Traveo2 BSP

Traveo2 NoOS BSP: Prerequisites

Hardware: Infineon Traveo II

CGI Studio has been tested on **Infineon Traveo II (CYTVII-C-2D-6M-500-BGA CPU REV-B) board and Infineon Traveo II (CYTVII-C-2D-4M-216 CPU REV-A) board** .

CMake

In order to build Candera, the Meta build system CMake needs to be installed with at least version mentioned here.
CMake makes it possible to create the cross compilation makefiles that can be used with Green Hills MULTI compiler (or IAR Embedded Workbench for ARM).

Generator

To allow cross compilation with makefiles, MinGW 4.2.1 was used for this release.

Toolchain

The build tool chain is not part of the delivery.

Green Hills

Green Hills MULTI IDE, Green Hills compiler and the corresponding Traveo II graphics driver have to be obtained directly from Green Hills and/or Infineon and should be installed according to their installation guide.
CGI Studio was tested with **MULTI IDE 7.1.4 (configured as suggested by Infineon - GHS Traveo II Debug Script Environment), Compiler 2017.1.4**

The Green Hills probe 5.6.4 for ARM was used for connection/debug on the target.

Embedded Workbench

IAR EW IDE for ARM, compiler and the corresponding Traveo II graphics driver have to be obtained directly from IAR and/or Infineon and should be installed according to their installation guide.

CGI Studio was tested with **IAR EW IDE 8.22.2 (configured as suggested by Infineon), Compiler IAR ANSI C/C++ Compiler 8.22.2.15995/W32 for ARM**

The IAR Systems I-jet probe 2.5.0.0 was used for connection/debug on the target.

Infineon Graphics Library

The graphics driver is not part of this release. CGI Studio was integrated and tested with the following versions:

- V1.2.0 for IAR EW
- V2.3.0 for GreenHills

Flashing tool

Infineon Auto Flash Utility 1.2 via MiniProg4 was used in order to flash asset file to the external flash (Hyperflash).

Traveo2 AUTOSAR BSP: Prerequisites

Hardware: Infineon Traveo II

CGI Studio has been tested on **Infineon Traveo II (custom 4M) board** .

CMake

In order to build Candera, the Meta build system CMake needs to be installed with at least version mentioned here.
CMake makes it possible to create the cross compilation makefiles that can be used with Green Hills MULTI compiler.

Generator

To allow cross compilation with makefiles, MinGW 4.2.1 was used for this release.

Toolchain

The build tool chain is not part of the delivery.

Green Hills

Green Hills MULTI IDE, Green Hills compiler and the corresponding Traveo II graphics driver have to be obtained directly from Green Hills and/or Infineon and should be installed according to their installation guide.
CGI Studio was tested with **MULTI IDE 7.1.4 (configured as suggested by Infineon - GHS Traveo II Debug Script Environment), Compiler 2017.1.4**

The Green Hills probe 5.6.4 for ARM was used for connection/debug on the target.

AUTOSAR

Autosar environment is not part of this release. The Autosar environment is provided by customer.

Infineon Graphics Library

The graphics driver is not part of this release. CGI Studio was integrated and tested with the libraries received from customer (V1e0.0)

BSP External memory initialization limitation

Please ensure that the external peripheral on the target device is supported by the BSP initialization routine for the external memory: **Cy_GfxEnv_FastInit(bUseSmif0, bUseSmif1, xxx)**
The support of external memory components can differ between BSP/Graphic driver versions!

In case of unsupported external memory initialization in the specific driver/BSP version compared to the Traveo II evaluation board the value will be set to 0 (disabled) :Cy_GfxEnv_FastInit(**0, 0**, xxx) (see corresponding initialization in \cgi_studio_devices\src\Traveo2\CourierPlatformDefs\Target_NoOs_ARM\KernelTargets\main.c) .
In this case, only the internal RAM and FLASH can be used on the target board.

Running the CGI Player Demo Application on Traveo2

Download the Application

To download and debug the application you need a Green Hills probe (or IAR Systems I-jet probe) that is configured accordingly for your Infineon Traveo II board.

