



WS-JAX-DEV

Development System

USER GUIDE

DOCUMENT CHANGE HISTORY

Version	Date	Authors	Description of Change
1.0	DEC 6, 2021		Initial release

Welcome! There are two key things you should do right away:

0. Sign up for the [NVIDIA Developer Program](#) – this enables you to ask questions and contribute on the [NVIDIA Jetson Forums](#), gives access to all documentation and collateral on the [Jetson Download Center](#), and more.
1. Read this User Guide! After that, check out these important links:
 - [Jetson FAQ](#) – Please read the FAQ.
 - [Support Resources](#) – This web page links to important resources, including the Jetson Forum and the Jetson Ecosystem page.
 - [NVIDIA Jetson Linux Driver Package Release Notes](#) – Jetson Linux Driver Package is a key component of the Jetson platform, and provides the sample filesystem for your Development System. Please read the latest release notes.

Thanks,
Wisdom Starry

目录

DOCUMENT CHANGE HISTORY.....	2
INCLUDED IN THE BOX.....	1
1.1 Development System INTERFACES.....	2
1.1.1 Parameter table.....	3
*Expansion capability.....	3
1.2 Description of interface.....	3
1.2.1 HDMI -TYPE-A.....	3
1.2.2 USB3.0.....	4
1.2.3 Micro USB.....	4
1.2.4 Gigabit Ethernet.....	4
1.2.5 12V POWER.....	5
1.2.6 PWM FAN.....	5
1.2.7 S1 Button.....	5
1.2.8 S2 Button.....	5
1.2.9 J9 Extended 20PIN (CON_1.27mm_2x10P_SMT)	5
1.3.0 J19 Extended 8PIN.....	5
1.3.1 J10 Extended 20PIN (CON_1.27mm_2x10P_SMT)	6
2.1 Prepare.....	8
2.2 Flashing.....	8
2.3 flashing custom image.....	9
2.4 verification.....	10

GETTING STARTED

The Development System provides a full-featured development platform designed to get you up and running NVIDIA JETSON quickly. The included carrier board exposes many standard hardware interfaces, enabling a highly flexible and extensible platform for rapid prototyping.

NVIDIA JetPack™SDK supports both your Development System and host development platform. It includes:

- Sample Linux filesystem with NVIDIA drivers
- AI and Computer Vision libraries and APIs
- Developer tools
- Documentation and sample code

Before using your Development System, you must install JetPack. A Linux host computer is required; for details, see [How to Install JetPack](#), below.

Minimum system requirements for the host computer are:

- Ubuntu Linux x64 v18.04
- A valid Internet connection
- At least 23GB of disk space

INCLUDED IN THE BOX

The Development System includes:

