SONY

nVidia-Xavier-NX-BSP-Customization for Supporting FCB-EV9500M

01 March 2022

秘 : CONFIDENTIAL

Introduction

This document describes NVIDIA Jetson module Board Support Package (BSP) customization for supporting Sony FCB-EV9500M MIPI camera. This particular guide talks about NX module, but the procedure is same for all other modules.
Versions, names of downloaded files etc are subject to change as NVIDIA keeps updating the releases. Always use the latest stable ones.

©2022 Sony Electronics Inc.

Key Components

- Tool chain binary
- Kernel and driver sources
- Sample root file system
- DSC patches for each board and module type
- NVIDIA build and flash package
- Host PC which runs Ubuntu 18.04 (64-bit) - i3 or higher processor, 8GB RAM and at least 100GB free storage is recommended
- USB cable and other accessories to flash board - see user manual for Floyd: http://www.diamondsystems.com/products/floyd
- Sony FCB-EV9500M MIPI camera
- MIPI bridge board (Converting between I-PEX 30pin and KEL 30pin connectors)
    - Including I-PEX 30pin micro-coax cable, KEL 30pin micro-coax cable, and 12v AC adapter

©2022 Sony Electronics Inc.

Setting up Tool chain

Download the required tool chain from NVIDIA® website using https://developer.nvidia.com/embedded/downloads . You may need to login. Create an NVIDIA developer account if you don't have one already. Once downloaded, extract this into a directory, preferably named tool chain. Extract the tool chain.

- GCC Tool Chain Sources for 64bit BSP: gcc-linaro-7.3-2018.05.tar.xz

Please open a terminal and type the followings:

sony@sony-PC:~/nx$ mkdir toolchain sony@sony-PC:~/nx$
sony@sony-PC:~/nx$ nx$ cp ~/Downloads/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu.tar.xz toolchain/
sony@sony-PC:~/nx$ cd toolchain/ sony@sony-PC :~/nx/toolchain$ ls gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu.tar.xz sony@sony-PC :~/nx/toolchain$ sony@sony-PC :~/nx/toolchain$ tar -xvJf gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu.tar.xz

Once extracted, we can see all the tools in the bin directory.

©2022 Sony Electronics Inc.

Setting up Environment

Now we need to add the bin directory to system path. It is recommended to add extra path info, cross compile prefix and arch to a script and source this script every time we open a new terminal, this will save a lot of time.

Please open a terminal and type the followings:

```
sony@sony-PC:~/nx$ nano setenv.sh
```

and enter the parameters as required. In our case it is as below :

```
export CROSS_COMPILE=aarch64-linux-gnu-
export PATH=~/nx/toolchain/gcc-linaro-7.3.1-2018.05-x86_64_aarch64-linux-gnu/bin:$PATH
export ARCH=arm64
export LOCALVERSION=-tegra
```

To source this, every time we open a terminal, use below command.

```
sony@sony-PC :~/nx$ source setenv.sh
sony@sony-PC :~/nx$
```


©2022 Sony Electronics Inc.

Setting up L4T release directory and root file system

Download the required L4T release package and sample root file system from NVIDIA® website using https://developer.nvidia.com/embedded/downloads link.

We have tested with the L4T 32.5: https://developer.nvidia.com/embedded/linux-tegra-r325
Note: The latest ver L4T 32.6.1 seems to be an issue with CSI camera controls.

Download the following files:

- L4T Driver Package (BSP): Tegra186_Linux_R32.5.0_aarch64.tbz2
- Sample Root Filesystem: Tegra_Linux_Sample-Root-Filesystem_R32.5.0_aarch64.tbz2

©2022 Sony Electronics Inc.

Setting up L4T release directory and root file system

Extract the L4T release package:

Please open a terminal and type the followings:

sony@sony-PC :~/nx$ sudo tar -pxvjf Tegra186_Linux_R32.5.0_aarch64.tbz2

We can notice that there is a new Linux_for_Tegra directory. This will contain all the flashing tools. We need to deploy our root file system, kernel and all other dependencies here.

Next, we will expand and deploy our root file system here.
sony@sony-PC :~/nx$ sudo tar -pxvjf Tegra_Linux_Sample-Root-Filesystem_R32.5.0_aarch64.tbz2 -C Linux_for_Tegra/rootfs/

This will take a while. After this you can see the root fs populated.

