

Jetpack Kernel Driver

Version 2.1

February 17 2021

Project Lead: Harry Li, Ph.D.

Team member:

Yusuke Yakuwa

Nitin Patil

Akshat Bhutinai

CTI One Corporation

3679 Enochs St

Santa Clara, CA, 95051

www.ctione.com

Table of Contents

1. Download 3 files.....	4
1.1 Create a work directory.....	4
1.2. Download Cross compiler tool (GCC 4.8.5 Tool Chain for 64-bit BSP).....	4
1.3. Download OS Source Code.....	4
1.4. Download Documents.....	4
2. Setup the Kernel compiling environment.....	4
2.1. Extract documentation.....	4
2.2. Extract the Toolchain.....	4
2.3. Extract the kernel sources.....	4
2.4. Install the utilities.....	5
2.5. Set the shell variable with the command.....	5
2.6. Make the build directory.....	5
2.7. Set the system architecture variable.....	5
2.8. Locate the kernel source directory.....	5
2.9. Create a .config.....	6
3. GUI MenuConfig.....	7
3.1. Run make menuconfig.....	7
3.2. Load .config.....	7
3.3. Modify SPI Driver settings.....	8
4. Build the Kernel source.....	10

Feb 4, 2021

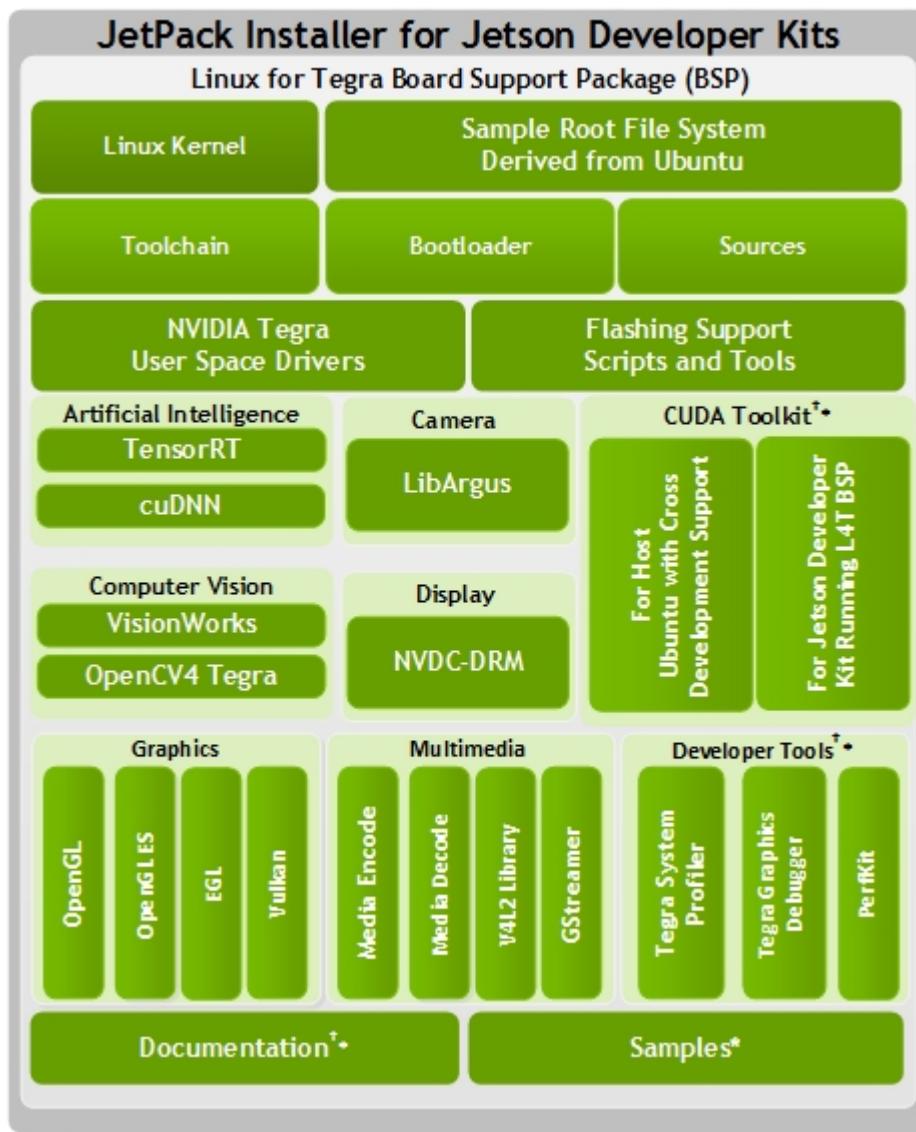
1. Source From :

In nnn-SPI-driver-JetsonTX2-v01-MO-2021-02-03.odp, the previous development was based on "getKernelSources.sh" to download source files, however, it can only be run on TX2.

(YY Feb 8, 2021)

Instead of that, I downloaded the source code from the website below.

<https://developer.nvidia.com/embedded/linux-tegra-r2821>



* Target System

1. Download 3 files

1.1 Create a work directory

```
mkdir /home/ctione/Documents/JetPack_build
```

1.2. Download Cross compiler tool (GCC 4.8.5 Tool Chain for 64-bit BSP)

gcc-4.8.5-aarch64.tgz
<https://developer.nvidia.com/embedded/dlc/l4t-gcc-toolchain-64-bit-28-2-ga>

Put gcc-4.8.5-aarch64.tgz to /home/ctione/Documents/JetPack_build

1.3. Download OS Source Code

public_sources.tbz2
<https://developer.nvidia.com/embedded/dlc/sources-r2821>

Put public_sources.tbz2 to /home/ctione/Documents/JetPack_build

1.4. Download Documents

NVIDIA_Tegra_Linux_Driver_Package.tar
<https://developer.nvidia.com/embedded/dlc/l4t-documentation-28-2-ga>

Put NVIDIA_Tegra_Linux_Driver_Package.tar to /home/ctione/Documents/JetPack_build

2. Setup the Kernel compiling environment

2.1. Extract documentation

```
mkdir NVIDIA_Tegra_Linux_Driver_Package
tar xvf NVIDIA_Tegra_Linux_Driver_Package.tar -C NVIDIA_Tegra_Linux_Driver_Package
```

2.2. Extract the Toolchain

```
tar -xvf gcc-4.8.5-aarch64.tgz
export CROSS_COMPILE=/home/ctione/Documents/JetPack_build/install/bin/aarch64-unknown-linux-gnu-
```

2.3. Extract the kernel sources

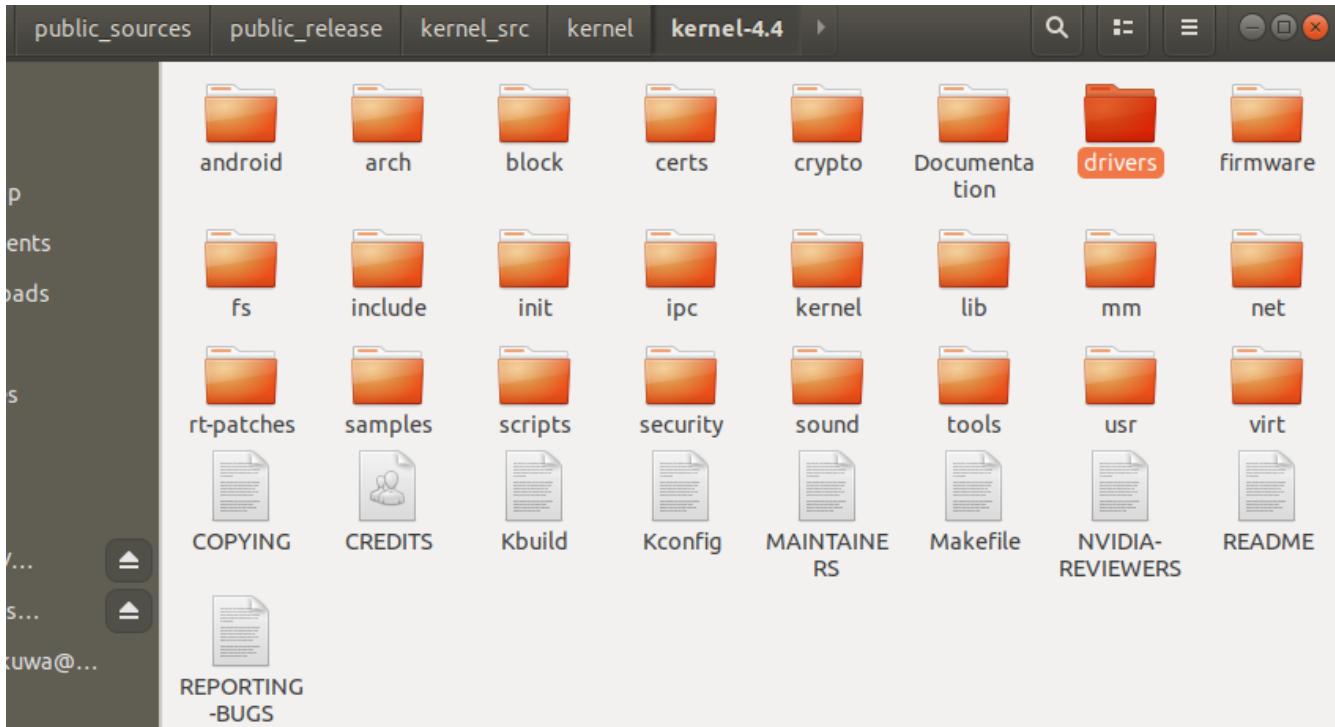
```
tar -xjf public_sources.tbz2
cd ./public_release/
tar -xjf kernel_src.tbz2
```

```
FreeRTOSV8.1.2_src.tbz2
FreeRTOSV8.1.2_src.tbz2.sha1sum
gsteegl_src.tbz2
gsteegl_src.tbz2.sha1sum
gstjpeg_src.tbz2
gstjpeg_src.tbz2.sha1sum
gstomx1_src.tbz2
gstomx1_src.tbz2.sha1sum
kernel_src.tbz2
kernel_src.tbz2.sha1sum
nvgstapps_src.tbz2
nvgstapps_src.tbz2.sha1sum
nvsample_cudaproCESS_src.tbz2
```

```

├── nvsample_cudaprocess_src.tbz2.sha1sum
├── public_sources_sha.txt
└── u-boot_src.tbz2 (To be checked later)
    └── u-boot_src.tbz2.sha1sum

```



2.4. Install the utilities

```

sudo apt install build-essential bc
sudo apt-get install libncurses5-dev
sudo apt-get install pkg-config

```

2.5. Set the shell variable with the command

```

$ TEGRA_KERNEL_OUT=<outdir>
<outdir> is the desired destination for the compiled kernel.