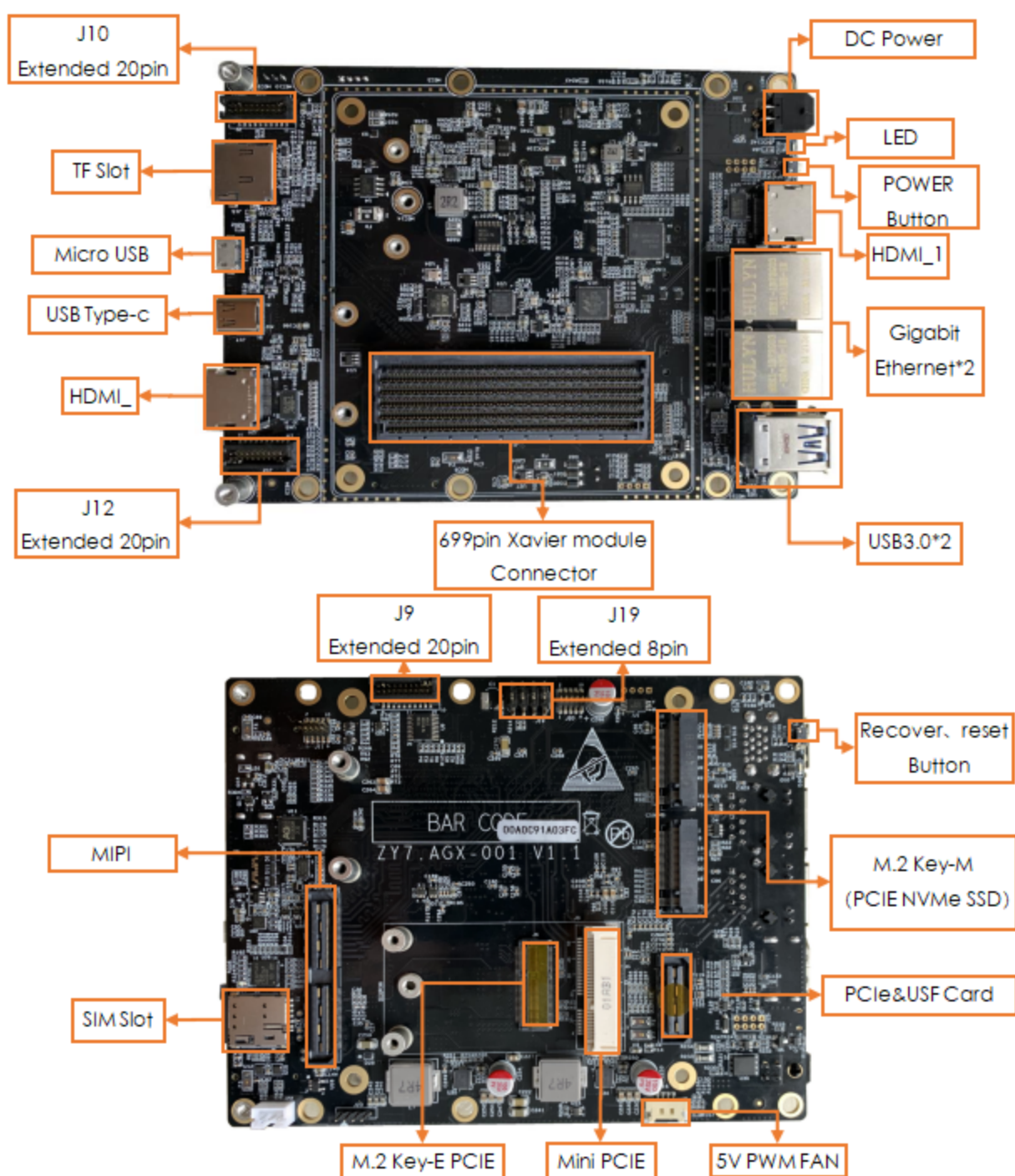
- NVIDIA Jetson AGX Xavier™ module (P2888) with thermal solution
- Reference carrier board (WS-MDN-601)
- Power supply with 19V4.5A AC cord

1.1 Development System INTERFACES

front view(left 45°) and Bottom view (bottom)



Top view of Development System carrier board



1.1.1 Parameter table

Processor module	NVIDIA Jetson AGX Xavier™
AI Performance	32 TOPs
OS	UBUNTU 18.04/20.04
CPU	8-core ARM v8.2 64-bit CPU, 8MB L2 + 4MB L3
GPU	512-core NVIDIA Volta™ GPU with 64 Tensor Cores
Memory	32GB 256-Bit LPDDR4x 137GB/s
Storage	32GB eMMC 5.1 2-M.2 M-key NVME 2280 SSD expansion*
Power	12V DC, 10W-30W
Display	HDMI2.0 (up to 6Gbps)
USB	USB3.0 x 2, USB-TYPE-C, Micro USB
NETWORK	Gigabit Lan x 2, 4G-LTE*, WIFI*
Interfaces	mPCIe, UART, GPIO, I2C, CAN 5V-PWM-FAN
Mechanical	carrier board 144x110mm, Jetson AGX Xavier™ module 100mm x 87mm Radiator 127g*
Temperature Range	carrier board -40°C-85°C Jetson AGX Xavier™ module -25°C-80°C*

*Expansion capability

1.2 Description of interface

1.2.1 HDMI-TYPE-A

PIN	Signal describe	PIN	Signal describe
1	TMDS Data2+	2	TMDS Data2 GND
3	TMDS Data2-	4	TMDS Data1+
5	TMDS Data1 GND	6	TMDS Data1-
7	TMDS Data0+	8	TMDS Data0 GND
9	TMDS Data0-	10	TMDS Clock+
11	TMDS Clock GND	12	TMDS Clock-
13	CEC	14	NC
15	DDC clock	16	DDC data
17	DDC GND	18	+5V
19	Hot Plug Detect		

