Target setup for Software Renderer 2D

This is a target setup guide for Software Renderer 2D Platform. it includes guides for all Targets of the Software Renderer 2D Platform

- Target setup guide for the ESP32S3 (ESP32S3-Touch-LCD-2.8C)
- Target Setup for the STM32MP135f and other Linux based targets

Target setup guide for the ESP32S3 (ESP32S3-Touch-LCD-2.8C)

Before you Begin

Before starting with the setup guide, we recommend the following to get familiar with CGI-Studio:

- Explore CGI-Studio

  Read the Quick Start Guide.
- Learn to create SceneComposer Solutions
  The documentation provides instructions on how to create simple SceneComposer solutions. To access SceneComposer navigate to:
  <cgi-studio-installation-path>\bin\SceneComposer\SceneComposer.exe
- Generate Assets for Simulation and Target Environments
  You can find guidance on generating assets from solutions (binary files containing solution information for the player) for both simulation and target environments in the documentation.

With this knowledge you should be fit to complete this guide.

Setup

Template Scenes

When using a template scene as the base for development, the surfaceFormat inside the RenderTarget needs to be set to SurfaceFormatRGB565.

IDE

To compile a CourierSampleApp Example for the target ESP IDF needs to be installed within Visual Studio Code: Getting Started with VS Code IDE - ESP32 - — ESP-IDF Programming Guide v4.3.1 documentation

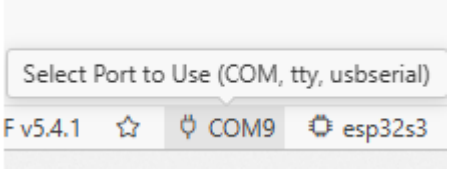
Device Connection to host PC

The ESP32S3-Touch-LCD-2.8C target from waveshare has two USB-C ports. For flashing connect the one labled "UART" to the host PC.

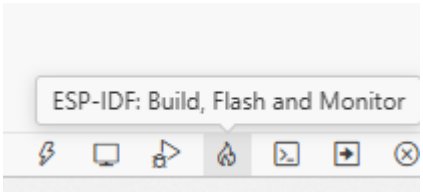
Compile and flash

Inside *cgi_studio_apps\References\ESP-IDF* a example project can be found. Open this project within Visual Studio code

It might be necessary to adjust the selected COM port. For this click the *COM<current port number>* on the bottom task bar and select one of the available com ports:



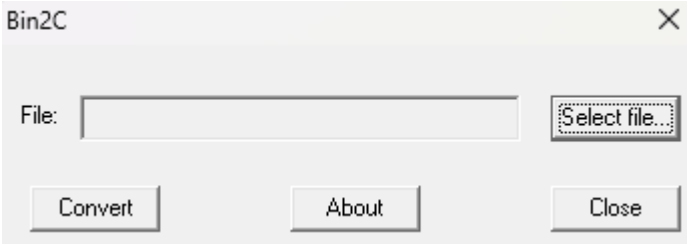
To compile and flash the target press the *ESP-IDF: Build, Flash and Monitor* on the bottom Task bar.



After compilation and flashing an example project can be seen on the display of the ESP32S3.

Replacing the Asset File

The current setup requires the Asset binary file to be converted to a C Array that can be placed directly within the code. For converting the binaries bin2c from segger can be used:



This program generates a file with a C array inside. This array needs to be copied into the following file:
cgi_studio_courier_apps\src\CourierSampleAppLight\AppPlatform\Target_FreeRTOS_XTENSA\Solution.c

The array needs to look like the following: *const unsigned char _ac2D_Basic*. Inside the AppEnvironment.cpp file of the same directory this array is referenced:

```
extern const unsigned char _ac2D_Basic[1130352UL + 1];
#define ASSET_START _ac2D_Basic
```

When updating the Solution.c make sure that the size of the array is the same inside Solution.c and in AppEnvironment.cpp

After the array has been replaced the project can be recompiled and flashed like described in the previous step.

Target Setup for the STM32MP135f and other Linux based targets

This guide aims to explain how to setup a Linux based target for the Software Renderer Platform and how to build a player for this target. The guide was tested on the STM32MP135f but the setup should be similar on most other Linux platforms that support DRM/KMS.

Before you begin

Before starting with the setup guide, we recommend the following to get familiar with CGI-Studio:

- Explore CGI-Studio

  Read the Quick Start Guide.
- Learn to create SceneComposer Solutions
  The documentation provides instructions on how to create simple SceneComposer solutions. To access SceneComposer navigate to:
  <cgi-studio-installation-path>\bin\SceneComposer\SceneComposer.exe
- Generate Assets for Simulation and Target Environments
  You can find guidance on generating assets from solutions (binary files containing solution information for the player) for both simulation and target environments in the documentation.

With this knowledge you should be fit to complete this guide.

Target and build setup

For setup of the ST32MP135f-DK follow the guide provided by STM: Populate the target and boot the image - stm32mpu

For setup of the SDK (inside a WSL or Linux host PC) to build a player for the STM32 follow this guide provided by STM: Install the SDK - stm32mpu

Building the Player

tested inside an Ubuntu WSL:

1. source the SDK like described in the Install the SDK Guide from STM
2. within the root directory of cgi trunk:
    1. mkdir build && cd build
    2. cmake -DCMAKE_TOOLCHAIN_FILE="../cgi_studio_devices/src/SoftwareRenderer2D/ToolchainFiles/GCC-toolchain-STM32MP135f-DrmKms-Linux.txt" -DCGIDEVICE_EVDEV_INPUT_ENABLED=ON ../cmake/Candera/Player/ && make -j

The Player should now be built and can be uploaded to the Target.

Uploading Player and Asset file to the target

1. Connect an USB-C Cable to CN7 USB-C of the target to the host pc
2. Connect USB-C Power cable to the PWR USB-C port
3. After the device boots a ssh connection can be established to the device:
    1. ssh root@192.168.7.162 #the static ip address of the target
    2. It might be necessary to kill weston if it is still running. It NEEDS to be killed otherwise there will be a conflict with the Player over display access
4. Copy the compiled player and in Scene Composer generated Asset.bin file to the target (using WinScp or scp) and run them like so:
    1. ./Player Asset.bin