```

```
export TEGRA_KERNEL_OUT=/home/ctione/Documents/JetPack_build/build
```

2.6. Make the build directory

```
mkdir /home/ctione/Documents/JetPack_build/build
```

2.7. Set the system architecture variable

```
export ARCH=arm64
```

2.8. Locate the kernel source directory

```
cd /home/ctione/Documents/JetPack_build/public_release/kernel/kernel-4.4
```

2.9. Create a .config

```
make O=$TEGRA_KERNEL_OUT tegra18_defconfig
```

The above command creates .config file in \$TEGRA_KERNEL_OUT direcotry based from kernel/kernel-4.4/arch/arm64/tegra18_defconfig

Jetson TX1: tegra21_defconfig
Jetson TX2: tegra18_defconfig

3. GUI MenuConfig

Verify or modify .config in /home/ctione/Documents/JetPack_build/build

3.1. Run make menuconfig

In public_release/kernel/kernel-4.4 directory;

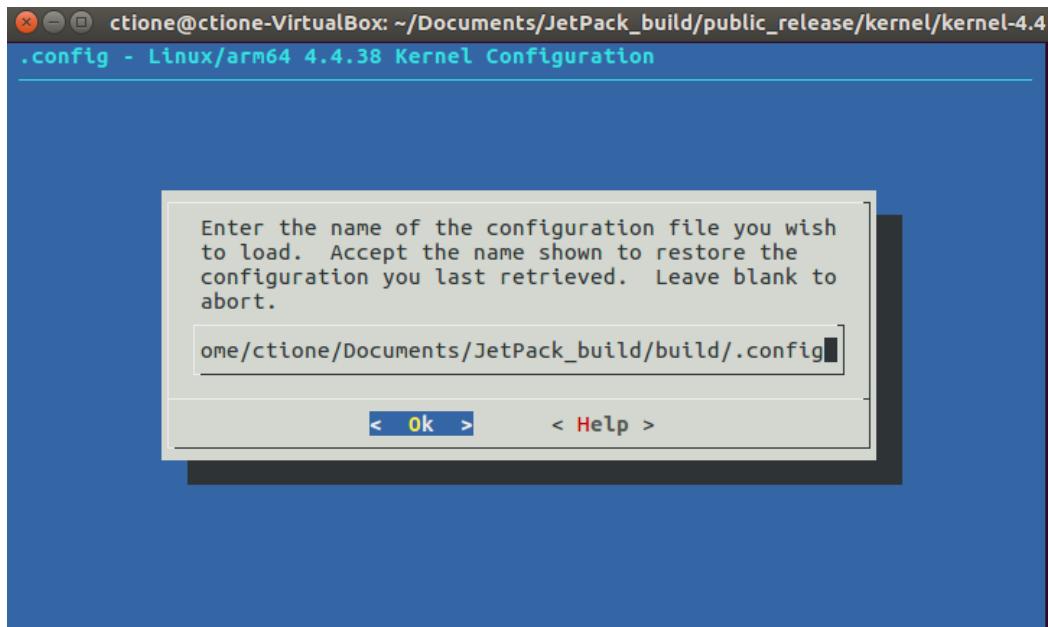
make menuconfig

3.2. Load .config

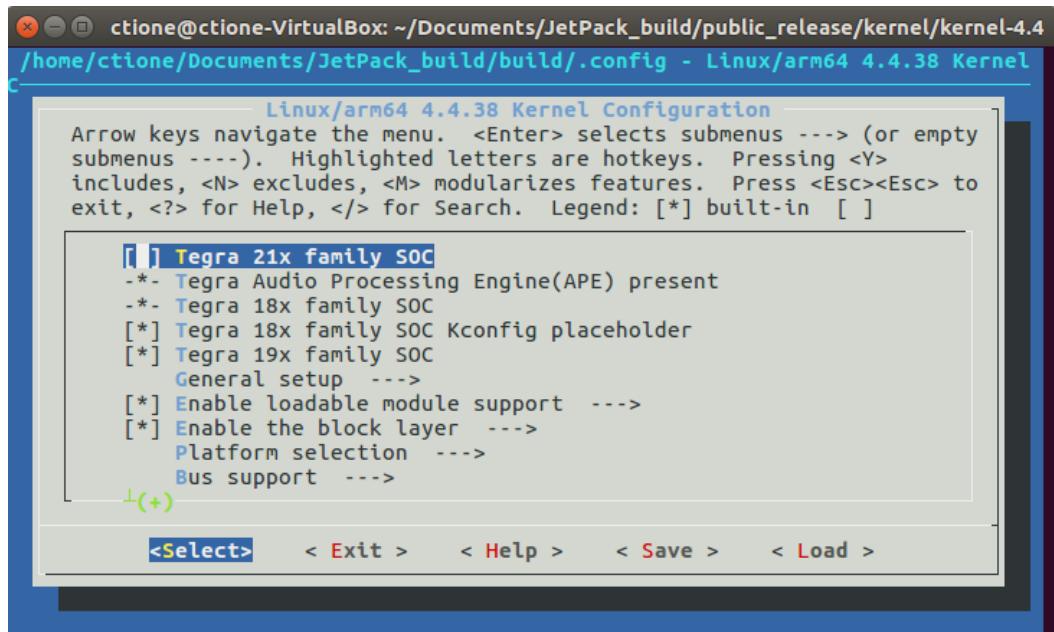
3.2.1 Choose < Load > and push Enter key

3.2.2 Type the target .config file path and push <OK>

/home/ctione/Documents/JetPack_build/build/.config



After loading /home/ctione/Documents/JetPack_build/build/.config



ctione@ctione-VirtualBox: ~/Documents/JetPack_build/public_release/kernel/kernel-4.4
/home/ctione/Documents/JetPack_build/build/.config - Linux/arm64 4.4.38 Kernel

Linux/arm64 4.4.38 Kernel Configuration

Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in []

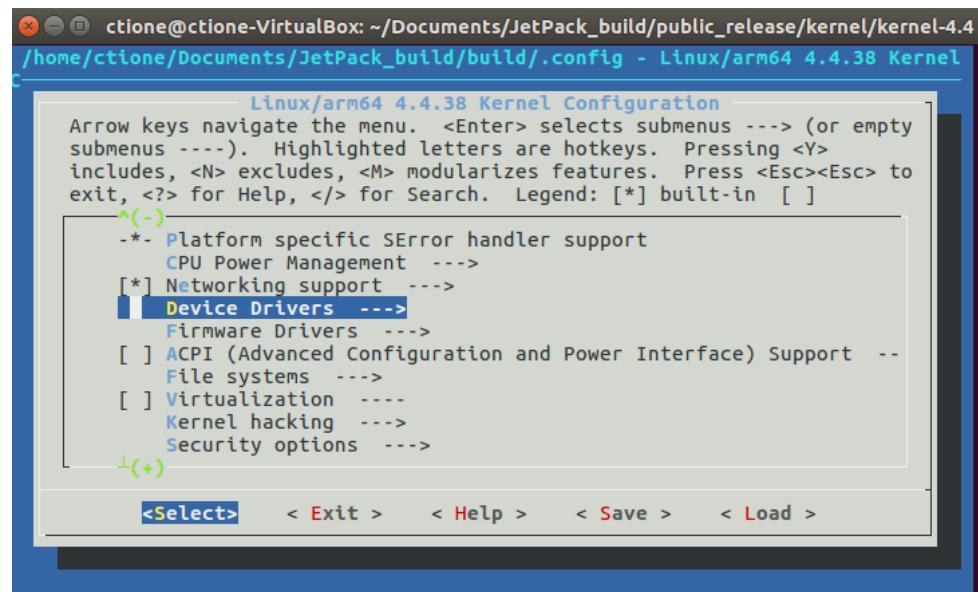
[] **Tegra 21x family SOC**

- *- Tegra Audio Processing Engine(APE) present
- *- Tegra 18x family SOC
- [*] Tegra 18x family SOC Kconfig placeholder
- [*] Tegra 19x family SOC
 - General setup --->
 - [*] Enable loadable module support --->
 - [*] Enable the block layer --->
 - Platform selection --->
 - Bus support --->

↳(+)

