Fbdev BSP

- Fbdev: Prerequisites
- Running the CGI Player Demo Application on Fbdev
- Building Own Application for Fbdev

Fbdev: Prerequisites

Hardware: Telechips Dolphin 5

CGI Studio has been tested on Telechips Dolphin 5 Board.

CMake

In order to build Candera, the Meta build system CMake needs to be installed with at least version mentioned here.
CMake is able to create the cross compilation make files that can be used with GCC (cross-compile) compiler.

Linux Toolchain (BSP)

- poky-telechips-systemd 4.0.17
- Image and SDK were prepared by the Candera.
- The GCC compiler is not part of the delivery. It is contained in the SDK that has to be provided by the customer.
- Compiler/s used for this release is GCC (GNU 11.2.1).

> SD Card image, SDK and cross-compiler/s are not part of the delivery and have to be provided by the customer.

Hardware: Allwinner T507

CGI Studio has been tested on Allwinner T507 Board.

CMake

In order to build Candera, the Meta build system CMake needs to be installed with at least version mentioned here.
CMake is able to create the cross compilation make files that can be used with GCC (cross-compile) compiler.

Linux Toolchain (BSP)

- To build the Player for the target platform, you need to download the additional library package: OKT507-linux-sdk20.tar.bz2
- The build requires an older version of Debian or Ubuntu. In our case, the entire build was performed on **Ubuntu 18.04.6 LTS** running in WSL on **Windows 11**.

- The GCC compiler is not part of the delivery. It is contained in the SDK that has to be provided by the customer.
- Compiler/s used for this release is GCC 5.3.1 (Linaro GCC 5.3-2016.05).

> SD Card image, SDK and cross-compiler/s are not part of the delivery and have to be provided by the customer.

Running the CGI Player Demo Application on Fbdev

Copy the application and the asset to the target. This can be done by copying the filed to the SD card containing the system, via ssh or with an USB flash drive.

Loading and running the application

Note that no other application that uses graphics hardware can be running when using the Player, the CourierSampleApp or the CitDemoApp.
If your Yocto image is built with Weston or X you must shut these down before running the application.

Player can be run by using the command:

```
$ /PathToPlayer/Player /PathToAsset/AssetName.bin
```

> If touch input handling is enabled, touch devices must be assigned to seat so that display identifier matches seat number. For example, display-0 must be assigned to seat0. Refer to udev documentation for more information on how to assign devices to seats.

Building Own Application for Fbdev

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 Player, follow below steps:

The first step for cross compiling is to make sure all the environment variables for the Yocto SDK are set!

1. Open a terminal.
2. Set the environment. This step is mandatory. Omitting this step will cause building with the host systems compiler instead of cross compiling.
   Dolphin 5 (Yocto based SDK)

   ```
   $source /<path to the Yocto SDK installation>/environment-setup-<platform specific suffix e.g. aarch64-poky-linux>
   ```

   Allwinner T507 (Buildroot based SDK)

   ```
   export SDKTARGETSYSROOT=/home/candera/Desktop/OKT507-linux-sdk20/out/t507/okt507/longan/buildroot/host/aarch64-buildroot-linux-gnu/sysroot
   export PATH=/home/candera/Desktop/OKT507-linux-sdk20/out/t507/okt507/longan/buildroot/host/bin:$PATH
   ```

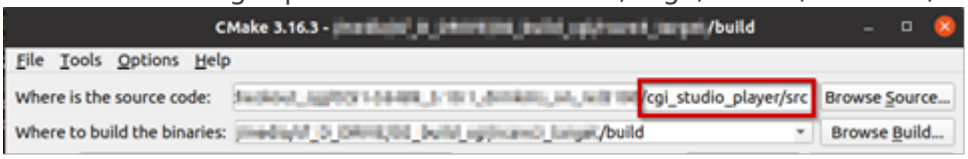
3. Important! From the same console with the set environment start the CMake GUI or run the CMake commandline. We will continue with the CMake GUI in this guide.
   Dolphin 5 (Yocto based SDK)

