

# EX713 BSP Setup Instructions

# EX713 BSP Setup Instructions

If you have difficulties to access the BSP download link, please contact technical support at [https://www.avermedia.com/professional/technical\\_support](https://www.avermedia.com/professional/technical_support) or e-mail us at [eusupport@avermedia.com](mailto:eusupport@avermedia.com) for further assistance.

## Install steps for L4T

The cifX driver can be installed in two ways, L4T kernel can support Ethercat driver so one is open kernel FLAG to build hilscher driver , other one is build hilscher driver from hilscher FAE.

- Use ubuntu14.04 PC to build L4T kernel and open Ethercat card driver in L4T kernel.

- Download L4T kernel source and toolchain.

(<https://developer.nvidia.com/embedded/downloads>)

<p>▼ GCC Tool Chain for 64-bit BSP</p> <p>This is a pre-built 64-bit ARM (aarch64) cross-compiling toolchain that runs on an x86_64 Linux host PC. Unpack this toolchain at the desired location and use standard cross-compilation directives to use it.</p> <p><a href="#">More Information &gt;</a></p>	<p>28.2</p> <p>2018/03/08</p> <p>DOWNLOADS</p> <p><a href="#">GCC 4.8.5 Tool Chain for 64-bit BSP</a></p>
<p>▼ L4T TX2 Sources</p> <p>This file contains source data for the 28.2.1 BSP. It contains sources for kernel, u-boot, gstegl, gstjpeg, gstomx, gstomx1, and nvgstapps.</p> <p><a href="#">More Information &gt;</a></p>	<p>28.2.1</p> <p>2018/06/13</p> <p>DOWNLOADS</p> <p><a href="#">L4T 28.2.1 TX2 Sources</a></p>

- Untar the file

- ◆ `tar xvf ./gcc-4.8.5-aarch64.tgz`
- ◆ `tar xvf public_sources.tbz2`
- ◆ `cd public_release`
- ◆ `tar xvf kernel_src.tbz2`

- Export parameter

- ◆ `export ARCH=arm64`
- ◆ `export CROSS_COMPILE=/home/jim/storage-disk/ex731/ethercat/web_kernel/install/bin/aarch64-unknown-linux-gnu-`

(PS. CROSS\_COMPILE path is from gcc-4.8.5-aarch64.tgz)

■ Build kernel

- ◆ cd ./public\_release/kernel/kernel-4.4
- ◆ mkdir kernel-out
- ◆ make O=./kernel-out/ tegra18\_defconfig

■ Modify the menuconfig and set local version (default jetpack have local version)

- ◆ make O=./kernel-out/ menuconfig
- ◆ path : General setup ---> Local version - append to kernel release

```
[ ] Compile also drivers which will not load
(-tegra) Local version - append to kernel release
[ ] Automatically append version information to the version string
((none)) Default hostname
```

■ Open UIO config

- ◆ path: Device Drivers ---> <M> Userspace I/O drivers ---> <M> Hilscher NetX Card driver

```
[*] DMA Engine support --->
[ ] Auxiliary Display support ----
<M> Userspace I/O drivers ----
< > VFIO Non-Privileged userspace driver framework ----
< > Generic driver for PCI 2.3 and PCI Express cards (NEW)
<M> Hilscher NetX Card driver
< > Texas Instruments PRUSS driver (NEW)
```

■ Open syn cookie

- ◆ path: [\*] Networking support ---> Networking options ---> [\*] IP: TCP syncookie support

```
[ ] IP: PIM-SM version 2 support
[*] IP: TCP syncookie support
< > IP: Foo (IP protocols) over UDP
[ ] IP: FOU encapsulation of IP tunnels
```

■ Save and build

- ◆ make O=./kernel-out/ -j4

■ Copy Image and module to device

- ◆ scp ./kernel-out/arch/arm64/boot/Image [nvidia@192.168.23.219:./](mailto:nvidia@192.168.23.219)
- ◆ scp ./kernel-out/drivers/uio/uio.ko [nvidia@192.168.23.219:./](mailto:nvidia@192.168.23.219:./)
- ◆ scp ./kernel-out/drivers/uio/uio\_netx.ko [nvidia@192.168.23.219:./](mailto:nvidia@192.168.23.219:./)

- Update TX2
  - ◆ `sudo cp -rf ./Image /boot`
- Reboot and install driver on TX2
  - ◆ `sudo insmod ./uio.ko`
  - ◆ `sudo insmod ./uio_next.ko`
- install libpciaccess-dev
  - ◆ `sudo apt-get update`
  - ◆ `sudo add-apt-repository universe`
  - ◆ `sudo apt-get update`
  - ◆ `sudo apt-get install module-assistant ufw`
  - ◆ `sudo apt-get install libpciaccess-dev`
- Build LAT kernel UIO feature and build hilscher driver.
  - Followed first way to open “Userspace I/O drivers” and build kernel.  
CD\_V1.1.4.0 hilscher driver only have uio\_next.ko driver so it need kernel open UIO default feature.
  - Build hilscher card
    - ◆ `export`  
`KDIR=/home/jim/storage-disk/ex731/ethercat/web_kernel/public_release/kernel/`  
`kernel-4.4/kernel-out/`
 (Need export your kernel path before build hilscher card.)
    - ◆ `export ARCH=arm64`
    - ◆ `export CROSS_COMPILE=`  
`/home/jim/storage-disk/ex731/ethercat/web_kernel/install/bin/`  
`aarch64-unknown-linux-gnu-`
    - ◆ `cd ~/storage-disk/ex731/ethercat/CD_V1.1.4.0/driver/uio_netx`
    - ◆ `make` (you can get uio\_next.ko driver file)
  - Update TX2
    - ◆ `scp ./kernel-out/arch/arm64/boot/Image nvidia@192.168.23.219:./`

- ◆ `scp ./kernel-out/drivers/uiso/uiso.ko nvidia@192.168.23.219:./`
- ◆ `scp ~/storage-disk/ex731/ethercat/CD_V1.1.4.0/driver/uiso_netx /uiso_netx.ko nvidia@192.168.23.219:./`
- ◆ `sudo cp -rf ./Image /boot`
- Reboot and install driver on TX2
  - ◆ `sudo insmod ./uiso.ko`
  - ◆ `sudo insmod ./uiso_next.ko`
- Install libpciaccess-dev
  - ◆ `sudo apt-get update`
  - ◆ `sudo add-apt-repository universe`
  - ◆ `sudo apt-get update`
  - ◆ `sudo apt-get install module-assistant ufw`
  - ◆ `sudo apt-get install libpciaccess-dev`