iMX6 Linux ARM

- iMX6 Linux ARM: Prerequisites
- Running the Player Demo Application on iMX6 Linux
- Building Own Application for iMX6 Linux

iMX6 Linux ARM: Prerequisites

Hardware: Freescale iMX6 SABRE

CGI Studio has been tested on Freescale iMX6 SABRE Lite and Automotive Infotainment

Linux Board Support Package (BSP)

The Freescale Release BSP imx-3.14.38-6QP_ga was used for SABRE Automotive Infotainment. For SABRE Lite and commercial projects the FSL Community BSP is recommended. For integration the BSP based on the fido branch (poky 1.8.1) was used (see https://github.com/Freescale/fsl-community-bsp-platform).

Vivante Graphics Driver for Linux

CGI Studio was integrated and tested on top of the drivers included in the above mentioned BSPs (gpu-viv version 5.0.11)

CMake

In order to generate makefiles for building Candera applications, the Meta build system CMake needs to be installed with at least version 3.5.2.

Known Issues

Intitialising vectors in shaders like, e.g. vec3 xyz = new vec3(0.0) in some cases leads to unexpected behaviour. Workaround: initialise with vec3 xyz = new vec3(0.0, 0.0, 0.0);

Running the Player Demo Application on iMX6 Linux

iMX6 System Preparation

For instructions how to install the toolchain please refer to the Yocto Project's Quickstart and Application Developer's Guide.

The application and asset can be copied into the folder /root on the SD-Card with the target root file system, and executed like this:

```
# Connect TeraTerm to COM1 at
# bits per second: 115200
# data bits: 8
# parity: none
# stop bits: 1
# flow control: none
# load the GPU Driver
modprobe galcore.ko
cd /root
# run CGIApplication
./CGIApplication_ARM_Linux_iMX6_Target_GCC <AssetLibrary>
```

Both applications use a target asset file.

Additional Steps for Frame Buffer Blending

If two display planes shall be blended on the iMX6 target, using Candera with the frame-bufffer implementation of the iMX6 drivers, following code can be used to resize and unblank the overlay plane. This is not avaliable if X11 is used.

```
# Use fbset to find out settings of /dev/fb0
fbset -fb /dev/fb0
# Example output
#mode "1920x1080-60"
#    # D: 148.500 MHz, H: 67.500 kHz, V: 60.000 Hz
#    geometry 1920 1080 1920 1080 16
#    timings 6734 148 88 36 4 44 5
#    hsync high
#    vsync high
#    rgba 5/11,6/5,5/0,0/0
#endmode
# Set /dev/fb1 to the same settings as /dev/fb0
```

```
fbset -fb /dev/fb1 -g 1920 1080 1920 1080 16
fbset -fb /dev/fb1 -t 6734 148 88 36 4 44 5
fbset -fb /dev/fb1 -hsync high
fbset -fb /dev/fb1 -vsync high
# Unblank /dev/fb1
echo 0 > /sys/class/graphics/fb1/blank
```

Building Own Application for iMX6 Linux

Compile Tool Chain Preparations

For instructions how to install the toolchain please refer to the Yocto Project's Quickstart and Application Developer's Guide.

To add the header and lib directories to your path, you might have to call a script to set your environment variables:

> This step is needed when you use our build-setup description.

```
cd /opt/poky/x.y.z (where x.y.z is your BSP version)
sudo chmod +x environment-setup-cortexa9hf-vfp-neon-poky-linux-gnueabi
source environment-setup-cortexa9hf-vfp-neon-poky-linux-gnueabi
```

> Should be done every new build (environment variables are valid per terminal session)

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.

CMAKE-GUI has to be run on same terminal after source command (step 1)

To configure and generate makefiles for compiling the Player for IMX6 Linux, 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 Unix Makefiles as generator for the project

- Specify toolchain file for cross-compiling
- Select the toolchain file \cgi_studio_candera\cmake\toolchain-files\GCC-toolchain-iMX6-Linux-yocto.txt
- Press Finish
- Afterwards “Configure” has to be pressed as often as there is still some entry red or the necessary configuration (e.g. CANDERA_TEXTSHAPER = harfbuzz) is done.

4. Press Generate

If the package supports "harfbuzz" as CANDERA_TEXTSHAPER please set this entry appropriately because by default it is set to "ComplexLib". Resume the configuration by pressing “Configure” again.

> Configure COURIER_PLATFORM and COURIER_PLATFORM_CONFIG parameters

Building Application

After generating your CMake files, simply navigate to the specified build directory and type "make" with the terminal or console to start the build.

The output file can be found in your build directory's subfolder CGIApplication.

Copy the application to a dedicated folder in the NFS root and start it as described in the previous chapter.