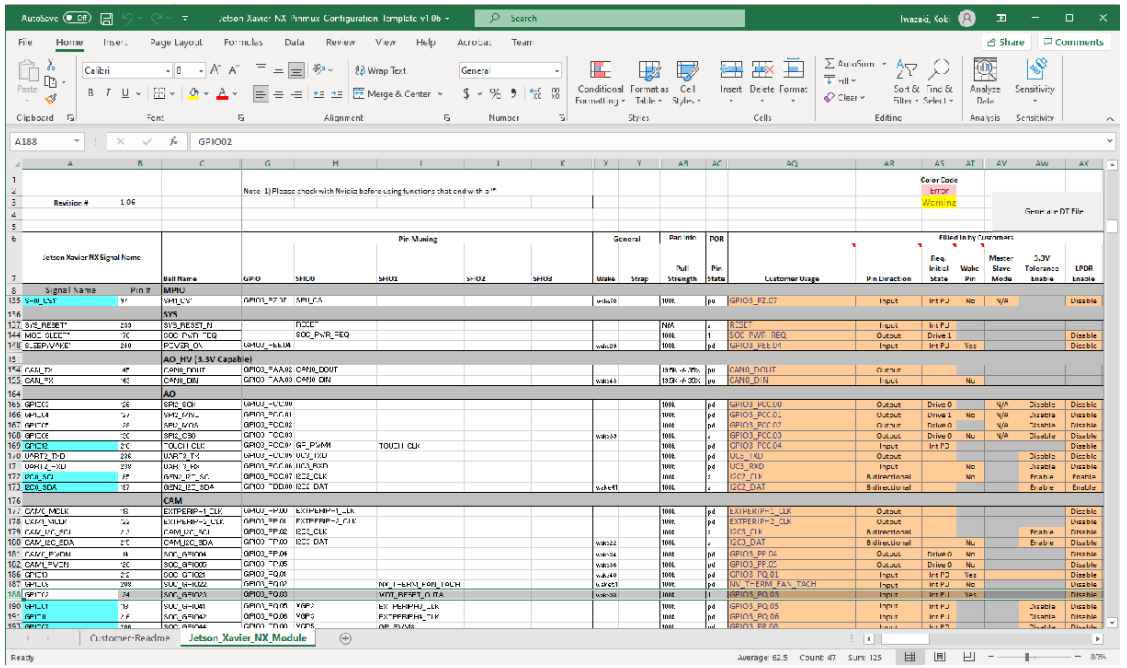
Once expanded, we need to install the packages into root file system. This is done by calling the script (part of L4T release package). This will deploy few packages into root file system. This will take a while.
sony@sony-PC :~/nx/Linux_for_Tegra$ sudo ./apply_binaries.sh

These are enough for flashing the module with stock BSP. We will now go on to build the customized kernel and show how to load into this BSP.

©2022 Sony Electronics Inc.

Pinmux modification

If Pinmux modifications are required for any reason, we can do it as follows.
Download the pinmux template sheet from NVIDIA site. This is an Microsoft Excel sheet with some macros enabled. It lets us to select the function of all user modifiable pins from a drop down list.

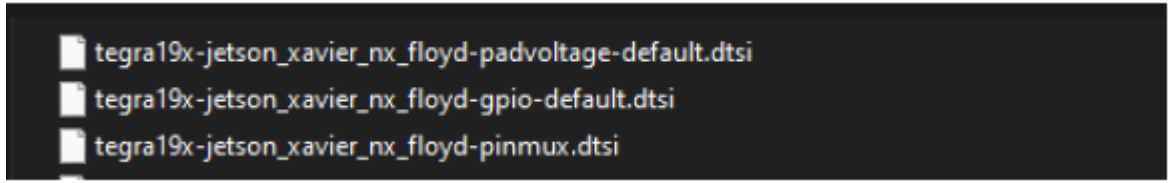


Notice the Generate DT button on the top right and column AQ has a drop down list. Select the required function from drop down list and save the doc. Once saved, click int he Generate DT button and Enter name for the pin mux dtsi files that will be generated by the macro.

©2022 Sony Electronics Inc.

