Nvidia Drive PX2 development environment setup

This document will go over how to set up a virtual machine for development with the Driveworks API that is used for the Nvidia Drive PX2.

Notes

This is in no way the correct way of doing things, but it works good enough if there is no access to a computer with Ubuntu-16.04 and an Nvidia GPU with the right specifications. The guide might work differently if the computer has a Nvidia GPU with correct specifications.

Nvidia has been contacted about the issues at step 12, if the installer gets past 77.78%, we believe that you can jump directly to creating a development environment.

If needed to reflash the PX2 then you need to install minicom (sudo apt install minicom) then sdkmanager --archived-versions and then when installing don't press skip flashing as in step 11 (if doing on a virtual machine there can be problems with usb ports).

Login for Nvidia Drive PX2 is Username:nvidia password:nvidia.

Prerequisites

- Virtual machine with Ubuntu-16.04 x86_64 or desktop pc
- Access to Nvidia Drive PX2 developer program
- Around a 100 GB storage

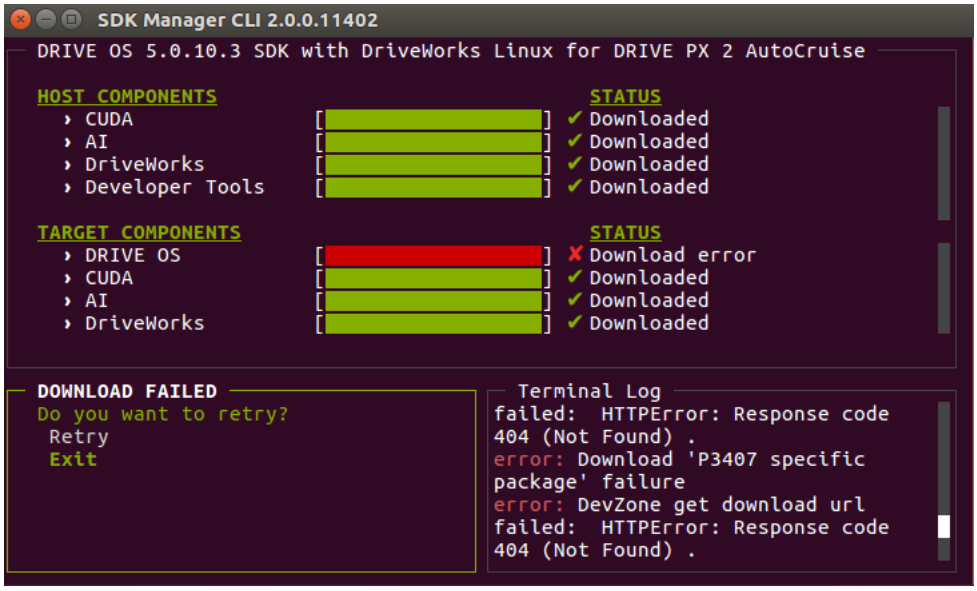
These are persons at MDU besides students in the HT2023 STAD project group that should have gotten access to the Nvidia Drive PX2 Developer Program (they might not even know they have it):
Martin Ekström - martin.ekstrom@mdu.se
Mikael Ekström - mikael.ekstrom@mdu.se
Carl Ahlberg - carl.ahlberg@mdu.se
Daniel Morberg - daniel.morberg@mdu.se
Henrik Falk - henrik.falk@mdu.se
Emil Persson - emil.persson@mdu.se
Lukas Dust - lukas.dust@mdu.se
Johan Hjorth - johan.hjorth@mdu.se
Fredrik Ekstrand - fredrik.ekstrand@mdu.se
Lennie Carlen Eriksson - lennie.carlen.eriksson@mdu.se
Niklas Persson - niklas.persson@mdu.se
Jonas Larsson - jonas.larsson@mdu.se

