

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+	10	VBUS
11	USB 2.0 D-	12	USB 2.0 D+
13	GND	14	SSRX-
15	SSRX+	16	GND
17	SSTX-	18	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		

26 pin 2.0mm

Pin	Pin Description	Pin	Pin Description
1	3.3V	2	3.3V
3	UART0_TX	4	UART0_RX
5	UART1_TX	6	UART1_RX
7	GPIO0	8	GPIO1
9	GPIO2	10	GPIO3
11	I ² C_CLK	12	I ² C_DAT
13	RECOVERY	14	RTC_BAT
15	RESET	16	REMOTE_TX
17	POWER_BUTTON	18	REMOTE_RX
19	GND	20	GND
21	CAN1H	22	CAN1L
23	CAN0H	24	CAN0L

25 UART3_TX (only for TX2-4G,
TX2i)

26 UART3_RX (only for TX2-4G,
TX2i)

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/ttyTHS2

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 (TCAN332DCN)

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

```
cd /sys/class/gpio #load gpio
echo '388'|sudo tee /sys/class/gpio/export
echo '298'|sudo tee /sys/class/gpio/export
echo '480'|sudo tee /sys/class/gpio/export
echo '486'|sudo tee /sys/class/gpio/export
```

#Set gpio output direction

```
cd gpio388
echo 'out'|sudo tee /sys/class/gpio/gpio388/direction
#gpio 3.3v
echo '1'|sudo tee /sys/class/gpio/gpio388/value
#gpio 0v
echo '0'|sudo tee /sys/class/gpio/gpio388/value
#Set gpio input direction under /sys/class/gpio
#Check gpio value, set input as '0'
cat ./gpio480/value
echo 'in'|sudo tee /sys/class/gpio/gpio480/direction
#Set high input, 3.3v will return '1'
cat /sys/class/gpio/gpio480/value
```

I²C

Pin	Pin Description	Pin	Pin Description
11	I ² C_GP1_CLK	12	I ² C_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
```

CAN1

Pin	Pin Description	Pin	Pin Description
21	CAN1H	22	CAN1L

URAT3

Pin	Pin Description	Pin	Pin Description
25	UART3_TX	26	UART3_RX

Note: Only works when using TX2-4G or TX2i

422 (UART2)

Pin	Pin Description	Pin	Pin Description
1	A	2	B
3	GND	4	Z
5	Y		

Note : /dev/ttyTHS1

FAN

Pin	Pin Description	Pin	Pin Description
1	GND	2	+5V
3	FAN_TACH	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-

Power

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

- Hardware Overview