<Select> < Exit > < Help > < Save > < Load >

3.3. Modify SPI Driver settings



ctione@ctione-VirtualBox: ~/Documents/JetPack_build/public_release/kernel/kernel-4.4
/home/ctione/Documents/JetPack_build/build/.config - Linux/arm64 4.4.38 Kernel

Linux/arm64 4.4.38 Kernel Configuration

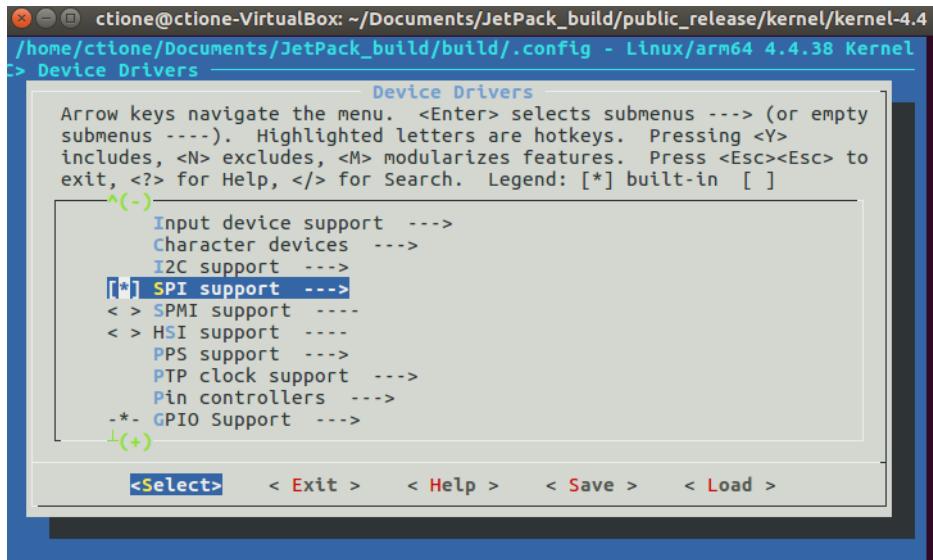
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in []

^(-)

- *- Platform specific SError handler support
- CPU Power Management --->
- [*] Networking support --->
- Device Drivers --->**
- Firmware Drivers --->
- [] ACPI (Advanced Configuration and Power Interface) Support --
 - File systems --->
- [] Virtualization ---->
- Kernel hacking --->
- Security options --->

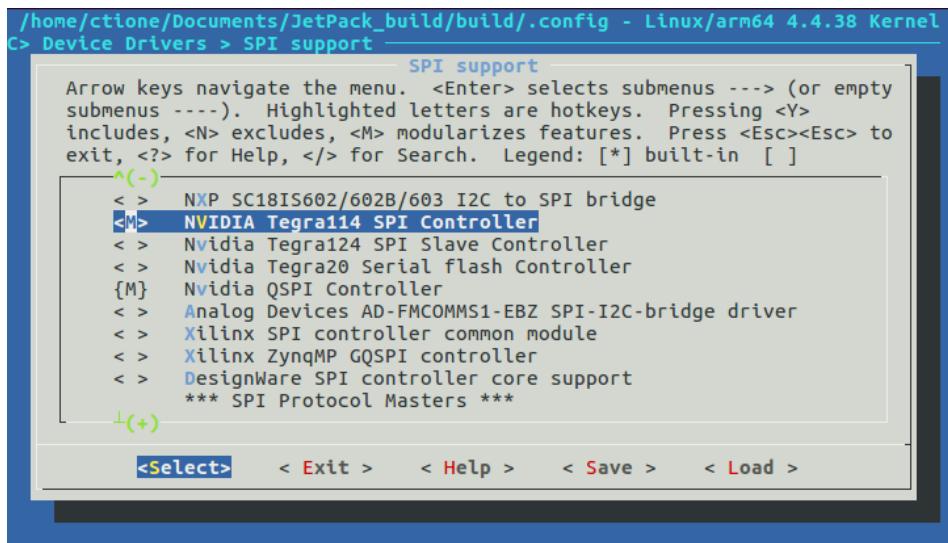
↳(+)

<Select> < Exit > < Help > < Save > < Load >

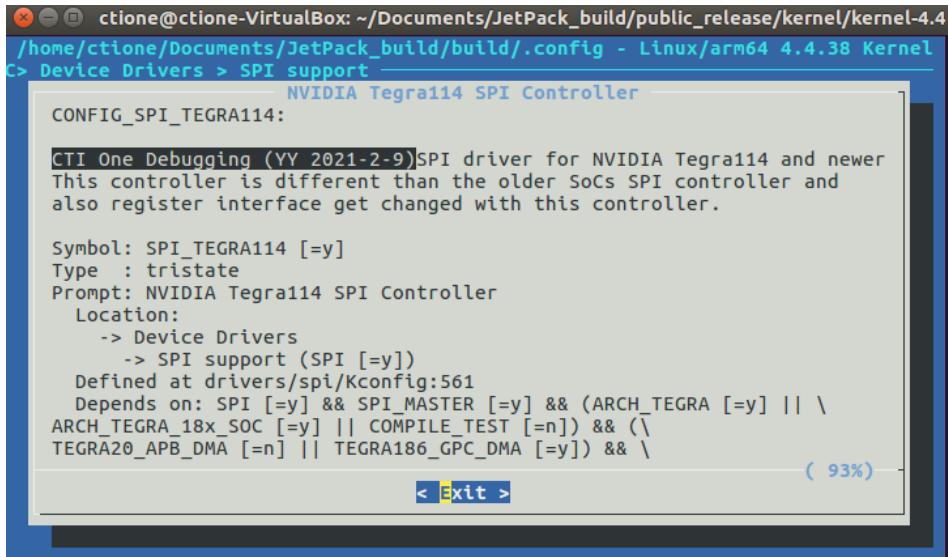


3.3.1 Choose NVIDIA Tegra114 SPI Controller and push M key

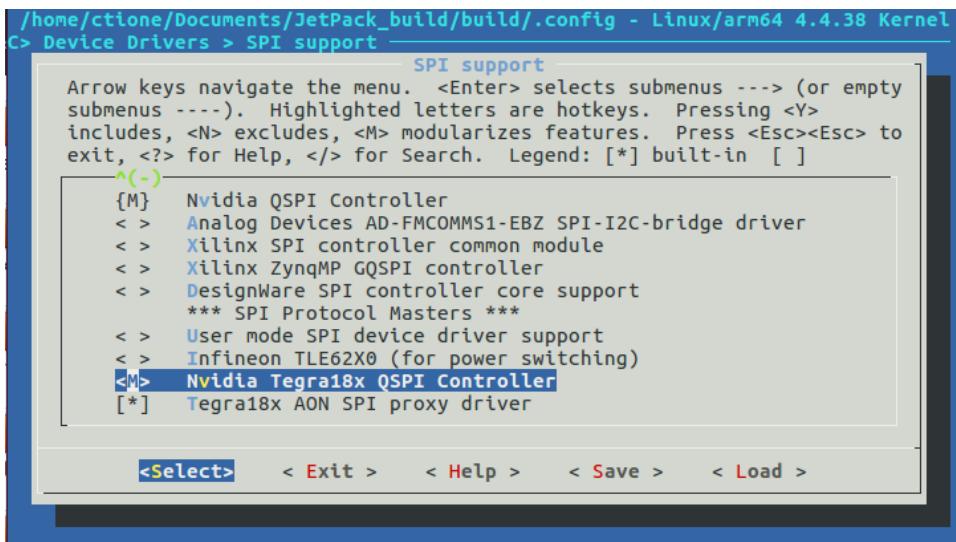
3.3.2 Choose Nvidia QSPI Controller and push M key



Help

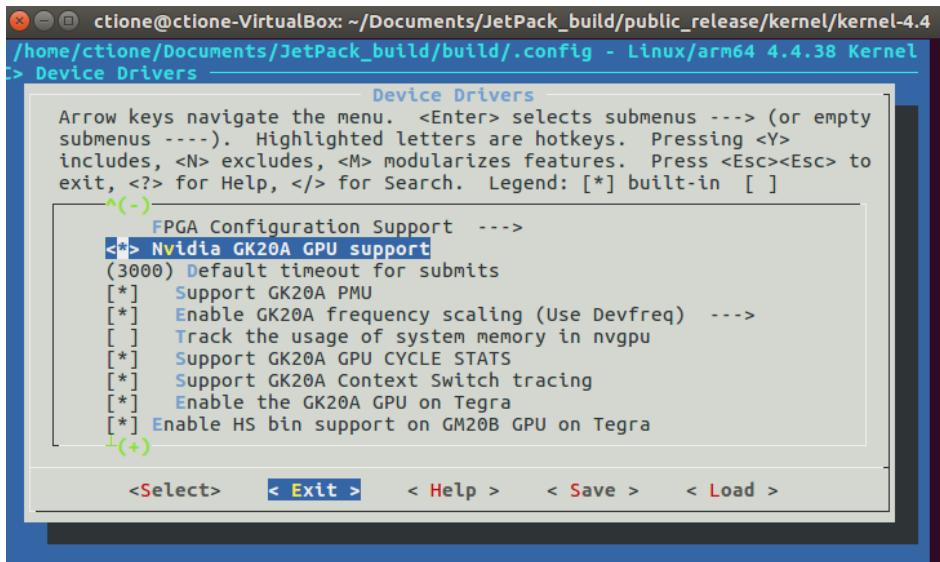


3.3.3 Choose Nvidia Tegra18x QSPI Controller and push M key



The screenshot shows the kernel configuration menu for a Linux/arm64 4.4.38 Kernel. The current section is 'Device Drivers > SPI support'. A sub-menu titled 'SPI support' is open, listing various SPI controllers. The 'Nvidia Tegra18x QSPI Controller' option is highlighted with a blue selection bar. At the bottom of the menu window, there are buttons for '<Select>', '< Exit >', '< Help >', '< Save >', and '< Load >'.