Executable Application

Green Hills

- In MULTI launcher select "Components" -> "Open debugger" and choose the application from your build folder with extension .ghsmc.
- Then press "D" (for "Detect"), once the script detects the board and sets all needed variables in the debugger window three new symbols will be available "F" "V" and "E".
- Press "F" (for "Flashing") the application on the board.
- Press "F5" on keyboard or "Go on Selected Items" ("Play") button to run the application.

> "surface_blit_display_basic" graphics driver sample’s project has to be set as WorkingDir (Workspace) in MULTI launcher.

IAR Embedded Workbench

Binary directory contains two IAR EW projects under "iar_projects" (for flash only and/or flash and debug).

- Open the EW workspace *.eww from the project in which you will select e.g. .eww.
- Rebuild ("Project" ->"Rebuild All") cm0plus project.
- Press ("Project" ->"Download and Debug") to flash the executable.
- Press "F5" on keyboard or "Debug -> Go" to run the application.

Loading the asset

The CGI Player demo application uses an asset file which could be added by the linker or loaded from external flash. CGIDEVICE_ASSET_LOCATION holds that information (options: Monolith or Flash).

In case of Monolith option for CGIDEVICE_ASSET_LOCATION, the asset file will be loaded to internal flash together with the application. You have to select the asset file in CMake via the variable ASSET_TARGET_LOCATION before you build the application.

In case of Flash option for CGIDEVICE_ASSET_LOCATION, the asset file will be loaded during startup of the application. A prerequisite is that it must be flashed to the external flash (Hyperflash) on the board before executing the app.

> This is the only available option for IAR Embedded Workbench toolchain.

Running the CGI CourierSampleAppLight Demo Application on Traveo2

Download the Application

To download and debug the application you need a Green Hills probe (or IAR Systems I-jet probe) that is configured accordingly for your Infineon Traveo II board.

Executable Application

Green Hills

- In MULTI launcher select "Components" -> "Open debugger" and choose the application from your build folder with extension .ghsmc.
- Then press "D" (for "Detect"), once the script detects the board and sets all needed variables in the debugger window three new symbols will be available "F" "V" and "E".
- Press "F" (for "Flashing") the application on the board.
- Press "F5" on keyboard or "Go on Selected Items" ("Play") button to run the application.

> "surface_blit_display_basic" graphics driver sample’s project has to be set as WorkingDir (Workspace) in MULTI launcher.

IAR Embedded Workbench

Binary directory contains two IAR EW projects under "iar_projects" (for flash only and/or flash and debug).

- Open the EW workspace *.eww from the project which you will select e.g. .eww.
- Rebuild ("Project" ->"Rebuild All") cm0plus project. Press ("Project" ->"Download and Debug") to flash the executable.
- Press "F5" on keyboard or "Debug -> Go" to run the application.

Loading the asset

The CGI CourierSampleAppLight demo application uses an asset file which could be added by the linker or loaded from external flash. CGIDEVICE_ASSET_LOCATION holds that information (options:

Monolith or Flash).

In case of Monolith option for CGIDEVICE_ASSET_LOCATION, the asset file will be loaded to internal flash together with the application. You have to copy the asset file in the directory and name it as **“AssetLibCff_Traveo2_Target.bin”** before you build the application.

In case of Flash option for CGIDEVICE_ASSET_LOCATION, the asset file will be loaded during startup of the application. A prerequisite is that it must be flashed to the external flash (Hyperflash) on the board before executing the app.

> This is the only available option for IAR Embedded Workbench toolchain.

Building Own Application for Traveo2

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 CGI CourierSampleAppLight, follow the steps below:

1. Point CMake to source directory
   - Folder containing top-level "CMakeLists.txt", e.g. or \ LightPlayer
2. Point CMake to build 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 (or Ninja) as generator for the project
   - Specify tool-chain file for cross-compiling
4. Select the toolchain-file from the delivery based on your target. (adapt for your toolchain installation and application specific needs)
   - (for MULTI) cgi_studio_devices\src\Traveo2\ToolchainFiles\MULTI-toolchain-TraveoII_C_6M.txt or cgi_studio_devices\src\Traveo2\ToolchainFiles\MULTI-toolchain-TraveoII_C_4M.txt
   - (for IAR EW) cgi_studio_devices\src\Traveo2\ToolchainFiles\IAR-toolchain-TraveoII_C.txt)
5. Adapt the CMake settings if necessary, e.g.:
   - CMAKE_BUILD_TYPE-> Debug or Release
   - Press Configure
6. Press Generate

Building Application

After generating your cmake files, in Command Prompt simply navigate to the specified build directory and type "mingw32-make" (or "ninja") to start the build.

