top-level "CMakeLists.txt", e.g. \cmake\Candera\Player
2. Point CMake to binary directory
   - May not exist, create folder with CMake if necessary
   - Root directory for generated makefiles / solutions
   - Root directory for build artifacts
3. Configure
   - Specify MinGW Makefiles as generator for the project
   - Specify tool-chain file for cross-compiling
4. Select the toolchain-file from the delivery (adapt for your toolchain installation and application specific needs)
   - \cgi_studio_devices\src\RCarD3\ToolchainFiles\MULTI-toolchain-RCarD3_custom.txt
5. Adapt the CMake settings if necessary, e.g.:
   - CMAKE_BUILD_TYPE-> Debug or Release
   - Press Configure
6. Press Generate

Building Application

Via the CMake variable CGIDEVICE_BSP_CFG_NAME, it is possible to choose the BSP library configuration. By default this is set to release (rel), however, it is also possible to choose debug (dbg), checked (chk) resp. coverage (cov).

On the custom board display id 1 was used for all our. Please make sure to have set display id to 1 in Scene Composer. If you want to change this, this also has to be changed for our samples in CMake via the variable CGIDEVICE_DISPLAY_ID.

For this board it is intended to use a shader compiler. By default, the 2D shaders are pre-compiled and already linked to the sample. This is done via the CMake variables CANDERA_EXTERNAL_SHADER_BUILDER_HEADER_INCLUDE, CANDERA_EXTERNAL_SHADER_BUILDER_HEADER_PATH and CANDERA_EXTERNAL_SHADER_BUILDER_SOURCE_PATH. If you want to change this, first uncheck RCARD3_EXTERNAL_SHADER_BUILDER_USAGE_DEFAULT, then it is possible to e.g. delete the value of the above mentioned variables in order to not use pre-compiled shaders. For all the other shaders, the compilation is done via plugin in CGI Studio. Please make sure to have the RCarD3 Shader Compiler Plugin (can be found under File/Preferences) set to “Binary”. Additionally, set the absolute path to the shader compiler *.exe. If the RCarD3 Shader Compiler Plugin is set to “Text”, the shader compiler is not used at all.

It is possible to have the asset stored in the eMMC resp. QSPI flash instead of having it linked to the monolith. Use the writing tool to store the asset into the flash. Make also sure to copy the flash wrapper layer to "cgi_studio_3psw\internal\Devices\RCarD3\storage_driver_wrapper_layer" where it is expected to be when using eMMC resp. QSPI as asset repository. In CMake it is required to change the variable RCARD3_ASSET_LOCATION from “Monolith” to “eMMC” resp. “QSPI”. Additionally, it is necessary to verify the path to the Flash tool (RCARD3_FLASH_TOOL_PATH) in order to link the correct libraries.

Revision #4
Created 2023-03-02 01:05:39 UTC by Tsuyoshi.Kato
Updated 2024-10-07 02:59:14 UTC by Tsuyoshi.Kato