   ```
   cd <build directory>
   cmake-gui .
   ```

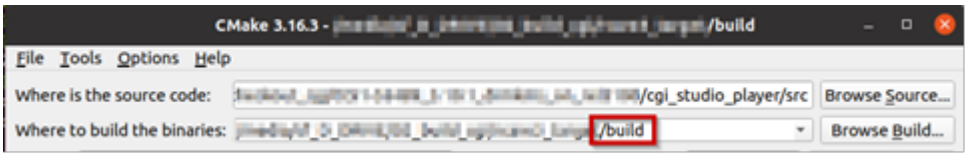
Allwinner T507 (Buildroot based SDK)

```
cd <build directory>
# Environment variables for sdk
export SDKTARGETSYSROOT="/home/candera/Desktop/OKT507-linux-
sdk20/out/t507/okt507/longan/buildroot/host/aarch64-buildroot-linux-gnu/sysroot"
export PATH="/home/candera/Desktop/OKT507-linux-
sdk20/out/t507/okt507/longan/buildroot/host/bin:$PATH"
export SYSROOT="$SDKTARGETSYSROOT"
export SDKSYSROOT="$SYSROOT"
export PKG_CONFIG="/usr/bin/pkg-config"
export PKG_CONFIG_SYSROOT_DIR="$SYSROOT"
export PKG_CONFIG_PATH="$SYSROOT/usr/lib/pkgconfig"
export CXX="/home/candera/Desktop/OKT507-linux-
sdk20/out/t507/okt507/longan/buildroot/host/bin/aarch64-linux-gnu-g++ --
sysroot=$SYSROOT"
export CC="/home/candera/Desktop/OKT507-linux-
sdk20/out/t507/okt507/longan/buildroot/host/bin/aarch64-linux-gnu-gcc --
sysroot=$SYSROOT"
export LD="/home/candera/Desktop/OKT507-linux-
sdk20/out/t507/okt507/longan/buildroot/host/bin/aarch64-linux-gnu-ld --
sysroot=$SYSROOT"
cmake-gui .
```

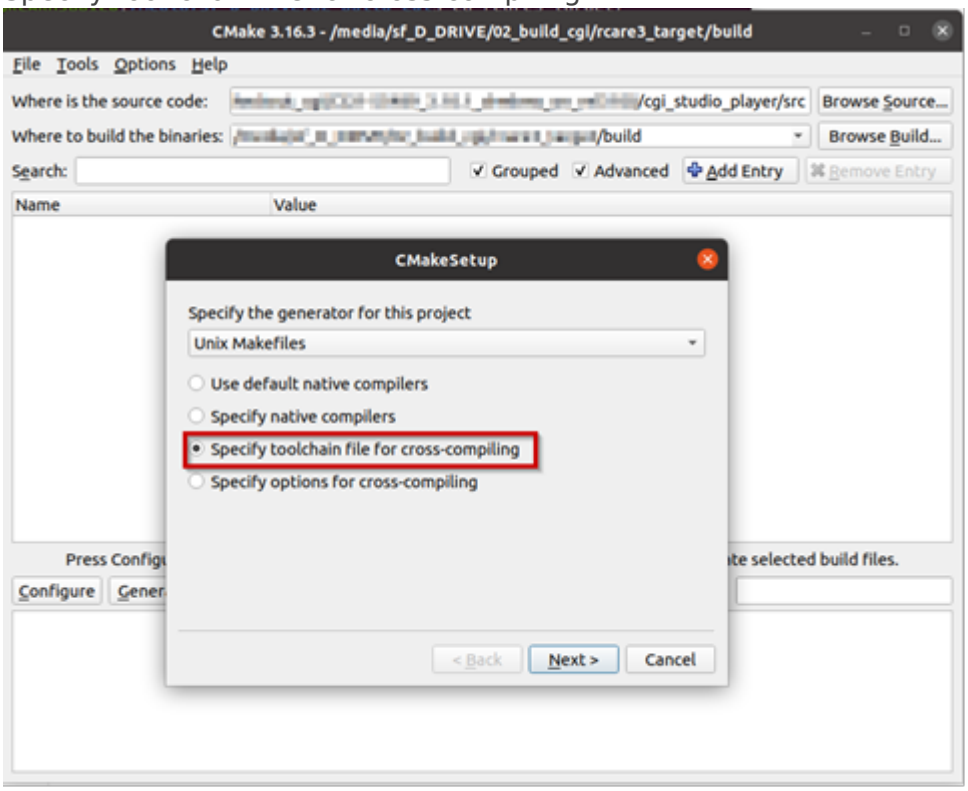
4. Point CMake to source directory
   - Folder containing top-level "CMakeLists.txt", e.g. /cmake/Candera/Player