秘 : CONFIDENTIAL

Pinmux modification



Copy these files to our build machine. The generated files will need to be processed further by a python script in host machine. This script is part of the L4T release package from NVIDIA.

sony@sony-PC :~/nx$ cd pinmux/ sony@sony-PC :~/nx/pinmux$ cd 01-rtsmodified/ sony@sony-PC :~/nx/pinmux/01-rtsmodified$ sony@sony-PC :~/nx/pinmux/01-rtsmodified$ ls Jetson_Xavier_NX_Pinmux_Configuration_Template_v1.06.xlsm tegra19x-gpio-rts-padvoltage-default.dtsi tegra19x-mb1-pinmux-p3668-a01.cfg tegra19x-gpio-rts-gpio-default.dtsi tegra19x-gpio-rts-pinmux.dtsi sony@sony-PC :~/nx/pinmux/01-rtsmodified$

Go to L4T directory, kernel/pinmux/<platform> . Platform could be t19x or t18x depending on the module. We can see the python script and some template files here.

sony@sony-PC nx/Linux_for_Tegra/kernel/pinmux/t19x$ ls addr_info.txt gpio_addr_info.txt mandatory_pinmux.txt pad_info.txt pinmux-dts2cfg.py por_val.txt README.txt sony@sony-PC :~/nx/Linux_for_Tegra/kernel/pinmux/t19x$

©2022 Sony Electronics Inc.

Pinmux modification

Now generate the pinmux cfg file as shown below.

sony@sony-PC :~/nx/Linux_for_Tegra/kernel/pinmux/t19x$ python pinmux-dts2cfg.py --pinmux addr_info.txt gpio_addr_info.txt por_val.txt --mandatory_pinmux_file mandatory_pinmux.txt ~/nx/pinmux/01-rtsmodified/tegra19x-gpio-rts-pinmux.dtsi ~/nx/pinmux/01-rtsmodified/tegra19x-gpio-rts-gpio-default.dtsi 1.0 > pinmux.cfg

Paths of the input files have to modified as required. There may be some warnings that will be print to console, this can be safely neglected. Once done we can see a new file pinmux.cfg generated in the script directory.

sony@sony-PC :~/nx/Linux_for_Tegra/kernel/pinmux/t19x$ ls addr_info.txt gpio_addr_info.txt mandatory_pinmux.txt pad_info.txt pinmux.cfg pinmux-dts2cfg.py por_val.txt README.txt sony@sony-PC :~/nx/Linux_for_Tegra/kernel/pinmux/t19x$

This has to be renamed to the exact pinmux file name. This comes from the flashing config file. In the case of NX, it is tegra19x-mb1-pinmux-p3668-a01.cfg

©2022 Sony Electronics Inc.

Pinmux modification

Here is how to obtain this :

sony@sony-PC :~/nx/Linux_for_Tegra$ ls -al | grep nx lrwxrwxrwx 1 root root 35 Jun 26 10:04 jetson-xavier-as-xavier-nx.conf -> p2822+p2888-0001-as-p3668-0001.conf lrwxrwxrwx 1 root root 34 Jun 26 10:04 jetson-xavier-nx-devkit.conf -> p3509-0000+p3668-0000-qspi-sd.conf lrwxrwxrwx 1 root root 36 Jun 26 10:04 jetson-xavier-nx-devkit-emmc.conf -> p3509-0000+p3668-0001-qspi-emmc.conf lrwxrwxrwx 1 root root 31 Jun 26 10:04 jetson-xavier-nx-devkit-qspi.conf -> p3509-0000+p3668-0000-qspi.conf sony@sony-PC :~/nx/Linux_for_Tegra$

Our flashing script jetson-xavier-nx-devkit-emmc.conf is a symlink to p3509-0000+p3668-0001-qspi-emmc.conf. Lets open that one.

p3509-0000+p3668-0001-qspi-emmc.conf: configuration for "P3668(SKU1) + P3509" # (T194 P3518). source "${LDK_DIR}/p3668.conf.common"; EMMC_CFG=flash_l4t_t194_spi_emmc_p3668.xml; EMMCSIZE=17179869184;