GPU Support is turned on by default



The screenshot shows the kernel configuration menu for a Linux/arm64 4.4.38 Kernel. The current section is 'Device Drivers'. A sub-menu titled 'FPGA Configuration Support' is open, listing options related to the Nvidia GK20A GPU. The 'Nvidia GK20A GPU support' option is highlighted with a blue selection bar. At the bottom of the menu window, there are buttons for '<Select>', '< Exit >', '< Help >', '< Save >', and '< Load >'.

Save and Exit

4. Build the Kernel source

Commands are run in public_release/kernel/kernel-4.4 directory.

(If setting OS variables have not been finished, execute following commands)

```
export CROSS_COMPILE=/home/ctione/Documents/JetPack_build/install/bin/aarch64-unknown-linux-gnu-
export TEGRA_KERNEL_OUT=/home/ctione/Documents/JetPack_build/build
export ARCH=arm64
```

4.1 Change .config to .config-old

```
mv .config .config-old
```

4.2. Build the kernel, all DTBs (Device Tree Blobs) and modules.
make ARCH=arm64 O=\$TEGRA_KERNEL_OUT -j<n>

<n> is the number of CPU cores

4.3. (Optional) Build only the kernel
make O=\$TEGRA_KERNEL_OUT zImage

4.4. (Optional) Build the kernel device tree components
make O=\$TEGRA_KERNEL_OUT dtbs

4.5. (Optional) Build the kernel modules, and install them
make O=\$TEGRA_KERNEL_OUT modules

(Optional Install modules)

make O=\$TEGRA_KERNEL_OUT modules_install INSTALL_MOD_PATH=<your_destination>

(2021-2-9 HL, Nitin, Akshat, Yusuke)

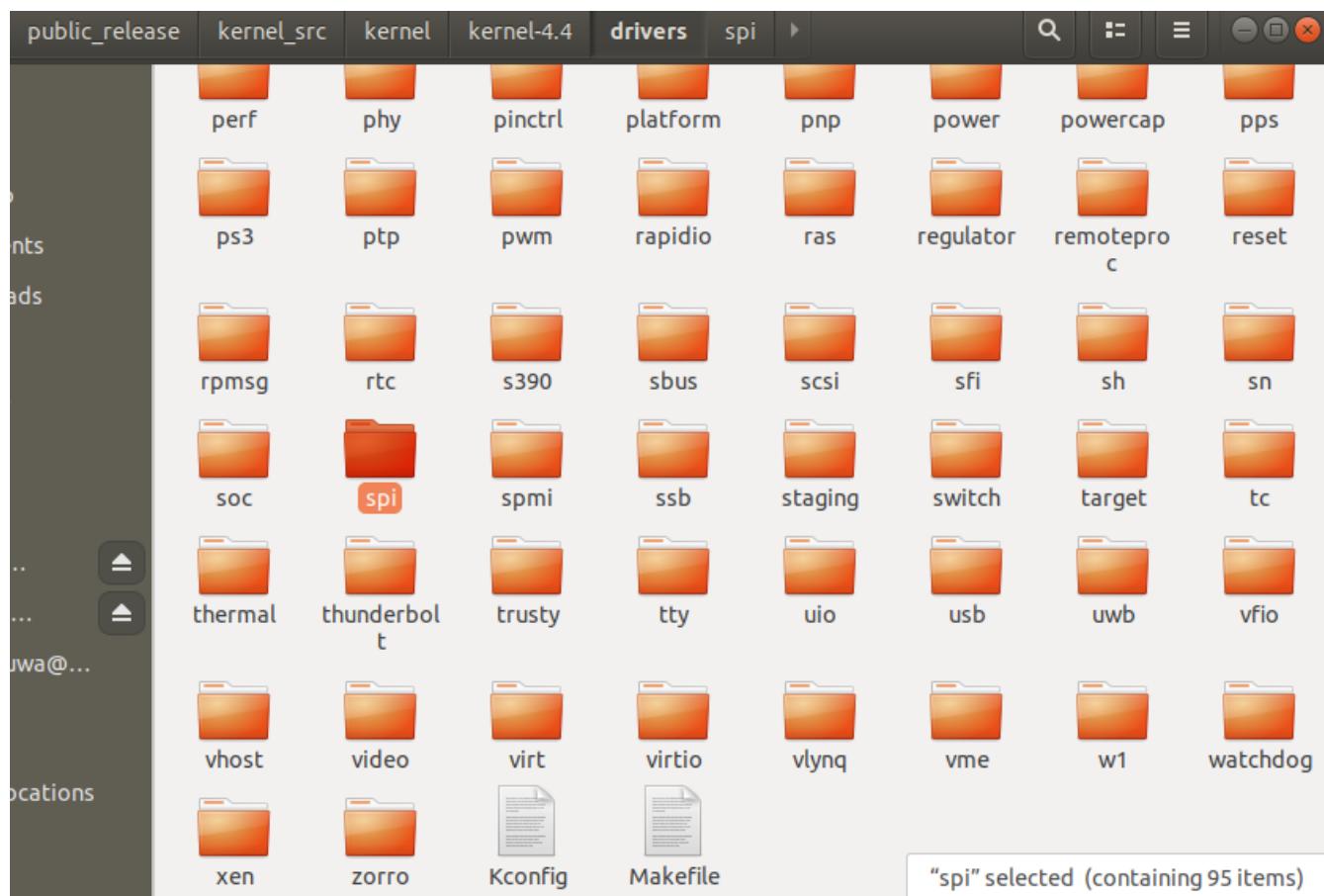
public_release/kernel_src/kernel/kernel-4.4/Kconfig

```
#  
# For a description of the syntax of this configuration file,  
# see Documentation/kbuild/kconfig-language.txt.  
#  
mainmenu "Linux/$ARCH $KERNELVERSION Kernel Configuration"
```

```
config SRCARCH  
    string  
    option env="SRCARCH"
```

source "arch/\$SRCARCH/Kconfig"

public_release/kernel_src/kernel/kernel-4.4/drivers



```
accessibility  
acpi  
amba  
android  
ata  
atm  
auxdisplay  
base
```

```
bcma
block
bluetooth
bus
cdrom
char
clk
clocksource
connector
cpufreq
cpuidle
crypto
dca
devfreq
dio
dma
dma-buf
edac
eisa
extcon
firewire
firmware
fmc
fpga
gpio
gpu
hid
hsi
hv
hwmon
hwspinlock
hwtracing
i2c
ide
idle
iio
infiniband
input
iommu
ipack
irqchip
isdn
Kconfig
leds
lguest
lightnvm
macintosh
mailbox
Makefile
mcb
md
media
memory
memstick
message
mfd
misc
mmc
mtd
```

```
net
nfc
ntb
nubus
nvdimm
nvme
nvmem
nvpmode
of
oprofile
padctrl
parisc
parport
pci
pcmcia
perf
phy
pinctrl
platform
pnp
power
powercap
pps
ps3
ptp
pwm
rapidio
ras
regulator
remoteproc
reset
rpmsg
rtc
s390
sbus
scsi
sfi
sh
sn
soc
spi
spmi
ssb
staging
switch
target
tc
thermal
thunderbolt
trusty
tty
uio
usb
uwb
vfio
vhost
video
virt
virtio
```

```
    └── vlynq
    └── vme
    └── w1
    └── watchdog
    └── xen
    └── zorro
```

public_release/kernel_src/kernel/kernel-4.4/drivers/Kconfig

```
56
57 source "drivers/spi/Kconfig"
58 |
59 source "drivers/i2c/Kconfig"
source "drivers/spi/Kconfig"
source "drivers/spmi/Kconfig"
```