1.2.2 USB3.0

PIN	Signal describe	PIN	Signal describe
1	VBUS	2	USB 2.0 D-
3	USB 2.0 D+	4	GND
5	SSRX-	6	SSRX+
7	GND	8	SSTX-
9	SSTX+		

1.2.3 Micro USB

PIN	Signal describe	PIN	Signal describe
1	VBUS	2	USB 2.0 D-
3	USB 2.0 D+	4	USB ID
5	GND		

NOTE: Debug Serial port, 编号为/dev/ttyTCU0

#Install debugging tool

```
sudo apt-get install cutecom
```

#Baud rate 115200,data bits 8,stop bits 1.

```
sudo cutecom
```

1.2.4 Gigabit Ethernet

PIN	Signal describe	PIN	Signal describe
1	TP0+	2	TP0-
3	TP1+	4	TP2+
5	TP2-	6	TP1-
7	TP3+	8	TP3-

1.2.5 12V POWER

PIN	Signal describe	PIN	Signal describe
1	VDD_12V	2	GND
3	VDD_12V	4	GND
Range: +9-19V			

1.2.6 PWM FAN

PIN	Signal describe	PIN	Signal describe
1	GND	2	+5V
3	FAN_TACH_CON	4	FAN_PWM

1.2.7 S1 Button

RECOVER Button: Press and hold 2-3 second to enter RECOVER mode for flash.

1.2.8 S2 Button

POWER Button: Switch POWER ON/OFF.

1.2.9 J9 Extended 20PIN (CON_1.27mm_2x10P_SMT)

PIN	Signal describe	PIN	Signal describe
1	AUDIO_MCLK01	2	I2S1_CLK
3	I2S1_LRCLK	4	I2S1_SDOUT
5	I2S1_SDIN	6	I2C_GP2_DAT_3V3
7	I2C_GP2_CLK_3V3	8	MCLK05_3V3
9	I2S2_FS_3V3	10	VDD_3V3
11	SPI2_CLK_UART7	12	SPI2_MISO_UART7
13	SPI2_MOSI_UART7	14	SPI2_CSI_UART8
15	GPIO11_CODEC_INT	16	GPIO17_40HEADER_3V3
17	I2S2_SDIN_3V3	18	I2S2_SDOUT_3V3
19	I2S2_CLK_3V3	20	GND

1.3.0 J19 Extended 8PIN

PIN	Signal describe	PIN	Signal describe
1	GND	2	FORCE_RECOVERY
3	SYS_RST	4	BUTTON_POWER_ON
5	SHORT:AOUT POWER ON ENABLE	6	OPEN:AOUT POWER ON DISABLE
7	CVB_STBY: LED	8	BBAT: Battery

1.3.1 J10 Extended 20PIN (CON_1.27mm_2x10P_SMT)

PIN	Signal describe	PIN	Signal describe
1	USB22_D_P	2	USB22_D_N
3	I2C_GP2_DAT_3V3	4	I2C_GP2_CLK_3V3
5	I2C_GP5_DAT_3V3	6	I2C_GP5_CLK_3V3
7	GPIO27_PWM2_3V3	8	BBAT
9	VDD_5V5	10	VDD_5V5
11	CAN0_DOUT	12	CAN0_DIN
13	CAN1_DOUT	14	CAN1_DIN
15	-	16	GPIO7_CAN0_WAKE
17	GPIO8_AO_DMIC_IN_DAT	18	GPIO9_CAN1_GPIO0
19	GND	20	GND

1.3.1.1 GPIO

PIN	Signal describe	PIN	Signal describe
7	GPIO27_PWM2_3V3	16	GPIO7_CAN0_WAKE
17	GPIO8_AO_DMIC_IN_DAT	18	GPIO9_CAN1_GPIO0

NOTE:

GPIO27_PWM2_3V3 is GPIO393

GPIO7_CAN0_WAKE is GPIO254

GPIO8_AO_DMIC_IN_DAT is GPIO256

GPIO9_CAN1_GPIO0 is GPIO257

GPIO Test Steps:

```
#list gpio
```

```
cd /sys/class/gpio
```

```
#enable gpio
```

```
echo '293'|sudo tee /sys/class/gpio/export
```

```
echo '254'|sudo tee /sys/class/gpio/export
```

```
echo '256'|sudo tee /sys/class/gpio/export
```

```
echo '257'|sudo tee /sys/class/gpio/export
```

```
#set gpio output direction
```

```
cd gpio393
```

```
echo 'out'|sudo tee /sys/class/gpio/gpio393/direction
```

```
#set gpio High level=3.3v
```

```
echo '1'|sudo tee /sys/class/gpio/gpio393/value
```

```
#gpio Low level=GND
```

```
echo '0'|sudo tee /sys/class/gpio/gpio393/value
```

```
#set gpioinput direction,cd /sys/class/gpio
```

```
#Check gpio value , set 0
```

```
cat ./gpio393/value (display 0)
```

```
echo 'in'|sudo tee /sys/class/gpio/gpio393/direction
```

```
#If high level connected, return 1
```

```
cat /sys/class/gpio/gpio393/value
```

1.3.1.2 CAN

PIN	Signal describe	PIN	Signal describe
11	CAN0_DOUT	12	CAN0_DIN
13	CAN1_DOUT	14	CAN1_DIN

NOTE:

#Install debugging tool

```
sudo apt-get install can-utils
```

#Debug can (For server end)

```
sudo -S modprobe can
```

```
sudo modprobe can_raw
```

```
sudo modprobe mttcan
```

```
sudo ip link set can0 type can bitrate 500000
```

```
sudo ip link set up can0
```

```
candump can0
```

(For client end)

```
cansend can0 1F334455#1122334455667788
```

1.3.2 J12 Extended 20PIN (CON_1.27mm_2x10P_SMT)

PIN	Signal describe	PIN	Signal describe
1	UART3_TX_DEBUG	2	UART3_RX_DEBUG
3	UART2_RX_422	4	UART1_RX_232
5	UART2_RX_232	6	UART5_RX
7	UART5_TX	8	VDD_1V8
9	VDD_3V3	10	VDD_3V3
11	UART1_TX_232	12	UART2_TX_232
13	UART1_RX_422	14	SEL3*
15	SEL2*	16	SEL1*
17	PWM01_3V3	18	GPIO27_PWM02_3V3
19	GND	20	GND

1.3.2.1 UART1

PIN	Signal describe	PIN	Signal describe
4	UART1_RX_232	11	UART1_TX_232

NOTE: /dev/ttyTHS0

1.3.2.2 UART2

PIN	Signal describe	PIN	Signal describe
3	UART2_RX_232	12	UART2_TX_232

NOTE: /dev/ttyTSH1

1.3.2.3 UART3

PIN	Signal describe	PIN	Signal describe
1	UART3_TX_232	2	UART3_RX_232

NOTE: /dev/ttyTCU0

1.3.2.4 UART5

PIN	Signal describe	PIN	Signal describe
6	UART5_RX_232	7	UART5_TX_232

NOTE: /dev/ttyTSH4

2、System flashing

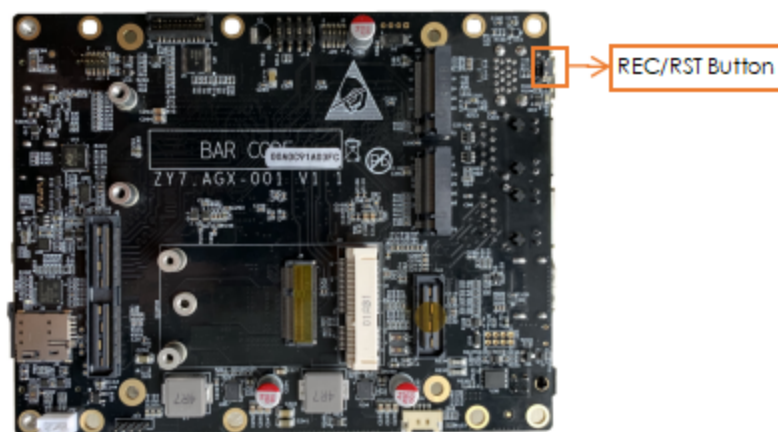
2.1 Prepare

Download Image compression package

[Jetson Download Center.](#)

2.2 Flashing

- (1) connect DC power supply;
- (2) Press and hold the rec / RST key for about 3S, and release the rec / RST key after seeing that the light of the carrier flashes.