The output executable file can be found in your build directory.

Load and debug the application as described in the previous chapter.

Useful Hints and Recommendations

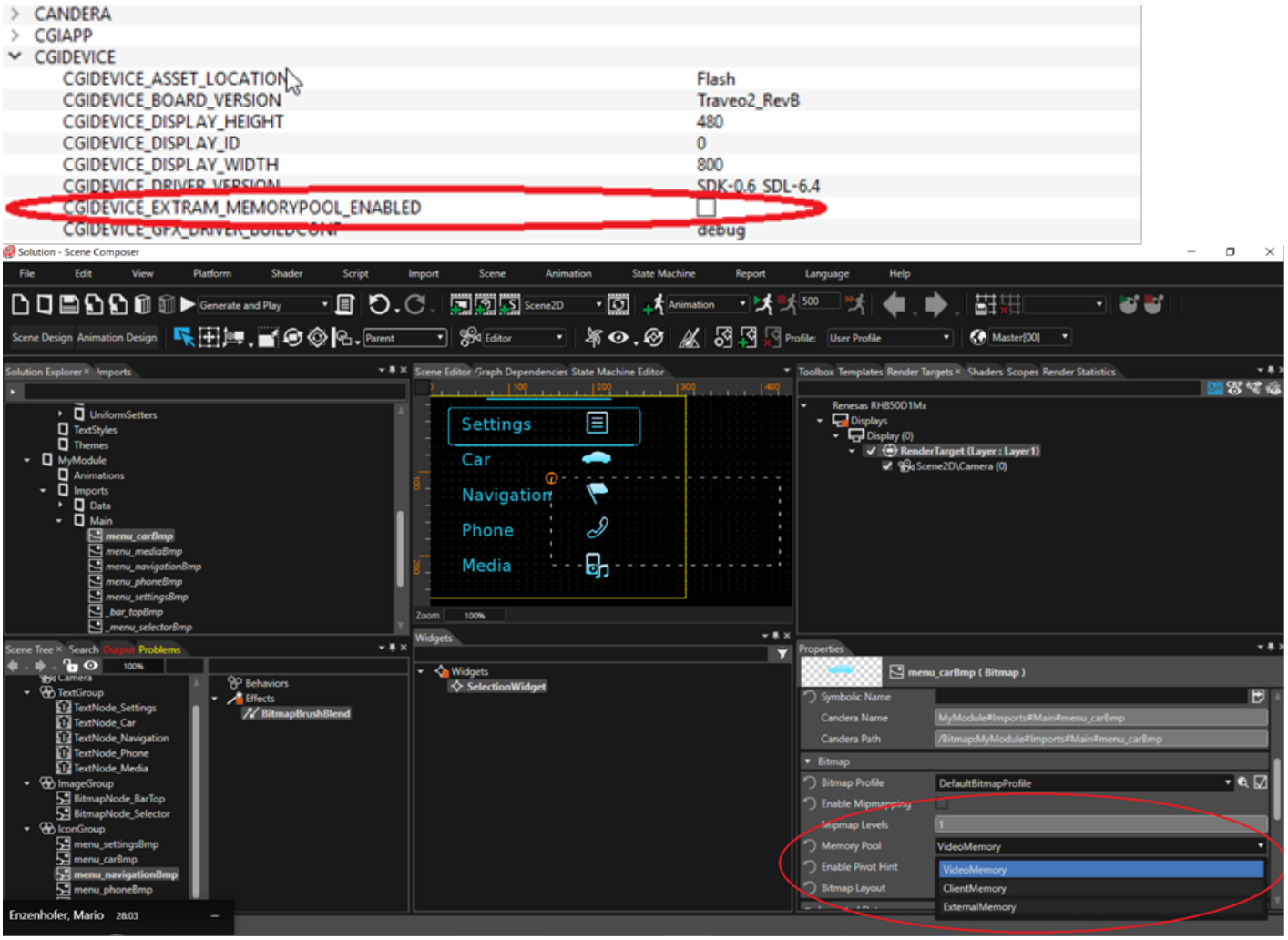
Setting for tasks

In CMake you have the possibility to set the size and the location of the memory for the command queues and also the COPSes reserved for that task.
Additionally, you can set the memory location for the frame buffers (in case they exist in the solution) for the layers.
Disable unused tasks by setting the size to 0. (This has to be in correlation with the solution which you want to execute on the target)