public_release/kernel_src/kernel/kernel-4.4/drivers/spi/

```
    └── Kconfig
    └── Makefile
    ├── spi-adi-v3.c
    ├── spi-altera.c
    ├── spi-ath79.c
    ├── spi-atmel.c
    ├── spi-au1550.c
    ├── spi-bcm2835aux.c
    ├── spi-bcm2835.c
    ├── spi-bcm53xx.c
    ├── spi-bcm53xx.h
    ├── spi-bcm63xx.c
    ├── spi-bcm63xx-hsspi.c
    ├── spi-bfin5xx.c
    ├── spi-bfin-sport.c
    ├── spi-bitbang.c
    ├── spi-bitbang-txrx.h
    ├── spi-butterfly.c
    └── Spi.c
    ├── spi-cadence.c
    ├── spi-clps711x.c
    ├── spi-coldfire-qspi.c
    ├── spi-davinci.c  omap initiative
    ├── spidev.c
    ├── spi-dln2.c
    ├── spi-dw.c
    ├── spi-dw.h
    ├── spi-dw-mid.c
    ├── spi-dw-mmio.c
    ├── spi-dw-pci.c
    ├── spi-efm32.c
    ├── spi-ep93xx.c
    ├── spi-falcon.c
    ├── spi-fsl-cpm.c
    ├── spi-fsl-cpm.h
    ├── spi-fsl-dspi.c
    └── spi-fsl-espi.c
```

```
└── spi-fsl-lib.c
└── spi-fsl-lib.h
└── spi-fsl-spi.c
└── spi-fsl-spi.h
└── spi-gpio.c
└── spi-img-spifi.c
└── spi-imx.c
└── spi-lm70llp.c
└── spi-meson-spifc.c
└── spi-mpc512x-psc.c
└── spi-mpc52xx.c
└── spi-mpc52xx-psc.c
└── spi-mt65xx.c
└── spi-mxs.c
└── spi-nuc900.c
└── spi-octeon.c
└── spi-oc-tiny.c
└── spi-omap-100k.c
└── spi-omap2-mcspi.c
└── spi-omap-uwire.c
└── spi-orion.c
└── spi-pl022.c
└── spi-ppc4xx.c
└── spi-pxa2xx.c
└── spi-pxa2xx-dma.c
└── spi-pxa2xx.h
└── spi-pxa2xx-pci.c
└── spi-qup.c
└── spi-rb4xx.c
└── spi-rockchip.c
└── spi-rspi.c
└── spi-s3c24xx.c Samsung
└── spi-s3c24xx-fiq.h
└── spi-s3c24xx-fiq.S
└── spi-s3c64xx.c Samsung
└── spi-sc18is602.c
└── spi-sh.c
└── spi-sh-hspi.c
└── spi-sh-msiof.c
└── spi-sh-sci.c
└── spi-sirf.c
└── spi-st-ssc4.c
└── spi-sun4i.c
└── spi-sun6i.c
└── spi-tegra114.c
└── spi-tegra124-slave.c
└── spi-tegra20-sflash.c
└── spi-tegra20-slink.c
└── spi-tegra210-qspi.c
└── spi-ti-qspi.c
└── spi-tle62x0.c
└── spi-topcliff-pch.c
└── spi-txx9.c
└── spi-xcomm.c
└── spi-xilinx.c
└── spi-xlp.c
└── spi-xtensa-xtfpga.c
└── spi-zynqmp-gqspi.c
```

public_release/kernel_src/kernel/kernel-4.4/drivers/spi/Kconfig

From line 9 to 11

```
bool "SPI support"
depends on HAS_IOMEM
help
```

(1) Keyword to capture:

```
bool "CTI One Testing"
depends on (SOME KEYWORDS, for example , HAS_IOMEM)
help (This is CTI One AIV100 solution, for further information please contact CTI One directly)
```

(2) syntax for Kconfig

Example from line 31 to 36

```
config SPI_DEBUG
    bool "Debug support for SPI drivers"
    depends on DEBUG_KERNEL
    help
        Say "yes" to enable debug messaging (like dev_dbg and pr_debug),
        sysfs, and debugfs support in SPI controller and protocol drivers.
```

Note:

(2.1) to create a category, use the following

```
config CTI_ONE_SPI # as a example
```

(2.2) the keywords for depends on includes

```
HAS_IOMEM
```

```
DEBUG_KERNEL
```

and so on.

(2.3) add one additional keywords

```
default SPI # line 45
```

(2.4) add additional example, line 57 to 59

```
tristate "Altera SPI Controller"
```

```
select SPI_BITBANG
```

```
help
```

This is the driver for the Altera SPI Controller.

Table 2.3 Kconfig syntax and Keywords table

Keywords	Keyword option (2 nd level)	Note
config	SPI_DEBUG	Example line 31 config SPI_DEBUG
bool		Example line 32 "Debug support for SPI drivers"
depends on	(1)HAS_IOMEM (2)DEBUG_KERNEL (3)ATH79 && GPIOLIB (target hardware) (4)HAS_DMA (5)depends on PARPORT (6)MFD_DLN2	Example line 33 depends on DEBUG_KERNEL Example line 65 depends on ATH79 && GPIOLIB Example line 73 depends on HAS_DMA depends on (ARCH_AT91 AVR32 COMPILE_TEST) Example line 81, BCM2835 is for Pie4 from Broadcom depends on ARCH_BCM2835 COMPILE_TEST

		Example line 165 depends on PARPORT depends on MFD_DL2
help		Example line 34-36 help Say "yes" to enable debug messaging (like dev_dbg and pr_debug), sysfs, and debugfs support in SPI controller and protocol drivers.
default		Example line 45 default SPI
tristate		Example line 56 tristate "Altera SPI Controller"
select		Example line 57 select SPI_BITBANG

Testing and debugging

Line 561-569

Modify:

```
config SPI_TEGRA114
    tristate "NVIDIA Tegra114 SPI Controller"
    depends on (ARCH_TEGRA || ARCH_TEGRA_18x_SOC) || COMPILE_TEST
    depends on (TEGRA20_APB_DMA || TEGRA186_GPC_DMA)
    depends on RESET_CONTROLLER && HAS_DMA
    help
        CTI One Debugging (YY 2021-2-9) SPI driver for NVIDIA Tegra114 and newer SPI Controller interface.
        This controller is different than the older SoCs SPI controller and
        also register interface get changed with this controller.
```

Tasks: To find which one for Tx2? (Line 561 to 605)

```
config SPI_TEGRA114
config SPI_TEGRA124
config SPI_TEGRA20_SFIAHS
config SPI_TEGRA20_SLINK
config QSPI_TEGRA210
config QSPI_TEGRA
```

google it? Or from NVDA developer site to find out which one is for Tx2?

Then select it, build device driver module, will need to use GUI interface, most likely like to invoke it by the command: \$make menuconfig

then upload this module to the target platform Tx2 (hardware is needed at this stage)

(2021-2-11 Yusuke, Akshat)

public_lease/kernel/kernel-4.4/drivers/spi/spi.c

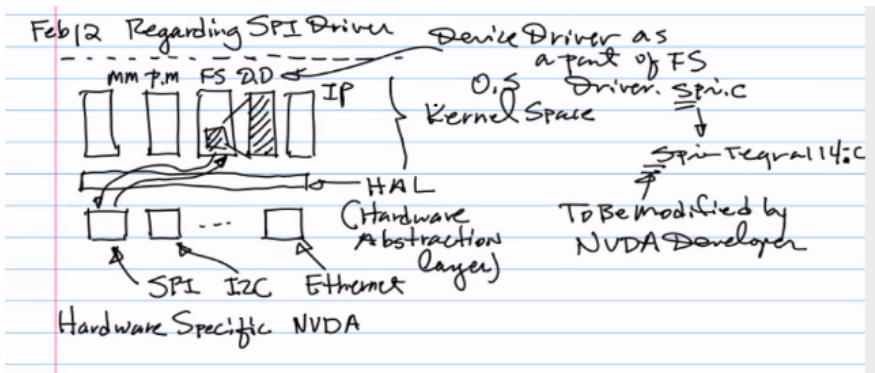
Line 401:

```
/* SPI devices should normally not be created by SPI device drivers; that
 * would make them board-specific. Similarly with SPI master drivers.
 * Device registration normally goes into like arch/.../mach.../board-YYY.c
 * with other readonly (flashable) information about mainboard devices.
 */
```

```
struct boardinfo {
    struct list_head list;
    struct spi_board_info    board_info;
};
```

(2021-2-12 HL, Yusuke, Akshat)

spi.c is general source for SPI, should not be modified for specific boards. Instead of that, spi-tegra114.c can be modified.



Line 2039:

```
int spi_setup(struct spi_device *spi)
```

kprintf can be put in line 2043

```
printf(KERN_ALERT "Hello, world from CTI One\n");
```

```
root# insmod ./hello.ko
Hello, world
root# rmmod hello
Goodbye cruel world
root#
```

(2021-2-15 Akshat)

(2021-2-15 HL, Yusuke)

[10:54, 2/15/2021] Akshat: <https://forums.developer.nvidia.com/t/how-to-enable-spi-spidev-on-28-1-on-target/53999>

[10:55, 2/15/2021] Akshat: pAGE NO. 117

[10:55, 2/15/2021] Akshat: of the datasheet

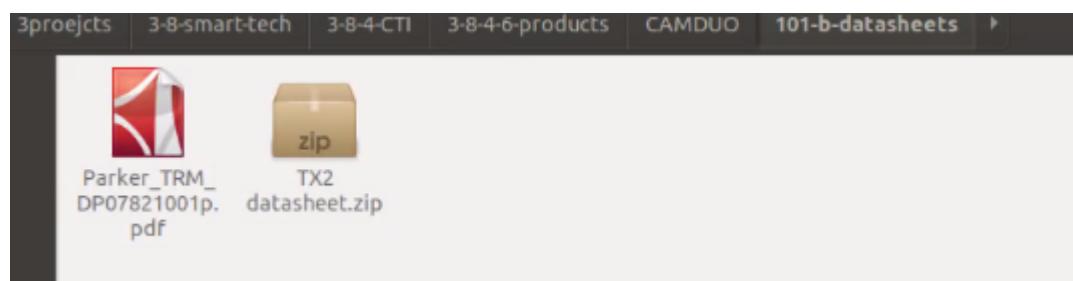
[10:56, 2/15/2021] Akshat: 117 to 133

[10:57, 2/15/2021] Akshat: Now we have to find which module is our TX2 as this is general datasheet for all Parker series

[10:59, 2/15/2021] Akshat: Page 17 is important to understand how to program the registers

[11:00, 2/15/2021] Akshat: Actually SPI special purpose registers are in chapter 39.3

[11:01, 2/15/2021] Akshat: Page 397



[10:54, 2/15/2021] Akshat: <https://forums.developer.nvidia.com/t/how-to-enable-spi-spidev-on-28-1-on-target/53999>

[10:55, 2/15/2021] Akshat: pAGE NO. 117

[10:55, 2/15/2021] Akshat: of the datasheet

[10:56, 2/15/2021] Akshat: 117 to 133

[10:57, 2/15/2021] Akshat: Now we have to find which module is our TX2 as this is general datasheet for all Parker series

[10:59, 2/15/2021] Akshat: Page 17 is important to understand how to program the registers

[11:00, 2/15/2021] Akshat: Actually SPI special purpose registers are in chapter 39.3

[11:01, 2/15/2021] Akshat: Page 397

1.1 Block Diagram

This diagram provides an overview of a Parker series processor.

