

WS-MDN-501使用说明



目录

一、产品介绍

1.1 产品概述.....	1
1.2 产品功能示意图.....	2
1.3 接口描述.....	3
1.4 24针 多功能接口.....	5
1.4.1多功能引脚定义.....	5
1.4.2功能测试	
1.4.2.1 URAT0.....	5
1.4.2.2 URAT1.....	5
1.4.2.3 CAN0.....	6
1.4.2.4 GPIO.....	7
1.4.2.5 I ² C.....	7

二、系统烧录

2.1 烧录准备.....	8
2.2 烧录过程.....	8
2.3 烧录自己的镜像.....	9
2.4 烧录完成.....	10

三、订货信息.....	11
-------------	----

产品介绍:

1.1 产品概述



WS-MDN-501板卡主要采用工业元器件, 本产品尺寸小, 温度范围-40℃-85℃。NVIDIA Jetson NANO/Xavier NX 底板, 支持 NANO, XAVIER NX 系列核心模块。

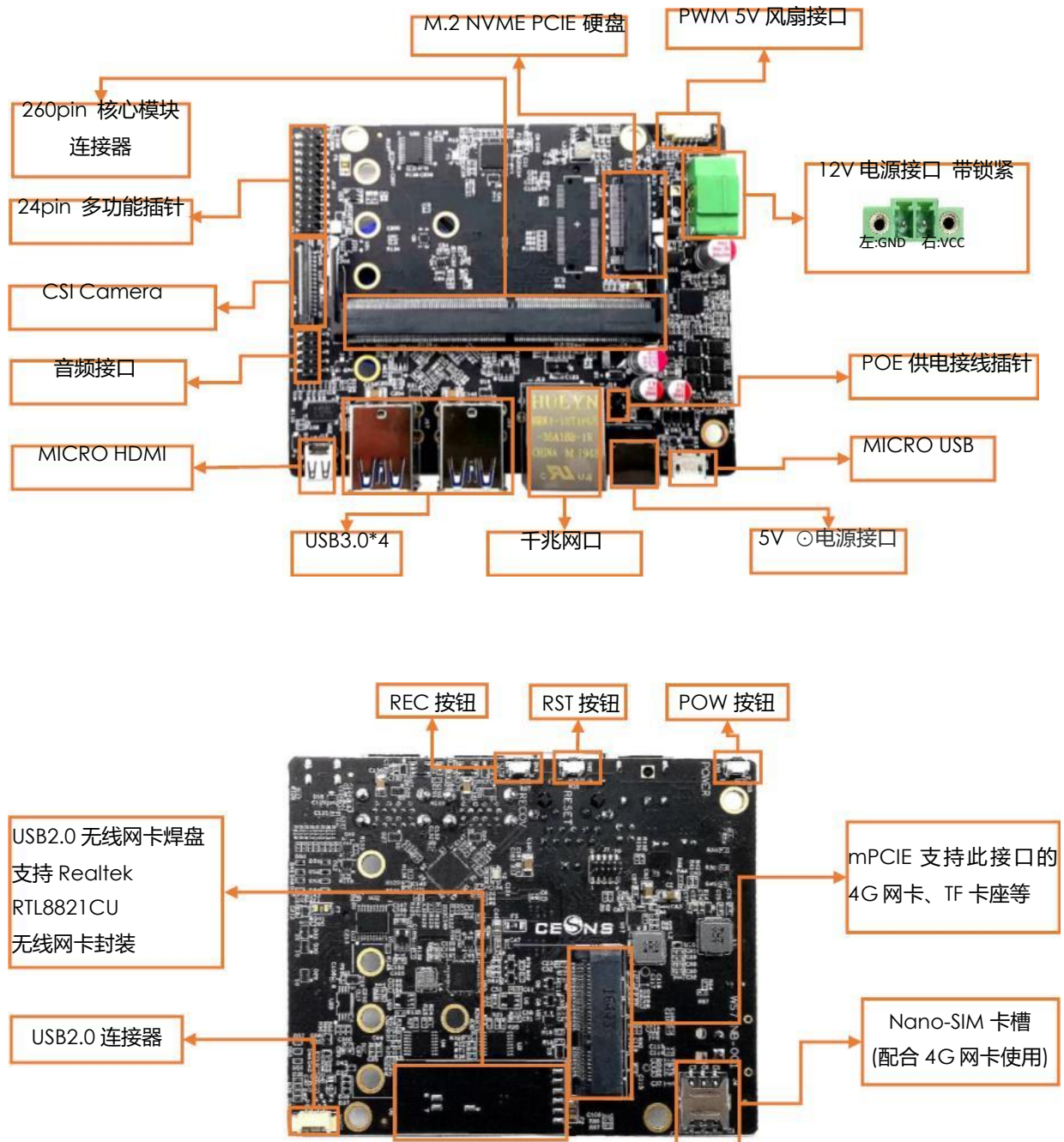
接口丰富,提供 m.2, mPCIe, GLAN, 4GLTE, USB3.0*4, HDMI2.0, UART, MICRO USB, GPIO, I²C, CAN, PWM-FAN 等丰富的外围接口。