Setup

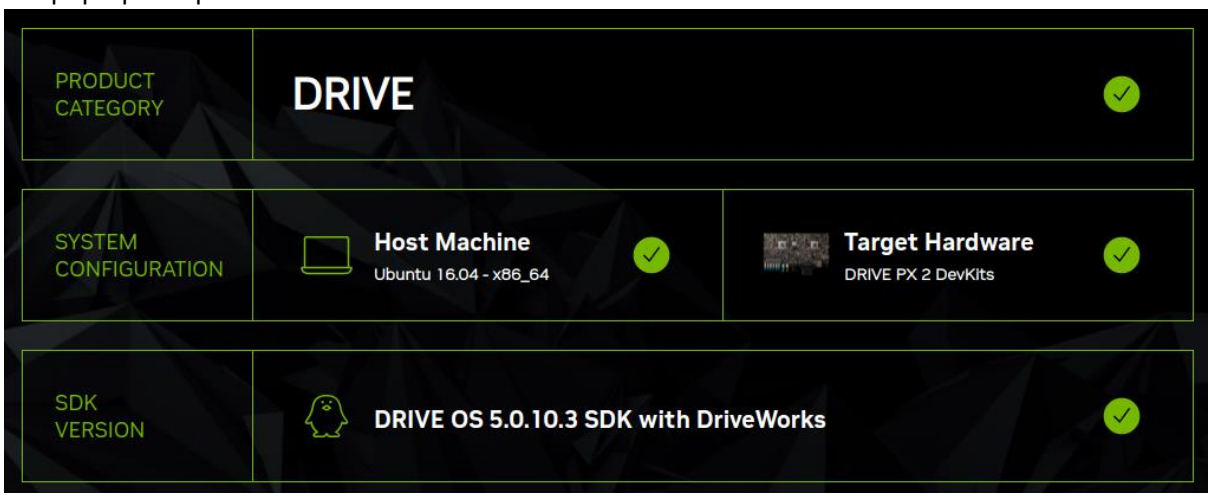
Terminal commands will be denoted with $

1. Start Ubuntu-16.04
2. $ sudo apt-get update && sudo apt-get upgrade
3. Download and install sdkmanager 2.0 from https://developer.nvidia.com/sdk-manager (requires login, see list above)
4. $ sdkmanager --cli --action downloadonly --login-type devzone --product DRIVE --target-os Linux --version OS_5.0.10.3_SDK --host --target PX2_AUTOCHAUFFEUR --license accept --archived-versions true

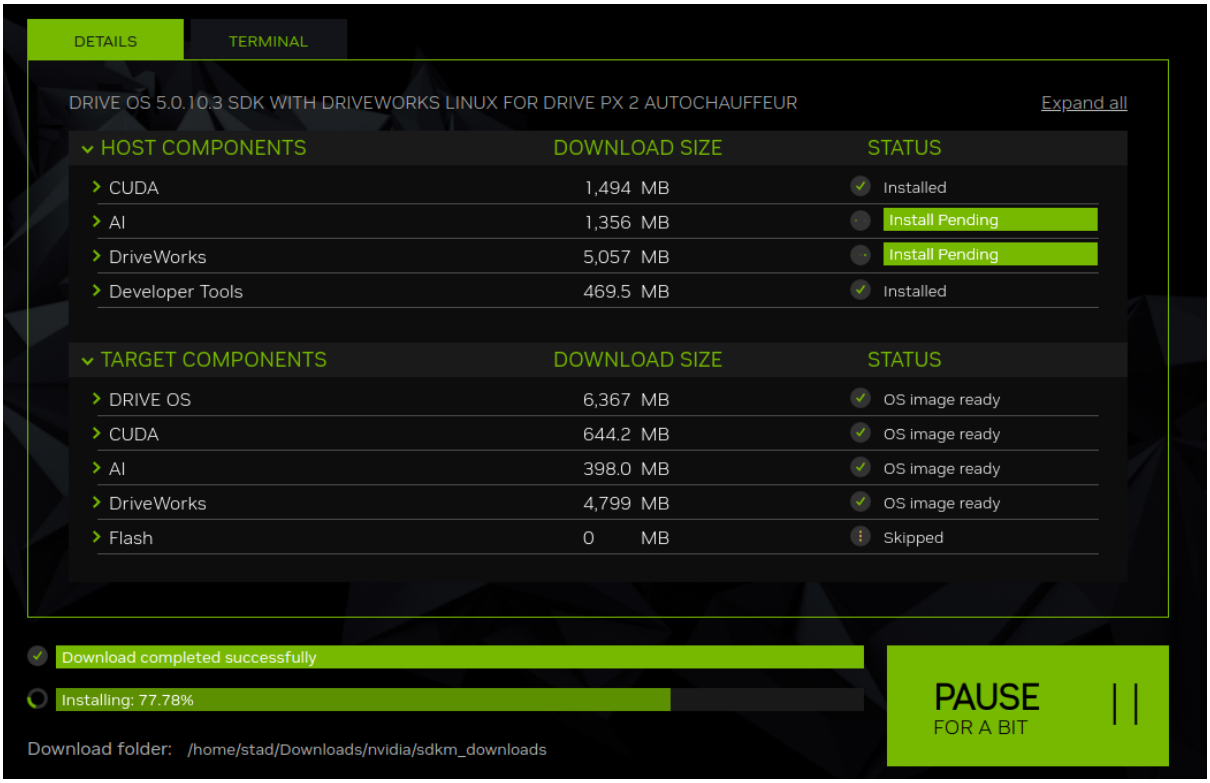
5. Login and enter passwords when prompted.
6. Download is complete when it looks like this and only error is about P3407