Figure 1: Parker Processor Block Diagram

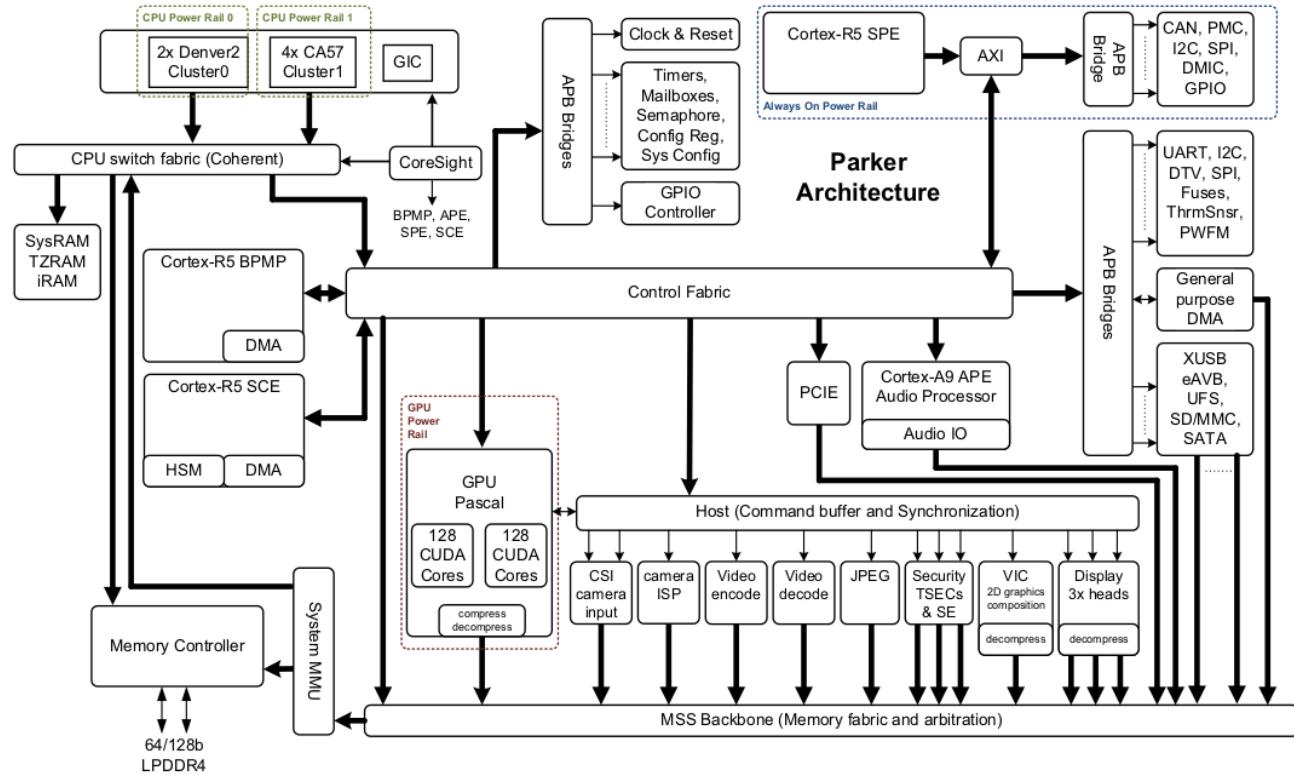
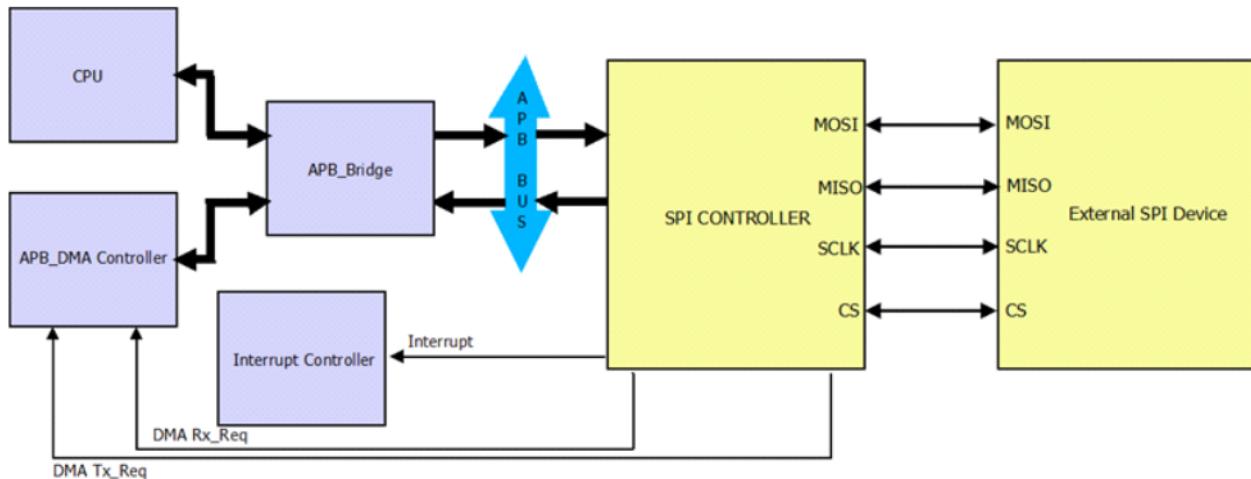


Figure 280: SPI Controller System Overview



Note: from passdown document, find the connectivity table to establish

39.2.1 DMA Mode

This mode is enabled by writing 1 to the DMA bit in the SPI DMA Control register. In this mode, the SPI controller transmits or receives the number of packets as indicated by the BLOCK_SIZE field in the SPI Block Size register.

If the PACKED bit is set and BIT_LEN is set to 7, then all FIFO words contain 4 packets to transfer (transmit or receive). Packets will be transferred as per the En_LE_Bit and En_LE_Byte bit configurations (see the En_LE_Bit and En_LE_Byte Modes section), with packet 0 in byte 0 of the FIFO and packet 3 in byte 3 of the FIFO.

In Unpacked Mode, if BIT_LEN is set to N, each packet will consist of (N + 1) bits. These bits will be transmitted/received in the Tx_FIFO/Rx_FIFO as per the En_LE_Bit and En_LE_Byte bit configurations (see the En_LE_Bit and En_LE_Byte Modes section). Any remaining bits in the FIFO will be ignored by the hardware. The maximum packet length is 32, which can be selected by setting BIT_LEN to 31. In this case, all data bits in the FIFO contain valid packet data.

A DMA request will be generated to GPC_DMA in this mode depending on the setting of Tx_TRIG and Rx_TRIG. If transmits are enabled, setting Tx_TRIG to 00 will generate a Tx DMA request whenever the Tx FIFO has one word of space available (if not full). Setting Tx_TRIG to 01 will generate a Tx DMA request whenever the Tx FIFO has 4 words of space available. If receives are enabled, setting Rx_TRIG to 00 will generate an Rx DMA request whenever the Rx FIFO has one word of data available (is not empty). Setting Rx_TRIG to 01 will generate an Rx DMA request whenever the Rx FIFO has 4 words of data available.

39.3 SPI Controller Registers

Refer to [Section 1.2: Reading Register Tables](#) in the I recommendations for accessing registers.

There are four sets of the SPI controller registers, one the offset of each register within its SPI controller's addressable SPI controller.

39.3.1 SPI_COMMAND_0

SPI Command1 Register

This register is used to set the bit length and to select the transfer mode.

Chip Select can also be selected to be in hardware mode as software mode with both active high and active low polarities to support devices with varying CS polarities.

