Micro HDMI

Pin	Pin Description	Pin	Pin Description
1	Hot Plug Detect	2	Utility
3	TMDS Data2+	4	TMDS Data2 Shield
5	TMDS Data2-	6	TMDS Data1+
7	TMDS Data1 Shield	8	TMDS Data1-
9	TMDS Data0+	10	TMDS Data0 Shield
11	TMDS Data0-	12	TMDS Clock+
13	TMDS Clock Shield	14	TMDS Clock-
15	CEC	16	DDC /CEC Ground
17	SCL	18	SDA
19	+5V Power		

USB 3.0

Pin	Pin Description	Pin	Pin Description
1	VBUS	2	USB 2.0 D-
3	USB 2.0 D+	4	GND
5	SSRX-	6	SSRX+
7	GND	8	SSTX-
9	SSTX+		

Micro USB

Pin	Pin Description	Pin	Pin Description
1	VBUS	2	USB 2.0 D-
3	USB 2.0 D+	4	USB ID
5	GND		

USB 2.0 connector

Pin	Pin Description	Pin	Pin Description
1	5V	2	USB 22_ D_N
3	USB 22_ D_P	4	GND

Audio connector (10 Pin)

Pin	Pin Description	Pin	Pin Description
1	VDD _5V_IN	2	GND
3	GPIO12_LS	4	GPIO09_LS
5	I²C2_SCL_LS	6	I²S0_SCLK_LS
7	I²C2_SDA_LS	8	I²S0_SDOUT_LS
9	I²S0_LRCK_LS	10	I²S0_SDIN_LS

Fan

Pin	Pin Description	Pin	Pin Description
1	GND	2	+5V
3	FAN_TACH_CON	4	FAN_PWM

Ethernet

Pin	Pin Description	Pin	Pin Description
1	TP0+	2	TP0-
3	TP1+	4	TP2+
5	TP2-	6	TP1-
7	TP3+	8	TP3-

12V Power

Pin	Pin Description	Pin	Pin Description
1	Power DC		
Input range：+12V			

5V Power

Pin	Pin Description	Pin	Pin Description
1	Power DC		
Input range：+5V			

POE

Pin	Pin Description	Pin	Pin Description
1	VC1	2	VC2
3	VC3	4	VC4

24 Pin

Pin	Pin Description	Pin	Pin Description
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	I²C_GP1_CLK	12	I²C_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

URAT0

Pin	Pin Description	Pin	Pin Description
3	UART0_TX	4	UART0_RX

Note：URAT0 is a debug serial port，you can debug via ubuntu system
you can install 'cutecom'：

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

Set port rate as 115200/8N1

URAT1

Pin	Pin Description	Pin	Pin Description
5	UART1_RX	6	UART1_TX

Note：/dev/ttyTHS0
On default it could debug，you could close debug function to use it as normal pin
Type the following command：

```
$ systemctl stop nvgetty
$systemctl disable nvgetty
$udevadm trigger
```

and restart the whole system

CAN0

Pin	Pin Description	Pin	Pin Description
23	CAN0H	24	CAN0L

Note：Testing can

#Install testing tool

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

#Run CAN server（save as : 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
```

Then run：

```
sudo ./can_server.sh
```

#Run CAN client（save as: 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
```

Run ：

```
sudo ./can_client.sh
```

GPIO

Pin	Pin Description	Pin	Pin Description
7	GPIO_0	8	GPIO_1
9	GPIO_2	10	GPIO_3

Note：GPIO testing：
#check gpio（Pin 7/8/9/10 match GPIO No. 421/393/422/424）

```
cd /sys/class/gpio #load 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
```

#Set gpio output direction

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

#gpio 3.3v

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

#gpio 0v

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

#Set gpio input direction under /sys/class/gpio
#Check gpio value，set input as '0'

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

#Set high input，3.3v will return '1'

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

I²C

Pin	Pin Description	Pin	Pin Description
11	I²C_GP1_CLK	12	I²C_GP1_DAT

Note：I²C testing：
#Check bus

```
sudo i2cdetect -l
```

#Check BusID，if values and 'UU address' shown, device detected successfully

```
sudo i2cdetect -y BUSID
```

#Read BusID and write（0x50 is the values and UU address），（0x00；0x20 is register address），（r means read）

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

#Write （0x77 0x77 is the modified value）

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