5. 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



6. Configure
   - Specify Unix Makefiles as generator for the project.
   - Specify tool-chain file for cross-compiling.



7. Select the toolchain-file from the delivery (adapt for your toolchain installation and application specific needs)
   cgi_studio_devices/src/Fbdev/ToolchainFiles/GCC-toolchain-<platform>-Fbdev-Linux.txt
8. Adapt the CMake settings, if necessary, e.g.:
   - CMAKE_BUILD_TYPE-> Debug or Release.
   - Enable/disable features.
   - Press Configure.
9. Press Generate.

Building Application

> Make sure that the environment variables for the Yocto/Buildroot SDK are set in the terminal! Either reuse the terminal used to start CMake for building the Player or call the environment setup script again.

Dolphin 5 (Yocto based SDK)

```
$source /<path to the Yocto SDK installation>/environment-setup-<platform specific suffix e.g. aarch64-poky-linux>
```

Allwinner T507 (Buildroot based SDK)

```
# Environment variables for sdk
export SDKTARGETSYSROOT="/home/candera/Desktop/OKT507-linux-sdk20/out/t507/okt507/longan/buildroot/host/aarch64-buildroot-linux-gnu/sysroot"
export PATH="/home/candera/Desktop/OKT507-linux-sdk20/out/t507/okt507/longan/buildroot/host/bin:$PATH"
export SYSROOT="$SDKTARGETSYSROOT"
export SDKSYSROOT="$SYSROOT"
export PKG_CONFIG="/usr/bin/pkg-config"
export PKG_CONFIG_SYSROOT_DIR="$SYSROOT"
export PKG_CONFIG_PATH="$SYSROOT/usr/lib/pkgconfig"
export CXX="/home/candera/Desktop/OKT507-linux-sdk20/out/t507/okt507/longan/buildroot/host/bin/aarch64-linux-gnu-g++ --sysroot=$SYSROOT"
export CC="/home/candera/Desktop/OKT507-linux-sdk20/out/t507/okt507/longan/buildroot/host/bin/aarch64-linux-gnu-gcc --sysroot=$SYSROOT"
export LD="/home/candera/Desktop/OKT507-linux-sdk20/out/t507/okt507/longan/buildroot/host/bin/aarch64-linux-gnu-ld --sysroot=$SYSROOT"
```

1. Open CMake GUI and configure the project (e.g Player, CourierSampleApp, CitDemoApp) according to Generate Makefiles with CMake.
2. After generating your cmake files, navigate to the specified build directory (binary directory created in CMake GUI) and type "make" to start the build or “make -j” to build with multithreading.
   With some hypervisors, the Linux VM may get unresponsive when using the –j option. Reducing the number of used threads to the number of available cores improves the responsiveness of the VM during the build.

> For CourierSampleApp and CitDemoApp asset file must be created with a matching SCHost.dll as the SCHost.dll for the CourierSampleApp and CitDemoApp contain specialized Widgets.