And it internally sources p3668.conf.common for all the config. Let's open that.

~~~~~~ SCR_COLD_BOOT_CONFIG="tegra194-mb1-bct-scr-cbb-mini-p3668.cfg"; SCR_CONFIG="tegra194-mb1-bct-scr-cbb-mini-p3668.cfg"; PINMUX_CONFIG="tegra19x-mb1-pinmux-p3668-a01.cfg"; PMIC_CONFIG="tegra194-mb1-bct-pmic-p3668-0001-a00.cfg"; PMC_CONFIG="tegra19x-mb1-padvoltage-p3668-a01.cfg"; ~~~~~~

©2022 Sony Electronics Inc.

秘 : CONFIDENTIAL
~~~~~~

Pinmux modification

And we can see the pinmux file name is tegra19x-mb1-pinmux-p3668-a01.cfg and is found in Linux_for_Tegra/bootloader/t186ref/BCT

sony@sony-PC :~/nx/Linux_for_Tegra/bootloader/t186ref/BCT$ ls | grep pinmux tegra186-mb1-bct-pinmux-quill-p3310-1000-a00.cfg tegra186-mb1-bct-pinmux-quill-p3310-1000-c00.cfg tegra186-mb1-bct-pinmux-quill-p3310-1000-c03.cfg tegra186-mb1-bct-pinmux-quill-p3489-1000-a00.cfg tegra19x-mb1-pinmux-p2888-0000-a00-p2822-0000-a00.cfg tegra19x-mb1-pinmux-p2888-0000-a04-p2822-0000-b01.cfg tegra19x-mb1-pinmux-p2888-0000-p2822-0000.cfg tegra19x-mb1-pinmux-p2888-slvs-0000-a00-p2822-0000-a00.cfg tegra19x-mb1-pinmux-p3668-a01.cfg tegra19x-mb1-pinmux-p3668-a01.cfg.stock
sony@sony-PC :~/nx/Linux_for_Tegra/bootloader/t186ref/BCT$

So our newly generated pinmux cfg needs to be renamed to this file and replaced with this one, or we can add a new entry by commenting out this entry.
That’s all. On next flashing, this will be updated to module.

Note: Please ask carrier board supplier for a pinmux file and copy the file to the above directory for a new build.

©2022 Sony Electronics Inc.

Building Kernel, modules and dtb

Download the required L4T release package and sample root file system from NVIDIA® website using https://developer.nvidia.com/embedded/downloads link.

We have tested with the L4T 32.5: https://developer.nvidia.com/embedded/linux-tegra-r325
Note: The latest ver L4T 32.6.1 seems to be an issue with CSI camera controls.

Download the following files:

- L4T Driver Package (BSP): Sources: public_sources.tbz2

Please open a terminal and type the followings:

```
sony@sony-PC :~/nx$ mkdir source sony@sony-PC :~/nx$ cp ~/Downloads/public_sources.tbz2 source\
sony@sony-PC :~/nx$ cd source/ sony@sony-PC :~/nx/source$ tar -pxvjf public_sources.tbz2
```

We are interested in kernel sources. Go ahead and extract kernel sources

```
sony@sony-PC :~/nx/source/Linux_for_Tegra/source/public$ tar -pxvjf kernel_src.tbz2
```

Notice the two additional directories created 'hardware' and 'kernel'.

©2022 Sony Electronics Inc.

FCB-EV9500M driver

This is a driver source code (ev9500m.c) for FCB-EV9500M MIPI camera.

Please copy this file to the following directory:

sony@sony-PC:~/nx/source/Linux_for_Tegra/source/public/kernel/nvidia/drivers/media/i2c/

In this folder, please open Makefile to add a following line:
obj-$(CONFIG_VIDEO_EV9500M) += ev9500m.o

Also, please open Kconfig to add a following line:

```
config VIDEO_EV9500M
        tristate “FCB-EV9500M camera support"
        depends on I2C && VIDEO_V4L2 && VIDEO_V4L2_SUBDEV_API
        ---help---
        This is a Video4Linux2 sensor-level driver for the Sony
        FCB-EV9500M MIPI camera

        To compile this driver as a module, choose M here: the module
        will be called ev9500m.
```


©2022 Sony Electronics Inc.

Device Tree for FCB-EV9500M

This is a device tree (tegra194-camera-ev9500m.dtsi)for FCB-EV9500M MIPI camera which sets CSI 4 lanes.