7. Exit
8. $ sdkmanager --archived-versions
9. Choose offline install and press start. **To access offline install, make sure to log out if logged in.**
10. These options should be set in the next menu, press continue and select Autochauffeur in the pop-up and press ok



11. In next window, accept the terms and press continue. This will start the installation.
    If promped to flash press skip.
12. When the installation reaches this step and has halted press pause and cancel installation



If the installation does not halt here I don't know what u did better than me but please update the guide.

13. $ cd ~/Downloads/nvidia/sdkm_downloads/
14. $ sudo apt install ./driveworks-v1.2.400-a7f5475-478955-nogcid-linux-amd64-ubuntu1604.deb
15. $ sudo apt install ./driveworks_data-v1.2.400-a7f5475-478955-nogcid-linux-amd64-ubuntu1604.deb
16. $ sudo apt install ./driveworks_cross_linux-v1.2.400-a7f5475-478955-12514001-drive-linux-5.0.10.3.deb
17. $ sudo apt install ./driveworks_samples-v1.2.400-a7f5475-478955-nogcid-linux-amd64-ubuntu1604.deb
18. $ sudo apt-get install openssh-server
19. $ sudo systemctl enable ssh –now
20. $ sudo systemctl start ssh

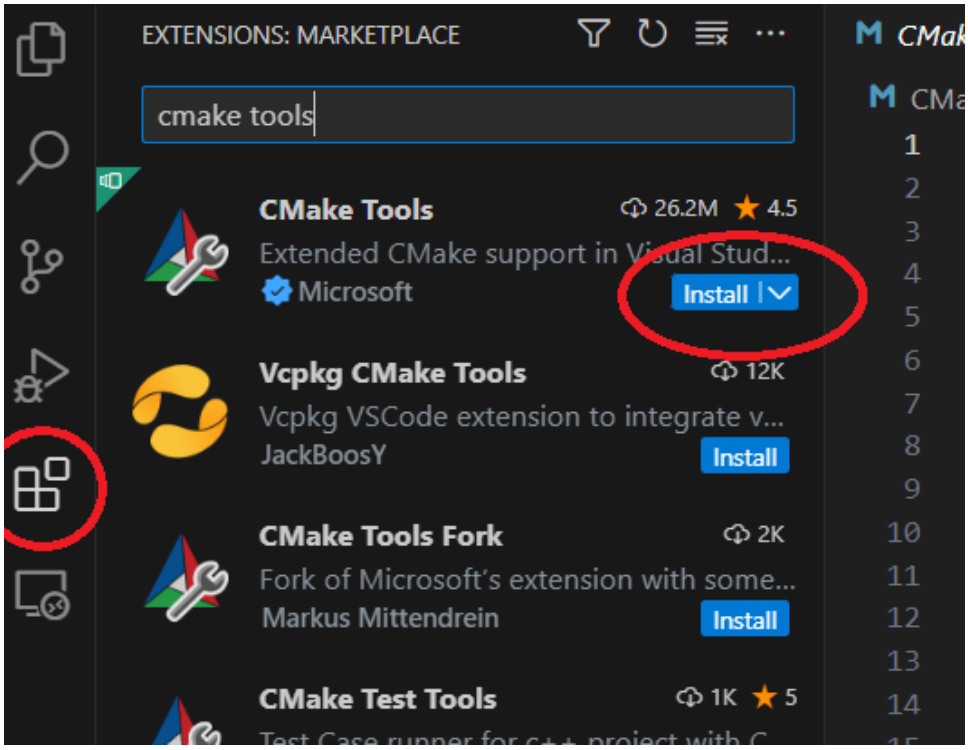
Creating a development environment

When all the libraries etc is installed this is how you setup a development environment in Vscode through ssh on another computer, skip step 1 if doing it directly on the computer with Driveworks installed.

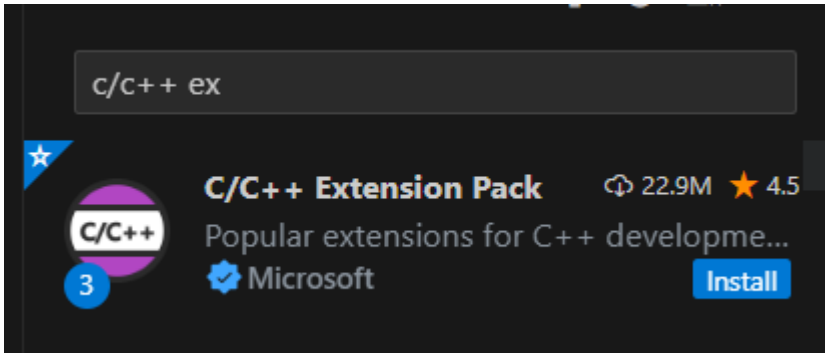
The documentation about the Driveworks API can be found in folder /usr/local/driveworks/doc open the index.html file.

1. Connect to machine through vscode with ssh (needs to install remote ssh plugin in vscode) Ip address can be found by using command ifconfig and then ssh using [username]@ipaddress to login. (Virtual machine might need to be given some permissions to receive an ip address)

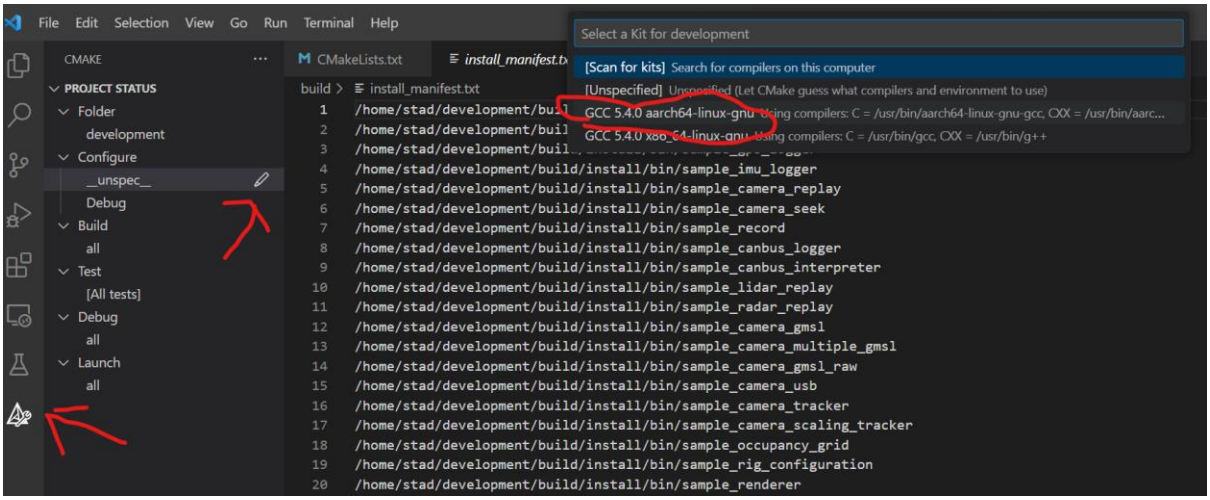
2. Open terminal in vscode
3. $ cd ~
4. $ cp -r /usr/local/driveworks/samples ./
5. $ mv samples <new folder name>
6. Open folder /home/<user>/<new folder name> in VS Code
7. Install cmake tools (used for syntax highlightning) in VS Code:



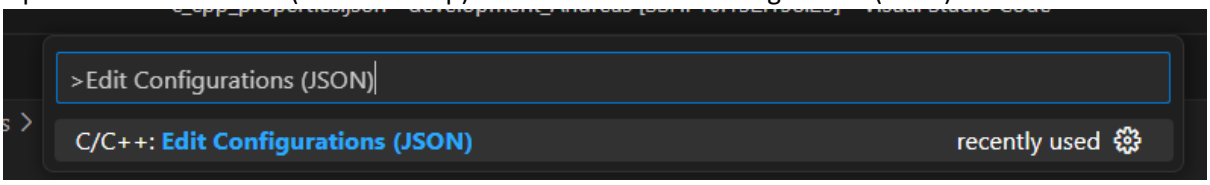
8. Install C/C++ Extenstion pack



9. $ sudo apt install cmake
10. In the CMake extension, click the edit icon on __unspec__ or [No Kit Selected] (may be something else depending on version) under Configure. Click GCC 5.4.0 aarch64-linux-gnu.



11. Open command Palette (ctrl + shift + p) and search for Edit Configuration (JSON) and click it.



In the file opened add the following and change <username>:

```
                "includePath": [
                    "${workspaceFolder}/**"
                    "/usr/local/driveworks-1.2/targets/aarch64-
      Linux/include/",
                    "/usr/local/cuda-9.2/targets/aarch64-
      linux/include/",
                "/home/<username>/nvidia/nvidia_sdk/
DRIVE_OS_5.0.10.3_SDK_with_DriveWorks_Linux_OS_PX2_AUTOCHAUFFEUR
      /DriveSDK/drive-t186ref-linux/include"
            ],
```

This step is for Driveworks syntax highlighting.

12. $ cd build
13. $ touch build.sh
14. $ chmod +x build.sh
15. Open build.sh and add the following, remember to change username (marked in yellow):

```
rm -rf 3rdparty
rm -rf CMakeFiles
rm -rf configured
rm -rf install
rm -rf src
rm -rf cmake_install.cmake
rm -rf CMakeCache.txt
```

```
rm -rf install_manifest.txt
rm -rf Makefile

cmake -DCMAKE_BUILD=Release \
    -DCMAKE_TOOLCHAIN_FILE=../cmake/Toolchain-V5L.cmake \
    -
DVIBRANTE_PDK:STRING=/home/<username>/nvidia/nvidia_sdk/DRIVE_OS_5
.0.10.3_SDK_with_DriveWorks_Linux_OS_PX2_AUTOCHAUFFEUR/DriveSDK/dr
ive-t186ref-linux \
    ..

make -j && make install
```

Note:
Formatting can be destroyed copy pasting, it should be formatted like this.

16. $ ./build.sh
17. $ cd install/bin
18. $ file sample_hello_world
19. If output says aarch64 and not x86_64 build is done correctly
20. Now you can make your own project from scratch and add it to <new folder name>/src or you can make a copy of one of the samples and rename it, when doing the project it should be added to this in CMakeLists.txt to be compiled.
21. Example:
    Creating a test project and adding it to be built
    $ mkdir src/test && cd src/test
    $ touch CMakeLists.txt main.cpp README.md
    Open CMakeLists.txt and add:

```
# Copyright (c) 2016, NVIDIA CORPORATION.  All rights reserved.

project(test C CXX)

# Project files
set(PUBLIC_DOCS
    README.md
)

set(SOURCES
    main.cpp
)

set(LIBRARIES
    ${Driveworks_LIBRARIES}
    )

# Final target
add_executable(${PROJECT_NAME} ${SOURCES})
target_link_libraries(${PROJECT_NAME} ${LIBRARIES})
```

```
set_property(TARGET ${PROJECT_NAME} PROPERTY FOLDER "Samples")

# Install target
sdk_add_sample(${PROJECT_NAME})
```