CGIDEVICE_INSTR_Q1_MEMLOC_TASK_WIN_PRIO_1	VramPool
CGIDEVICE_INSTR_Q1_SIZE_TASK_WIN_PRIO_1	65536
CGIDEVICE_INSTR_Q1_TASK_COPSES_WIN_PRIO_1	64
CGIDEVICE_INSTR_Q2_MEMLOC_TASK_WIN_PRIO_2	VramPool
CGIDEVICE_INSTR_Q2_SIZE_TASK_WIN_PRIO_2	65536
CGIDEVICE_INSTR_Q2_TASK_COPSES_WIN_PRIO_2	0
CGIDEVICE_INSTR_Q3_MEMLOC_TASK_WIN_PRIO_3	VramPool
CGIDEVICE_INSTR_Q3_SIZE_TASK_WIN_PRIO_3	65536
CGIDEVICE_INSTR_Q3_TASK_COPSES_WIN_PRIO_3	0
CGIDEVICE_INSTR_Q4_MEMLOC_TASK_MEM_PRIO_0	VramPool
CGIDEVICE_INSTR_Q4_SIZE_TASK_MEM_PRIO_0	65536
CGIDEVICE_INSTR_Q4_TASK_COPSES_MEM_PRIO_0	0
CGIDEVICE_INSTR_Q5_MEMLOC_TASK_MEM_PRIO_1	VramPool
CGIDEVICE_INSTR_Q5_SIZE_TASK_MEM_PRIO_1	65536
CGIDEVICE_INSTR_Q5_TASK_COPSES_MEM_PRIO_1	64
CGIDEVICE_INSTR_Q6_MEMLOC_TASK_MEM_PRIO_2	VramPool
CGIDEVICE_INSTR_Q6_SIZE_TASK_MEM_PRIO_2	65536
CGIDEVICE_INSTR_Q6_TASK_COPSES_MEM_PRIO_2	0
CGIDEVICE_LAYER0_MEMLOC	VramPool
CGIDEVICE_LAYER1_MEMLOC	VramPool
CGIDEVICE_LAYER2_MEMLOC	VramPool
CGIDEVICE_LAYER3_MEMLOC	VramPool
CGIDEVICE_LAYER4_MEMLOC	VramPool

EXTRAM support

By default, all the frame buffers for the layers, the instruction buffers, and the bitmaps (if they are not stored in Client memory) are stored in the VRAM.
However, it is possible to configure the memory location to EXTRAM (HyperRAM) instead. The memory location of the frame buffers and instruction buffers is configured via CMake (see Figure 2).
The memory location of the bitmaps is selected via Scene Composer (refer to Figure 3).



For target build, please keep in mind to check (enable) CGIDEVICE_EXTRAM_MEMORYPOOL_ENABLED in CMake in case you want to store images in external RAM.
Without that check it is not possible to allocate memory in the HyperRAM and the VRAM will be used instead.

Limitations

Each OTF layer (window) must use different TaskPrio ID.

OTF window surface height is minimum two slices (16 lines) and maximum (2 * window height - 1).

Maximum width for LBO blit area: 1600 (Maximum width for OTF layer (window) is 1600)

Supported values for OTF lines: 16, 32, 64, 128, 256

> Supported CGI Studio samples for this target are:
> LightPlayer
> Empty

- 2DBasic
- 2DEffects
- AdvancedCluster
- Layouter
- Transition

CourierSampleAppLight

- Default CSALight’s solution
- New Traveo2Sample (covers few Traveo2 specific features)

Traveo 2 Sample

Summary

This solution is a sample of a cluster that can be executed on Traveo II and is meant to showcase some of Traveo II's new functionality.

Purpose

This solution allows you to see how various Traveo II features work in conjunction to form a cluster sample.

Description

The sample uses following features:

- Blending
- Rotation
- Blitting
- Scaling
- Decompression
- Line-based (IBO) rendering
- Frame-based (LBO) rendering
- On-the-fly (OTF) rendering