产品硬件参数 (搭配 NANO/NX)

处理器模组	NVIDIA JETSON NANO	NVIDIA JETSON XAVIER NX
AI Performance	472 GFLOPs	21 TOPs
OS	UBUNTU 18.04	UBUNTU 18.04
CPU	Dual-Core Denver 1.5 64-Bit CPU and ARM-A57	6-core 64-bit CPU NVIDIA Carmel ARMv8.2
GPU	256-core NVIDIA Pascal GPU	384-core NVIDIA Volta GPU
Memory	4 GB 64-bit LPDDR4	8 GB 128-bit LPDDR4x
Storage	16GB eMMC 5.1 M.2 M key NVME 2242 固态硬盘扩展	16GB eMMC 5.1 M.2 M key NVME 2242 固态硬盘扩展
Power	5V/12V DC 5W/10W	5V/12V DC 10W/15W
Display	MICRO HDMI	MICRO HDMI
USB	USB3.0*4, MICRO USB2.0*1, USB2.0*2	USB3.0*4, MICRO USB2.0*1, USB2.0*2
NETWORK	Gigabit Lan ,4G-LTE, WIFI	Gigabit Lan ,4G-LTE, WIFI
Interfaces	mPCIe,UART,GPIO,I ² C,CAN 5V-PWM-FAN	mPCIe,UART,GPIO,I ² C,CAN 5V-PWM-FAN
Mechanical	底板 100*80*26mm,底板 73g, (nano 核心 23g、散热 39/51/54g)	底板 100*80*26mm,底板 73g, (NX 核心 26g、散热 52g)
Temperature	-45℃-85℃	-45℃-85℃

产品介绍:

1.2 产品功能示意图



产品介绍:

1.3 接口描述

1.3.1 MICRO HDMI

引脚	信号名称	引脚	信号名称
1	Hot Plug Detect	2	Utility
3	TMDS Data2+	4	TMDS Data2 Shield
5	TMDS Data2-	6	TMDS Data1 +
	TMDS Data1 Shield		TMDS Data1-
9	TMDS Data0+	10	TMDS Data0 Shield
11	TMDS Data0-	12	TMDS Clock+
13	TMDS Clock Shield	14	TMDS Clock-
15	SCL CEC	16	DDC /CEC Ground
17		18	SDA
19	+5V Power		

1.3.2 USB 3.0

引脚	信号名称	引脚	信号名称
1	VBUS	2	USB 2.0 D-
3	USB 2.0 D+	4	GND
5	SSRX-	6	SSRX+
7	GND	8	SSTX-
9	SSTX+		

1.3.3 MICRO USB

引脚	信号名称	引脚	信号名称
1	VBUS	2	USB 2.0 D-
3	USB 2.0 D+	4	USB ID
5	GND		

1.3.4 USB 2.0 连接器

引脚	信号名称	引脚	信号名称
1	5V	2	USB 22_D_N
3	USB 22_D_P	4	GND

产品介绍:

1.3.5 音频接口

引脚	信号名称	引脚	信号名称
1	VDD_5V_IN	2	GND
3	GPIO12_LS	4	GPIO09_LS
5	I ² C2_SCL_LS	6	I ² S0_SCLK_LS
7	I ² C2_SDA_LS	8	I ² S0_SDOU_OUT_LS
9	I ² S0_LRCK_LS	10	I ² S0_SDIN_LS