Open main.cpp and add the following or other code if wanted:

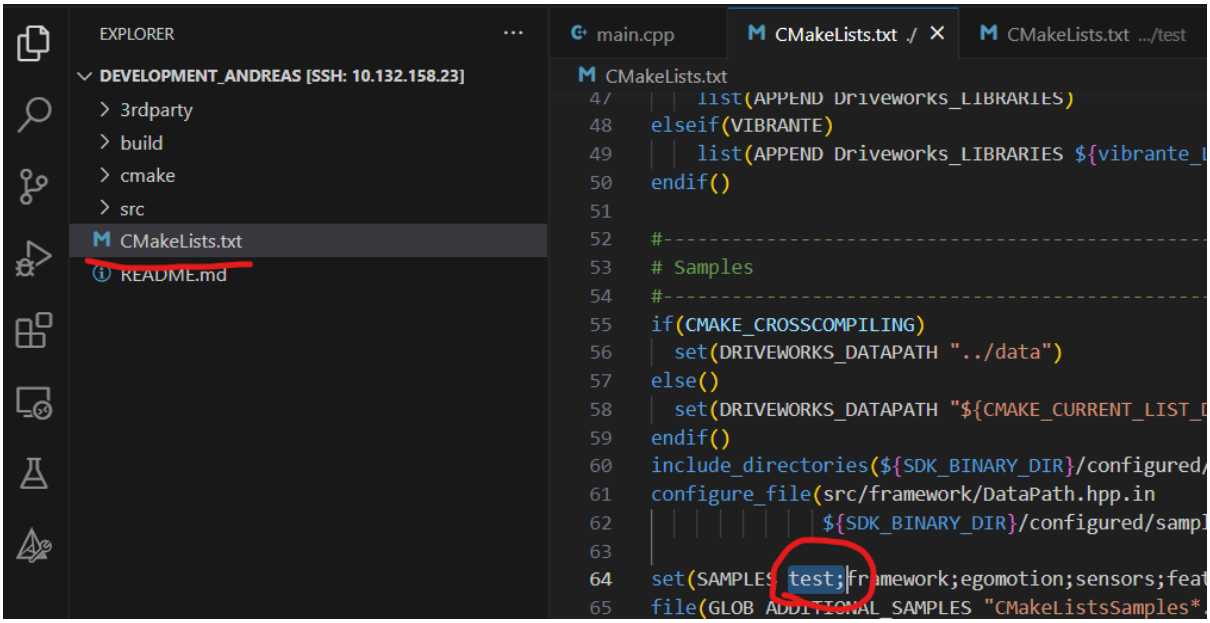
```
#include <dw/core/Context.h>
#include <stdio.h>
#include <iostream>

int main (int argc, char**
argv)
{
    (void)argc;
    (void)argv;

    dwVersion sdkVersion;
    dwGetVersion(&sdkVersion);

    std::cout << "Version: "
        << sdkVersion.major << "."
        << sdkVersion.minor << "."
        << sdkVersion.patch << std::endl;
}
```

Add test to the list of samples that should be built in <new folder name>/CMakeLists.txt (shown in next figure)



Go to <new folder name>/build and do ./build.sh again and check that the new project has been compiled and is also available in <new folder name>/build/install/bin

Note: All samples under src can be removed or modified but then you need to change the CMakeLists.txt to mirror the changes. Some samples build upon each other so if utilizing samples make sure to not delete the needed samples.

22. Now that the files are compiled they can be sent to the Nvidia Drive PX2 with for example scp if on the same network as the Nvidia Drive PX2.
    If in build folder do the following command
    $ scp install/bin/test nvidia@<PX2 ip address>:~
    Then enter password nvidia
    Log into the PX2 and check that test lies in /home/nvidia/ folder try running it from terminal and check that it can print Driveworks version