Offset: 0x0 | Read/Write: R/W | Reset: 0x43d00000 (0b01000011110100000000xxxx000000)

Bit	Reset	Description
31	0x0	PIO: Program 1 after all the other bits in the SPI_COMMAND2 and SPI_COMMAND1 registers are programmed to start the transfer. Hardware clears this bit automatically after the transfer is done. Clearing of this bit by software will stop the shifter and latch the partial data into the buffer. 0 = STOP (default) 1 = PIO

4.4 Cortex-R5 Clusters Sequencer Programming

The default mode out of reset is software async override mode. The switches (AON and APB) and the PLL are controlled through the override settings in the following registers:

PARKER | TRM | DP-07281-001_v1.0p | www.nvidia.com

Page 49:

Table 2: System Address Map

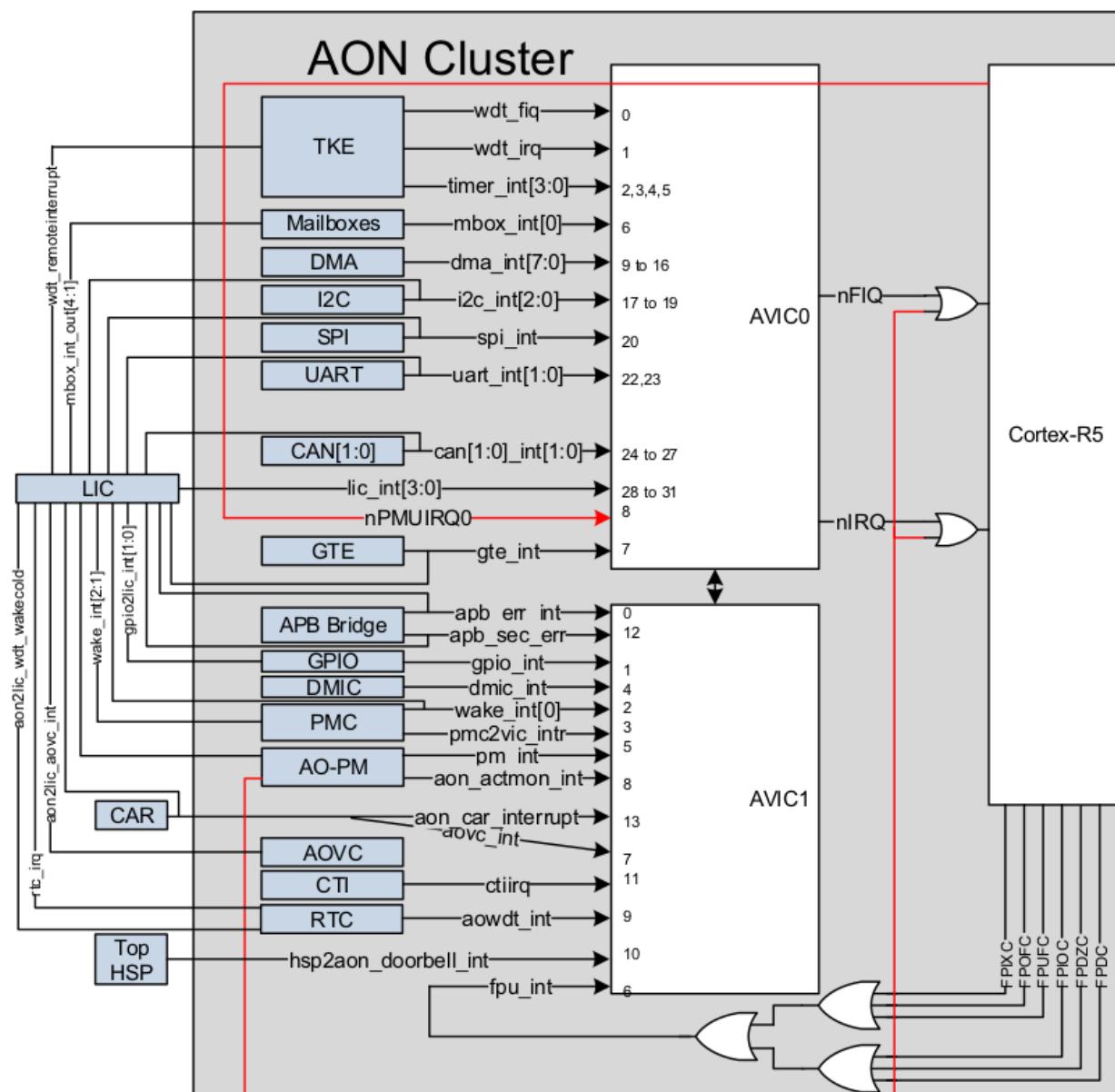
AMAP Aperture Name (Description)	Parent Aperture	Address Start	Address End	Locality	List of headers corresponding to this aperture
SPI1		0x03210000	0x0321ffff		arspi.h
SPI3		0x03230000	0x0323ffff		arspi.h
SPI4		0x03240000	0x0324ffff		arspi.h
QSPI		0x03270000	0x0327ffff		arqspi.h

Page 68:

Table 5: Parker Interrupt Mapping

Source Module	Interrupt Controller	Interrupt Number	Interrupt Name	Interrupt Description
QSPI	LIC	35	QSPI	SPI interrupts; 5 total
SPI1	LIC	36	SPI1	
SPI2	LIC	37	SPI2	
SPI3	LIC	38	SPI3	
SPI4	LIC	39	SPI4	

Figure 4: Interrupt Structure



SPI Connection

https://github.com/Myzhar/Lepton3_Jetson.git
<https://lepton.flir.com/getting-started/raspberry-pi-lepton/>

103-1b-CAMDUO-camera-module-hl-MO-2021-01-12.odp

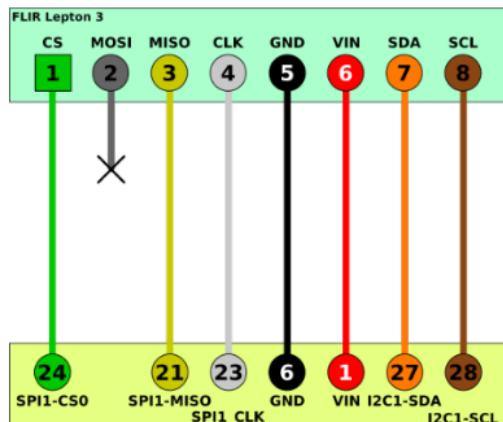


Figure. Tx2 Connect to Lepton

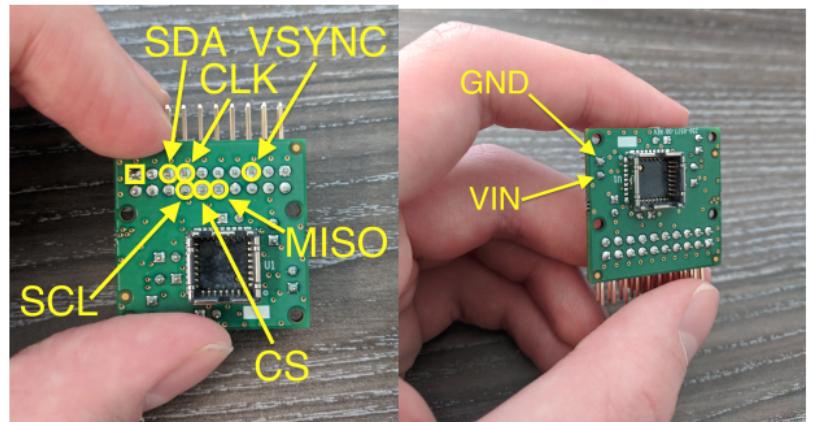
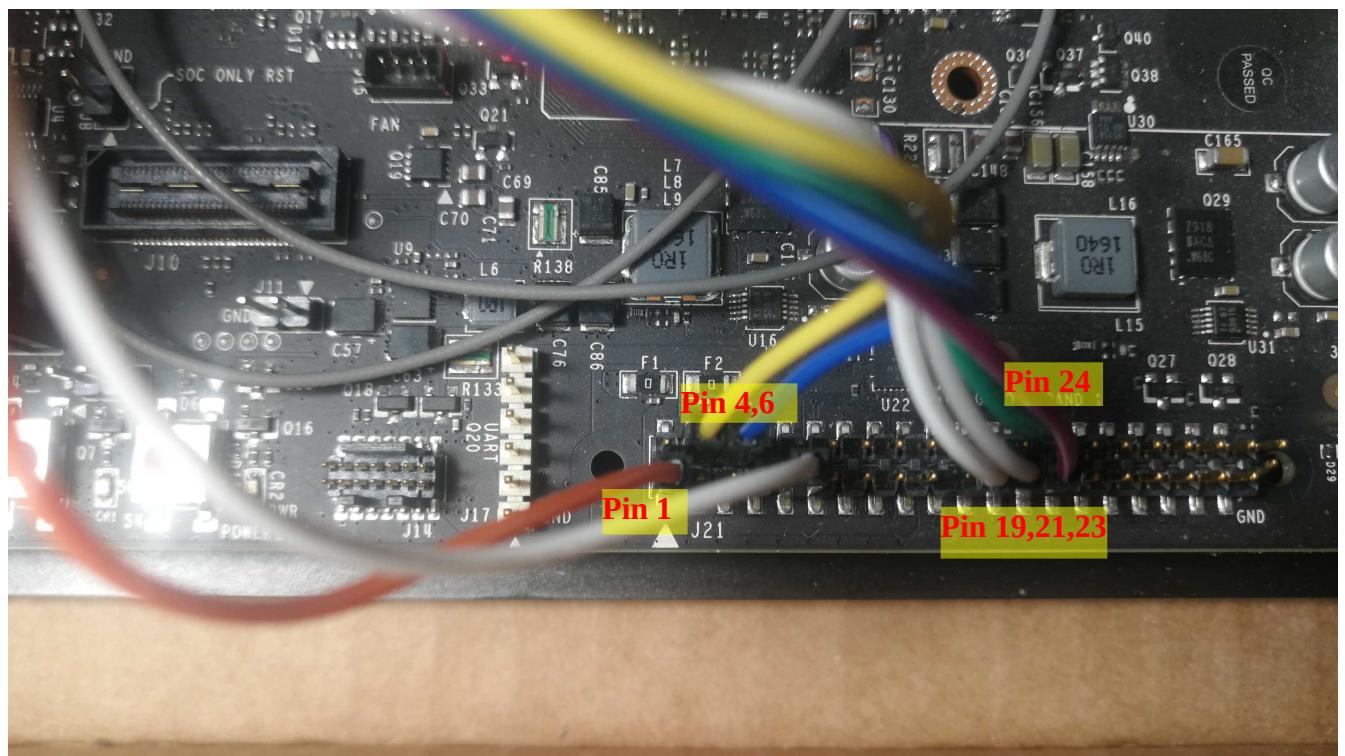
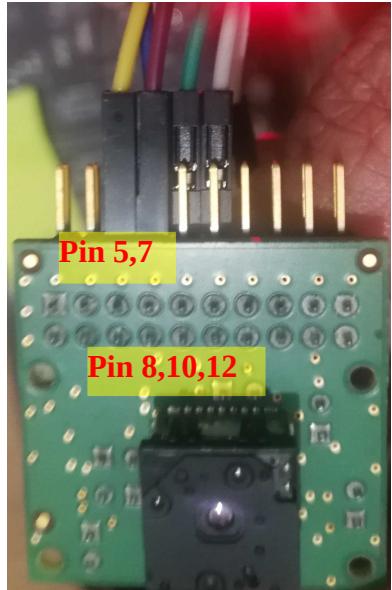