Please copy this file to the following directory:

sony@sony-PC:~/nx/source/Linux_for_Tegra/source/public/hardware/nvidia/platform/t19x/jakku/kernel-dts/common/

In this folder, please open tegra194-p3668-common.dtsi file to add a following line:
#include "tegra194-camera-ev9500m.dtsi“

Also, please open tegra194-p3509-0000-a00.dtsi file to comment out the following:
//#include "tegra194-camera-jakku-rbpcv2-imx219.dtsi“

©2022 Sony Electronics Inc.

tegra_defconfig: module configuration

This is a tegra_defconfig file to set modules you want to compile.

Please open tegra_defconfig file in the following directory:

sony@sony-PC:~/nx/source/Linux_for_Tegra/source/public/kernel-4.9/arch/arm64/config/

Please add a following line:
CONFIG_VIDEO_EV9500M=y

©2022 Sony Electronics Inc.

秘 : CONFIDENTIAL

BSP Patch for DSC carrier boards

NVIDIA stock BSP is targeted for NVIDIA evaluation boards. When building BSP for DSC carrier boards, certain changes are to be made for the BSP. The below section talks about this.
Patches are provided for each carrier supplier. These patches are made against specific L4T version. Those version info is provided. When applying patches to an updated L4T version, these may fail. You may have to modify the patches to match the new BSP.

Patching may be done manually if L4T release package is not L4T 32.5.0. Once patching is done, go for build procedure below:
We can now start building kernel

Note: Remember, we already have sourced our setenv.sh scrip before. Terminal that does the following steps need to have those environment settings. If you are working in a new terminal, don't forget to source the script again.

©2022 Sony Electronics Inc.

BSP Patch for DSC carrier boards

Configure the kernel sources
sony@sony-PC :~/nx/source/Linux_for_Tegra/source/public/kernel$ cd kernel-4.9/ make clean
sony@sony-PC :~/nx/source/Linux_for_Tegra/source/public/kernel/kernel-4.9$ make tegra_defconfig

Sometimes, the extra patches that we apply for customizing BSP may provide a separate config file. Use that configuration file in such cases.
start building
sony@sony-PC :~/nx/source/Linux_for_Tegra/source/public/kernel/kernel-4.9$ make -j4 Image .

This will take a while depending on your host system capabilities. If everything goes smooth, we should be able to see kernel binary getting generated.

Next we will build modules.
sony@sony-PC :~/nx/source/Linux_for_Tegra/source/public/kernel/kernel-4.9$ make -j4 modules

Now, we need to deploy the modules we built into our root file system
sony@sony-PC:~/nx/source/Linux_for_Tegra/source/public/kernel/kernel-4.9$ sudo -E make INSTALL_MOD_PATH=~/nx/Linux_for_Tegra/rootfs/ modules_install

©2022 Sony Electronics Inc.

秘 : CONFIDENTIAL

BSP Patch for DSC carrier boards

Copy our built kernel and dtb to L4T directory.

[Copy Image]
sony@sony-PC:~/nx/Linux_for_Tegra$ cp -rfv ~/nx/source/Linux_for_Tegra/source/public/kernel/kernel-4.9/arch/arm64/boot/Image kernel/

[Copy DTBS]
sony@sony-PC:~/nx/Linux_for_Tegra$ cp -rfv ~/nx/source/Linux_for_Tegra/source/public/kernel/kernel-4.9/arch/arm64/boot/dts/* kernel/dtb/

©2022 Sony Electronics Inc.

Flashing module with new BSP

Connect module in recovery mode. (See user manual of your carrier board for more details). Once the module is out in recovery, it should be listed as a USB device in host. Type lsusb and you should be able to see something like this.

```
sony@sony-PC :~/nx$ lsusb
~~~~
Bus 001 Device 102: ID 0955:7f21 NVidia Corp.
~~~~
```

Go to L4T directory and start flashing.

```
sony@sony-PC :~/nx/Linux_for_Tegra$ sudo ./flash.sh jetson-xavier-nx-devkit-emmc mmcblk0p1
```

This will take about 20 minutes, and the module will restart itself after flashing.

Now if Jetson Xavier NX has connected the display, it will ask for some user configuration details and you will be directed to the Ubuntu desktop.