1.3.6 风扇接口

引脚	信号名称	引脚	信号名称
1	GND	2	+5V
3	FAN_TACH_CON	4	FAN_PWM

1.3.7 千兆网口

引脚	信号名称	引脚	信号名称
1	TP0+	2	TP0-
3	TP1+	4	TP2+
5	TP2-	6	TP1-
7	TP3+	8	TP3-

1.3.8 12V 电源接口

引脚	信号名称	引脚	信号名称
1	电源 DC 接口		

输入电压范围: +12V

1.3.9 5V 电源接口

引脚	信号名称	引脚	信号名称
1	电源 DC 接口		

输入电压范围: +5V

1.3.10 POE 供电插针

引脚	信号名称	引脚	信号名称
1	VC1	2	VC2
3	VC3	4	VC4

产品介绍:

1.4 24 针多功能接口

1.4.1 多功能引脚定义

引脚	信号名称	引脚	信号名称
1	3.3V	2	3.3V
3	UART0_TX	4	UART0_RX
5	UART1_RX	6	UART1_TX
7	GPIO_0	8	GPIO_1
9	GPIO_2	10	GPIO_3
11	I ² C_GP1_CLK	12	I ² C_GP1_DAT
13	RECOVERY	14	RTC_BAT_INPUT
15	RESET	16	PC_LED+
17	POWER_BUTTON	18	PC_LED-
19	GND	20	GND
21	-	22	-
23	CAN0H	24	CAN0L

1.4.2 功能测试

1.4.2.1 URATO

引脚	信号名称	引脚	信号名称
3	UART0_TX	4	UART0_RX

注: URATO 为调试串口, 可进行 ubuntu 系统调试, /dev/ttyTHS0
测试工具可以参考使用 cutecom: sudo apt-get install cutecom
连接 NX/NANO 和主机打开 cutecom, 波特率设置为 115200/8N1。

1.4.2.2 URAT1

引脚	信号名称	引脚	信号名称
5	UART1_RX	6	UART1_TX

注: 串口为/dev/ttyTHS0
此串口默认有 DEBUG 功能, 可关闭此功能只保留普通串口功能
输入命令如下:
\$ systemctl stop nvgetty
\$ systemctl disable nvgetty
\$udevadm trigger
重启即可

产品介绍:

1.4.2.3 CAN0

引脚	信号名称	引脚	信号名称
23	CAN0H	24	CAN0L

注: 测试 can

#安装测试工具

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

#执行 can 服务端脚本 (另存为 can_server.sh)

```
#!/bin/bash
can_init(){
echo "nvidia" | 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
}
NN=`ifconfig | grep "can0" >findout && cat findout`
if [ "$NN" = "" ]
then
can_init
fi
#candump can0 && cansend can1 1f334455#1122334455667788
candump can0
```

运行: sudo ./can_server.sh

#执行 can 客户端脚本 (另存为 can_client.sh)

```
#!/bin/bash
can_init(){
echo "nvidia" | sudo -S ip link set can0 type can bitrate 500000
sudo ip link set up can0
}
NN=`ifconfig | grep "can0" >findout && cat findout`
if [ "$NN" = "" ]
then
can_init
fi
cansend can0 1F334455#1122334455667788
```

运行: sudo ./can_client.sh

产品介绍:

1.4.2.4 GPIO

引脚	信号名称	引脚	信号名称
7	GPIO_0	8	GPIO_1
9	GPIO_2	10	GPIO_3

注: GPIO 测试方法:

```
#查看 gpio (7/8/9/10 引脚对应 GPIO 编号为 421/393/422/424)
```

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

```
#加载 gpio
```

```
echo '421'|sudo tee /sys/class/gpio/export
```

```
echo '393'|sudo tee /sys/class/gpio/export
```

```
echo '422'|sudo tee /sys/class/gpio/export
```

```
echo '424'|sudo tee /sys/class/gpio/export
```

```
#设置 gpio 输出方向
```

```
cd gpio421
```

```
echo 'out'|sudo tee /sys/class/gpio/gpio421/direction
```

```
#gpio3.3v 电压
```

```
echo '1'|sudo tee /sys/class/gpio/gpio421/value
```

```
#gpio 0v 电压
```

```
echo '0'|sudo tee /sys/class/gpio/gpio421/value
```

```
#设置 gpio 输入方向,在/sys/class/gpio 目录下
```

```
#查看 gpio 值, 将要输入的设置成 0
```

```
cat ./gpio393/value (显示 0)
```