Figure. Lepton IO Pin

TX2 Current Connection



Lepton Current Connection



Front side J2 Pin



Back side J2 Pin



Back side J3 Pin

Jetson TX2 J21 Header						
Sysfs GPIO	Connector Label	Pin	Pin	Connector Label	Sysfs GPIO	
	3.3 VDC Power	1	2	5.0 VDC Power		
	SDA1 General I2C Data 3.3V, I2C Bus 1	3	4	5.0 VDC Power		
	SCL1 General I2C Clock 3.3V, I2C Bus 1	5	6	GND		
gpio390	GPIO_GCLK Audio Master Clock (1.8/3.3V)	7	8	TXD0 UART #0 Transmit		
	GND	9	10	RXD0 UART #0 Receive		
gpio406	GPIO_GEN0 UART #0 Request to Send	11	12	GPIO_GEN1 Audio I2S #0 Clock	gpio392	
gpio397	GPIO_GEN2 Audio Code Interrupt	13	14	GND		
gpio255	GPIO_GEN3 From GPIO Expander (P17)	15	16	GPIO_GEN4 Unused	gpio290	
	3.3 VDC Power	17	18	GPIO_GENS Modem Wake AP GPIO	gpio481	
gpio429	SPI_MOSI SPI #1 Master Out/Slave In	19	20	GND		
gpio428	SPI1_MISO SPI #1 Master In/Slave Out	21	22	GPIO_GEN6 From GPIO Expander (P16)	gpio254	
gpio427	SPI_SCLK SPI #1 Shift Clock	23	24	SPI_CE0_N SPI Chip Select #0	gpio430	
	GND	25	26	SPI_CE1_N SPI #1 Chip Select #1		
	ID_SD General I2C #1 Data (3.3V), I2C Bus 0	27	28	ID_SC General I2C #1 Clock (3.3V), I2C Bus 0		
gpio398	GPIO5 Audio Reset (1.8/3.3V)	29	30	GND		
gpio298	GPIO6 Motion Interrupt (3.3V)	31	32	GPIO12 Unused	gpio297	
gpio389	GPIO13 AP Wake Bt GPIO	33	34	GND		
gpio395	GPIO19 AUDIO I2S #0 Left/Right Clock	35	36	GPIO16 UART #0 Clear to Send	gpio467	
gpio388	GPIO26 (3.3V)	37	38	GPIO20 Audio I2S #0 Data In	gpio394	
	GND	39	40	GPIO21 Audio I2S #0 Data In	gpio393	

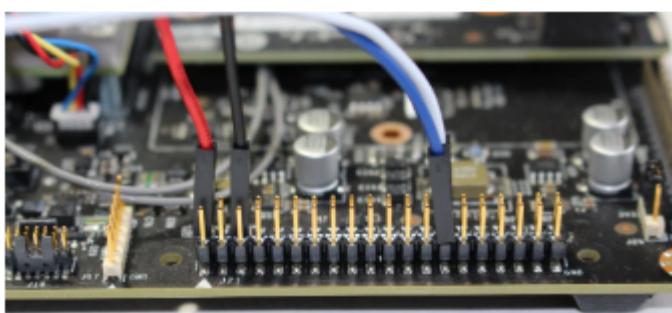


Table ???: Current Cable Connection (2021-2-17 Verified by Yusuke, Nitin, Akshat)

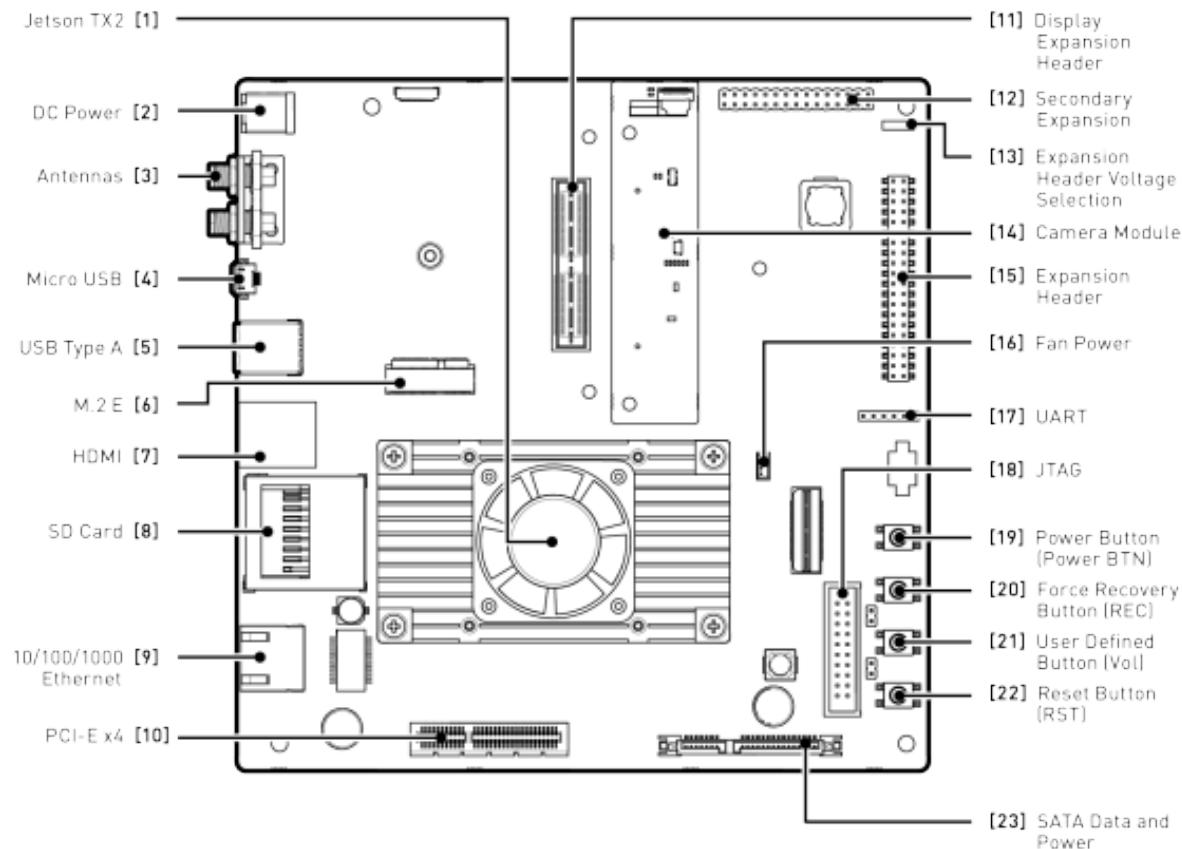
	Cable	TX2	Lepton
1	Orange	Pin 1: 3.3 VDC	(J3 Pin) Pin 2: VIN
2	Yellow	Pin 4: 5.0 VDC -> Should be Pin 27: ID_SD (I2C #1 Data (3.3V), I2C Bus 0)	(J2 Pin) Pin 5: SDA
3	Blue	Pin 6: GND -> Should be Pin 28: ID_SC (I2C #1 Clock (3.3V), I2C Bus 0)	(J2 Pin) Pin 8: SCL
4	White 1	Pin 9: GND	(J3 Pin) Pin 1: GND
5	White 2	Pin 19: SPI_MOSI (Master Out/Slave In)	Not Connected
6	Gray	Pin 21: SPI_MISO (Master In/Slave Out)	(J2 Pin) Pin 12: MISO
7	Brown	Pin 23: SPI_SCLK (SPI Shift Clock)	(J2 Pin) Pin 7: CLK
8	Green	Pin 24: SPI_CE0_N (SPI Chip Select #0)	(J2 Pin) Pin 10: CS

* For connecting to PINOD, (J2 Pin) Pin 15 is connected to VSYNC, GPIO 17 on PINOD.

(2021-2-17 Yusuke)

Flash Kernel

Jetson TX2 Reference Board Layout



To determine the success of a driver update

Execute the following command on a booted target device:

```
shalsum -c /etc/nv_tegra_release
```

1Jetson TX2 Module

- [1.1Processing Components](#)
- [1.2Ports & Peripherals](#)
- [1.3Form-Factor](#)
- [1.4Software Support](#)
- [1.5Jetson TX2i Module](#)
- [2Jetson TX2 Developer Kit](#)

Reference

L4T: Linux for Tegra R28.2.1

<https://developer.nvidia.com/embedded/linux-tegra-r2821>

Jetson Download Center

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

The thread about Toolchain(Cross-Compiler) of Jetson TX2 on NVIDIA developers forum

<https://forums.developer.nvidia.com/t/toolchain-cross-compiler-of-jetson-tx2/56059/5>