©2022 Sony Electronics Inc.

Setup Procedure

This section describes how to verify the setup before start testing Gstreamer pipelines.
To know the details of available plugins, please refer to the
Accelerated_Gstreamer _User_Guide.pdf from Jetson Download Cente

The steps to verify the setup before testing Gstreamer pipelines are as follows:

1. Run the following command to check the Gstreamer-1.0 version.

   $ gst-inspect-1.0 --version

   The output message appears as shown below.

   gst-inspect-1.0 version 1.14.5
   GStreamer 1.14.5

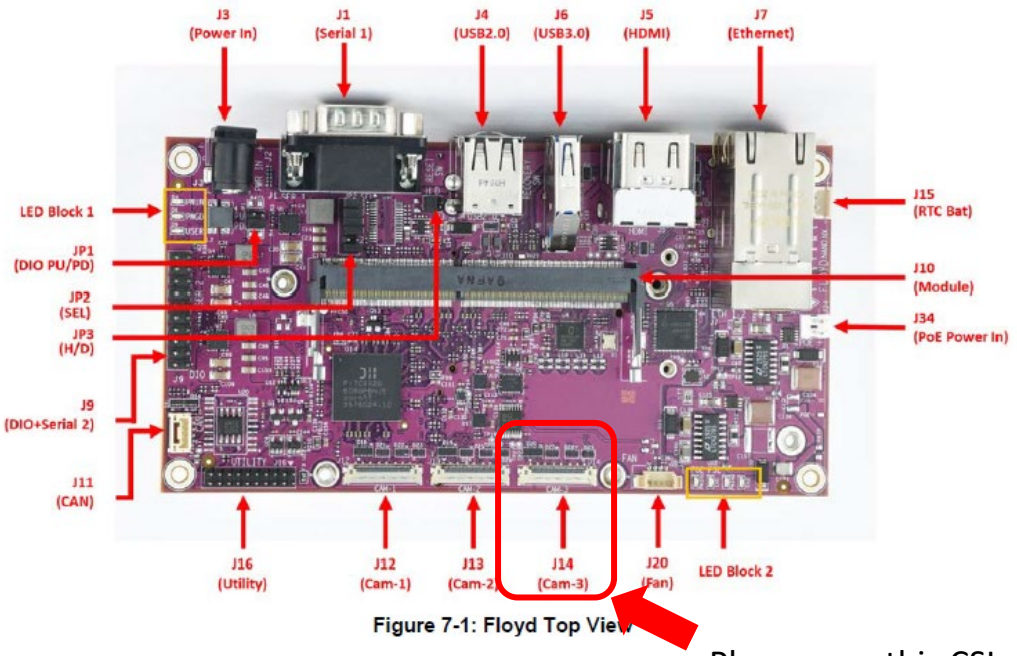


Figure 7-1: Floyd Top View

Please use this CSI port

Note: Make sure that FCB-EV9500M with a converter board is connected and the required drivers are loaded. During booting, the module drivers for FCB-EV9500M will be loaded automatically in the Diamond Systems FLOYD NX carrier
board.

©2022 Sony Electronics Inc.

秘 : CONFIDENTIAL

Setup Procedure

2. Run the following command to confirm whether the camera is initialized.

   $ dmesg | grep -i “ev9500m"

   The output message appears as shown below.

   subdev ev9500m 2-0010 bound

   The output message indicates that the camera is initialized properly.

3. Run the following command to check the presence of video node.

   $ ls /dev/video*

   The output message appears as shown below.

   /dev/video

   where (*) depends on the number for cameras connected to the Diamond Systems FLOYD carrier board. The number of video nodes displayed depends on the number of cameras connected.

©2022 Sony Electronics Inc.