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

```
#接高电压, 3.3v 返回值为 1
```

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

1.4.2.5 I²C

引脚	信号名称	引脚	信号名称
11	I ² C_GP1_CLK	12	I ² C_GP1_DAT

注: I²C 测试方法:

```
#查看总线
```

```
sudo i2cdetect -l
```

```
#查看总线上是否识别到从设备 (BUSID 为总线编号), 结果有数字和 UU 代表有设备
```

```
sudo i2cdetect -y BUSID
```

```
#读取 16 位数据 (BUSID 为总线编号) (w 写入 2 位) (0x50 为数字和 UU 对应的地址), (0x00; 0x20 为寄存地址)  
(r 为读取 16 位)
```

```
sudo i2ctransfer -f -y BUSID w2@0x50 0x00 0x20 r16
```

```
#写 4 位数据 (BUSID 为总线编号) (w 写入 4 位) (0x50 为数字和 UU 对应的地址), (0x00; 0x20 为寄存地址) (0x77  
0x77 为修改新内容)
```

```
sudo i2ctransfer -f -y BUSID w4@0x50 0x00 0x20 0x77 0x77
```

系统烧录:

2.1 烧录准备

下载镜像压缩包: 爱视图灵产品资料 (内含: NX 镜像包, 8821 网卡驱动)

链接: <https://pan.baidu.com/s/11BalbjgTOa5sGM4vIhTrGQ>

提取码: 8cdg

2.2 烧录过程

(a) 解压下载的安装包例如: NX_img_L4T_R32.4.2.zip 或 NANO_img_L4T_R32.4.2.zip

(b) 进入解压的目录, 运行:

`sudo ./install_basic.sh` (安装初始镜像脚本)

`sudo ./install_costem_img_cuda_trt_opencv_deepstream_+wifi.sh` (安装 JETPACK 完整安装镜像的安装脚本)

板载无线网卡驱动: `nx-nano-wif-linux-driver-source.zip`

解压后运行安装驱动: `sudo ./install.sh`。

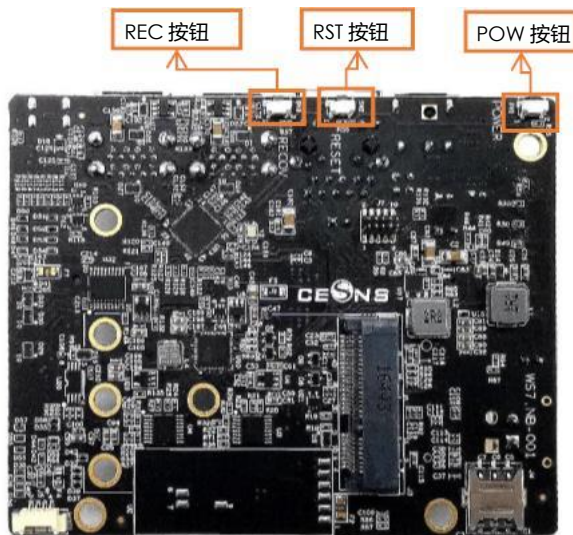
(c) (1) 底板+NANO 或 NX 核心+散热组装完成, 接上 12V 绿端子电源;

(2) 按一下 power 键 (POW) 开机;

按住 Recovery 键 (REC) ,不松开;

同时,按下 Reset 键 (RST) 并松开;