- (3) To determine whether or not to enter the recovery mode successfully, you can use the lsusb command to check whether there are "NVIDIA Corp" devices.

```
hcg@ubuntu:~$ lsusb
Bus 001 Device 004: ID 0955:7c18 Nvidia Corp.
Bus 001 Device 001: ID 1d0b:0002 Linux Foundation 2.0 root hub
```

如图所示即表示已进入 recovery 模式

(4) following to install.sh The text description when the run is completed, or the readme.txt The introduction of the document flash.sh Burning operation. For example: burning records

```
sudo ./flash.sh jetson-agx-xavier-devkit mmcblk0p1
```

```
[ 1824.4696 ] Writing partition kernel with boot_sigheader.img.encrypt
[ 1824.4762 ] [.....] 100%
[ 1824.4994 ] Writing partition kernel_b with boot_sigheader.img.encrypt
[ 1824.5098 ] [.....] 100%
[ 1824.5342 ] Writing partition kernel-dtb with tegra186-ws-tx2001-10-hdmi-2usb_sigheader.dtb.encrypt
[ 1824.5526 ] [.....] 100%
[ 1824.5711 ] Writing partition kernel-dtb_b with tegra186-ws-tx2001-10-hdmi-2usb_sigheader.dtb.encrypt
[ 1824.5810 ] [.....] 100%
[ 1824.6844 ]
[ 1824.6871 ] tegradevflash_v2 --write BCT br_bct_BR.bct
[ 1824.6893 ] Bootloader version 01.00.0000
[ 1824.6121 ] Writing partition BCT with br_bct_BR.bct
[ 1824.6131 ] [.....] 100%
[ 1824.6707 ]
[ 1824.6852 ] tegradevflash_v2 --write MB1_BCT mb1_cold_boot_bct_MB1_sigheader.bct.encrypt
[ 1824.6870 ] Bootloader version 01.00.0000
[ 1824.6899 ] Writing partition MB1_BCT with mb1_cold_boot_bct_MB1_sigheader.bct.encrypt
[ 1824.6908 ] [.....] 100%
[ 1824.7591 ]
[ 1824.7618 ] tegradevflash_v2 --write MB1_BCT_b mb1_cold_boot_bct_MB1_sigheader.bct.encrypt
[ 1824.7642 ] Bootloader version 01.00.0000
[ 1824.7679 ] Writing partition MB1_BCT_b with mb1_cold_boot_bct_MB1_sigheader.bct.encrypt
[ 1824.7712 ] [.....] 100%
[ 1824.8255 ]
[ 1824.8256 ] Flashing completed

[ 1824.8257 ] Coldbooting the device
[ 1824.8283 ] tegradevflash_v2 --reboot coldboot
[ 1824.8307 ] Bootloader version 01.00.0000
[ 1824.8378 ]
*** The target t186ref has been flashed successfully. ***
Reset the board to boot from internal eMMC.
```

2.3 flashing custom image

2.3.1 Backup custom image :

```
sudo ./flash.sh -r -k APP -G backup.img jetson-agx-xavier-devkit mmcblk0p1
```

Generated backup.img.raw Image file of (recommended to be compressed to zip file storage)

2.3.2 Restore custom image :

Copy the image of to Linux_for_Tegra / bootloader / directory and rename to system.img

```
sudo mv backup.img.raw bootloader/system.img
```

```
cd Linux_for_Tegra/bootloader/
```

```
sudo ./flash.sh -r jetson-agx-xavier-devkit mmcblk0p1
```

Note: -r parameter refers to using the system.img to flash.

Note: both backup and recovery must be in the mode of receiver.

2.4 verification

Connect the HDMI interface display to power on verification, and the image burning is successful when the system is star.