Test Video Image using Gstreamer

Note: Please run the following commands to change the power mode to maximum for better performance and to get maximum frame rate.

```
$ sudo nvpmodel -m 0
$ sudo jetson_clocks
```

Run the following command for a video test:

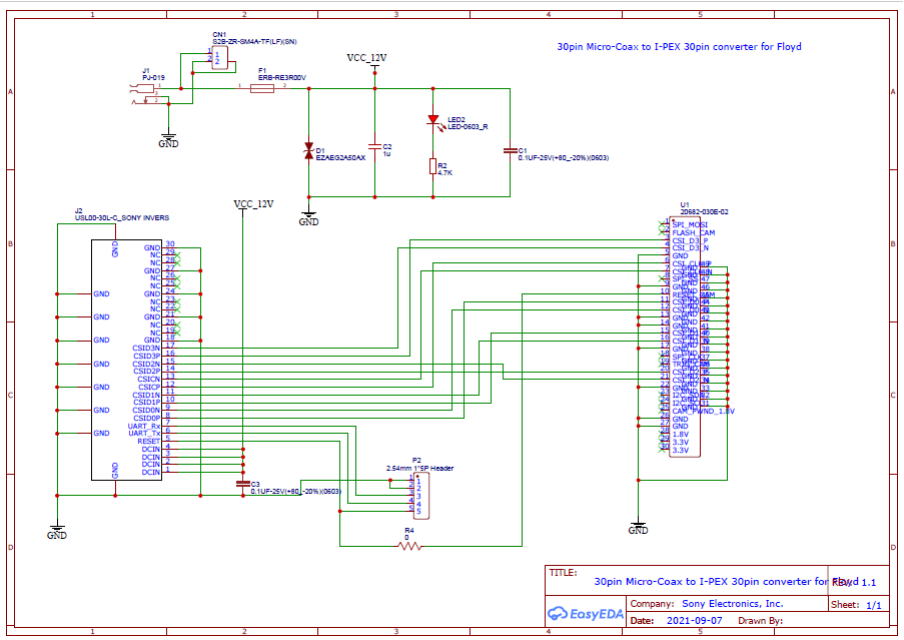
```
gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw,format=UYVY ! videoconvert ! xvimagesink
```


©2022 Sony Electronics Inc.

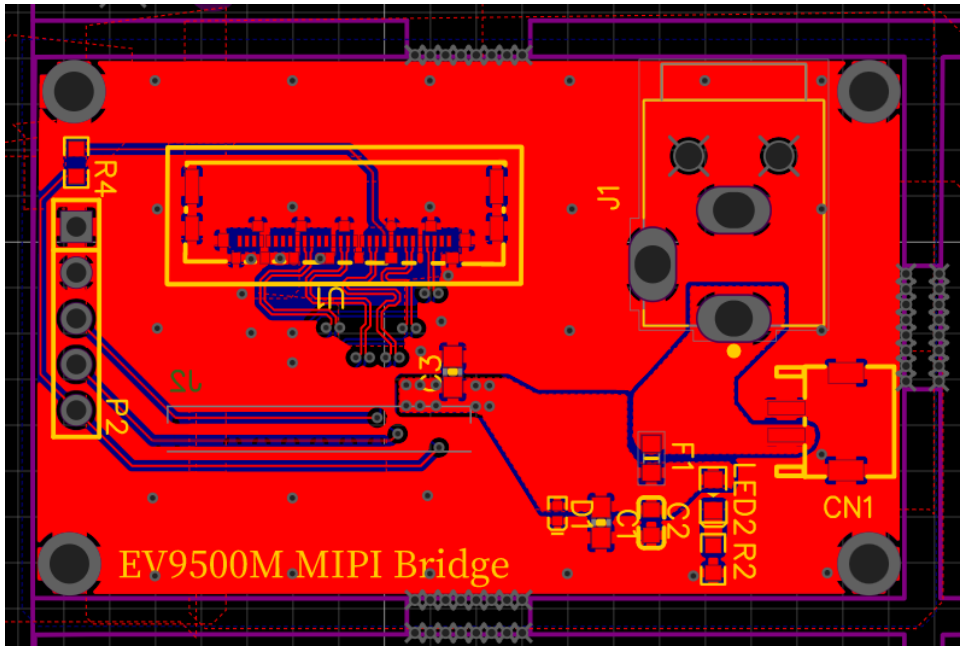
Connector Converter Board Design

Each carrier board has unique connectors for CSI interface. The Diamond System FLOYD system has I-PEX 30pin connector(20682-030E-02) which requires a connector converter board to connect FCB-EV9500M KEL 30pin connector.