继续保持按住 Recovery 键 2-3 秒, 然后松开 Recovery 键。



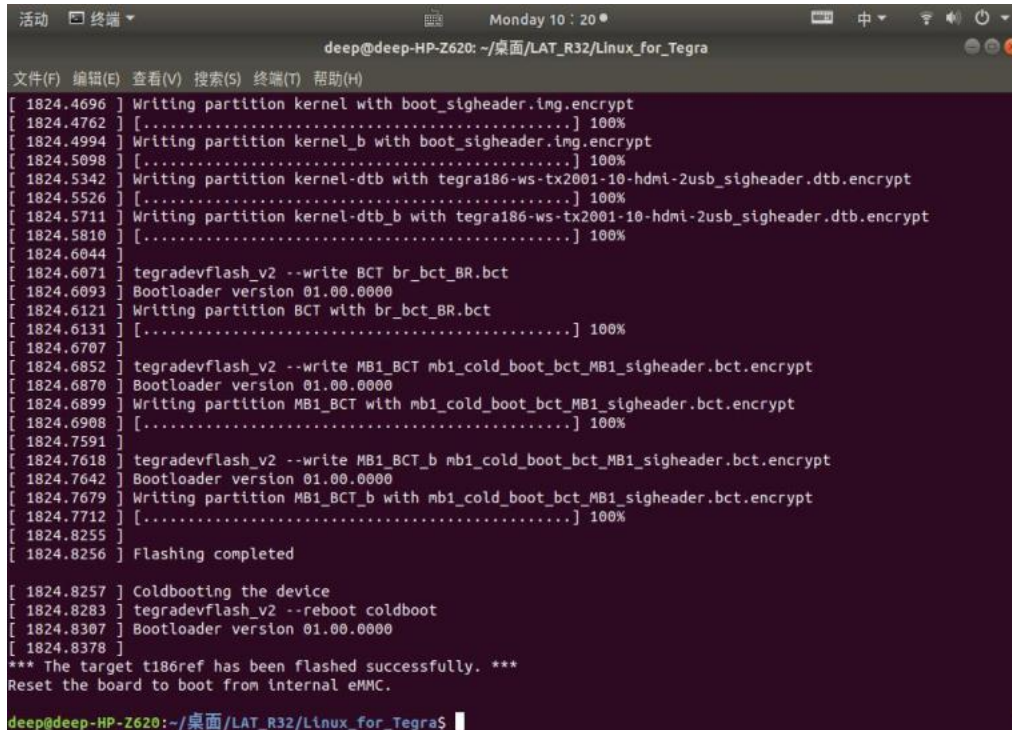
(3) 判断是否成功进入 Recovery 模式, 可以使用 `lsusb` 命令查看是否有 “NVIDIA Corp” 的设备。

```
hcq@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

(d) 按照 `install.sh` 运行完成时的文字说明、或解压缩路径下的 `readme.txt` 文件的介绍进行 `flash.sh` 的烧录操作。如: 烧录 `sudo ./flash.sh jetson-xavier-nx-devkit-emmc mmcblk0p1`

系统烧录:



```
deep@deep-HP-Z620: ~/桌面/LAT_R32/Linux_for_Tegra
[ 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%
[ 1824.6044 ]
[ 1824.6071 ] tegradevflash_v2 --write BCT br_bct_BR.bct
[ 1824.6093 ] 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.
deep@deep-HP-Z620:~/桌面/LAT_R32/Linux_for_Tegra$
```

2.3 烧录自己的镜像

2.3.1 备份镜像方法:

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

然后备份生成的 backup.img.raw 的镜像文件 (建议压缩为 zip 文件存储)

2.3.2 恢复镜像方法:

将备份的文件名为 backup.img.raw 的镜像拷贝到 Linux_for_Tegra/bootloader/目录下重命名为 system.img:

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

在 Linux_for_Tegra/bootloader/目录下执行:

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

注: -r 参数指使用 bootloader 目录下的 system.img 烧录。

注: 备份和恢复均需进入 RECOVER 模式下。

订货信息：

订货	描述
WS-MDN-501板卡	JETSON NANO/NX 模块的接口底板
NVIDIA Jetson NANO B01 核心模块	NVIDIA Jetson NANO B01 16G EMMC 核心模块
NVIDIA Jetson XAVIER NX 核心模块	NVIDIA Jetson XAVIER NX 16GEMMC 核心模块
WS-PDN-501-001整机	包含 NVIDIA Jetson NANO B01(16G)核心模块, WS-MDN-501板卡 M.2 NVME 128G 固态硬盘 和全套内部配件
WS-PDN-501-004整机	包含 NVIDIA Jetson XAVIER NX(16G)核心模块, WS-MDN-501板卡 M.2 NVME 128G 固态硬盘 和全套内部配件
	128G

技术支持：

相关技术支持请联系：

电话：010-56865039

传真：010-56865040

邮箱：wangyanyong@zhiyongxingkong.com

了解公司相关产品动态、技术交流、下单采购请扫描下列二维码



店铺



技术支持

公司地址：北京市通州区兴光四街一号A座203室

联系人：王雁勇 **手机：**13501240697 **电话：**010-56865030