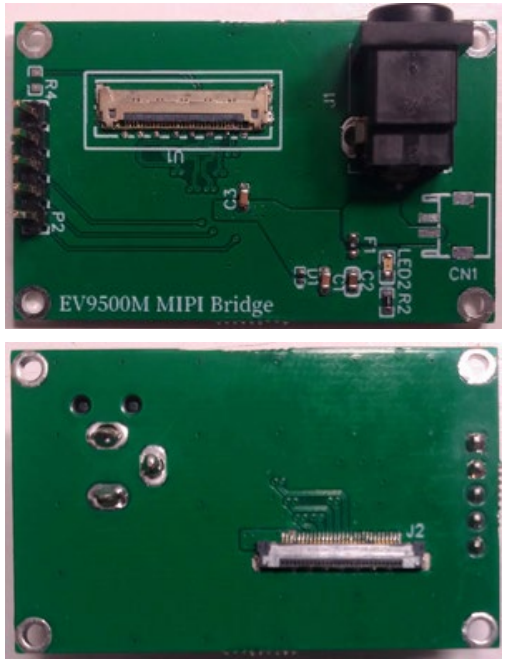
We used EasyEDA online tool to design the board: https://easyeda.com/



Schematic



PCB Layout



Actual Board

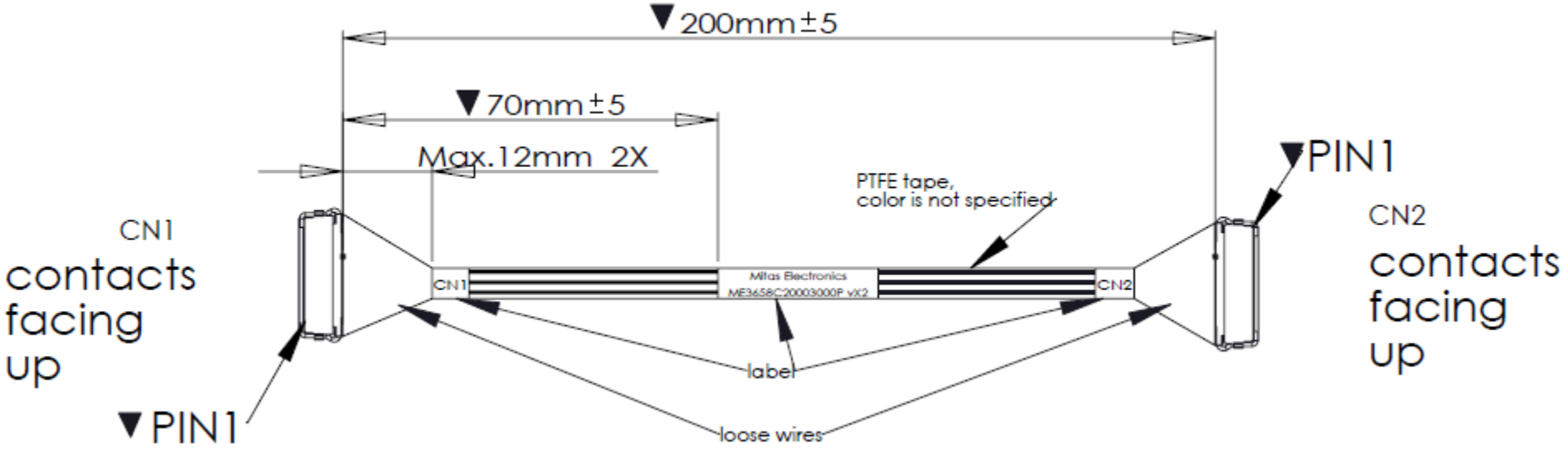
©2022 Sony Electronics Inc.

秘 : CONFIDENTIAL

I-PEX 30pin Micro-Coax Cable

I-PEX 30pin Micro-Coax cable can be ordered from the following suppliers:

- HKsino-meda: andy@hksino-media.com
- Raynool: oscar@raynool.com
- Mitas-Electronics: esales@mitaselectronics.com



©2022 Sony Electronics Inc.

秘 : CONFIDENTIAL

SONY is a registered trademark of Sony Corporation.
Names of Sony products and services are the registered trademarks and/or trademarks of Sony Corporation or its Group companies.
Other company names and product names are registered trademarks and/or trademarks of the respective companies.