'ODROID-N2' on this page refers to the ODROID-N2 series (**N2**, **N2+**, **N2L**).

How to enable UART on the pin header

- Want to see the previous documents? Please visit this page.
  - https://wiki.odroid.com/legacy/common/application_note/gpio/uart

- If you have the kernel version written in below, you have to use the **Device Tree Overlay** to enable the GPIO functions.
  - **ODRODI-XU4**: **5.4** or above.
  - **ODRODI-N2/N2+**: **4.9.230** or above.
  - **ODRODI-C4**: **4.9.230** or above.
  - Introduction of Device Tree Overlay: https://wiki.odroid.com/common/application_note/software/device_tree_overlay
  - Brief to know how to enable DTBO using **config.ini** file: https://wiki.odroid.com/common/application_note/gpio/enable_spi_i2c_uart_with_dtbo

You can use a general-purpose UART interface on the onboard pin header.

Please see the pin expansion descriptions to know how UART pins are exposed on the board before reading this guide.

- ODRODI-XU4
- ODRODI-N2/N2+
- ODRODI-C4
- ODRODI-M1

As you can see, there're several sets of UART pins on the onboard pin header. Those locations differ from each other but in common, pin #8 and #10 will always be used for the first exposed UART pin set. Additionally, each board has the other UART pins available.

In this guide, you will know how you can enable the UART pins on the board.

How to use

Overview

- If you enable UART functionality of the pins, you cannot use them as GPIO or the other alternative function.

Basically, the necessary nodes for UART functionality are already defined in the device source tree. If you enable the nodes by editing the device tree, you can use the function without any other bothers.

It seems easy but the real work doesn't. But thanks to the Device Tree Overlay, you can enable them just edit the **/media/boot/config.ini** file (ODROID-XU4/N2/C4) or **/boot/config.ini**(ODROID-M1).

Enable UART pins

Open **/media/boot/config.ini** / **/boot/config.ini** file with your favorite text editor.

Then edit the DTBO parts of the file as shown below. See the **overlays** variable having **“uart0”** or **“uart1”** value.

The following example shows uart0 added to config.ini.

- If you set nothing, then all pins will work as a general-purpose I/O mode.

```
; Device Tree Overlay
overlay_resize=16384
overlay_profile=
overlays="uart0"

[overlay_custom]
overlays="i2c0 i2c1"

[overlay_hktft32]
overlays="hktft32"

[overlay_hktft35]
overlays="hktft35
```

Reboot to apply the new DTBO settings. After reboot, you can find the extra **/dev/ttyS1** (ODROID-XU4/N2/C4) or **/dev/ttyS0** (ODROID-M1) device files in the device list.

target

```
$ ls -al /dev/ttyS?
crw--w---- 1 root tty     238, 0 May 28 18:36 /dev/ttyS0
crw-rw---- 1 root dialout 238, 1 Jan 29  2018 /dev/ttyS1
```

All of the available UART pins table

Each board has a **different number** of UART pin sets. Here's how DTBO settings in **config.ini** look like if all available UART pins enabled.

Odroid-XU4

UART	DTBO	Pin header	Phy Pin # (RX)	Phy Pin # (TX)	GPIO Pin # (RX)	GPIO Pin # (TX)	Device File
UART_0	uart0.dtbo	CON10 (2*15 pins)	6	8	171 (GPA0.0)	172 (GPA0.1)	/dev/ttySAC0
		Shifter Shield	10	8			

```
; Device Tree Overlay
overlay_resize=16384
overlay_profile=
overlays="uart0"
```

Odroid-N2/N2+

UART	DTBO	Pin header	Phy Pin # (RX)	Phy Pin # (TX)	GPIO Pin # (RX)	GPIO Pin # (TX)	Device File
UART_EE_A	uart0.dtbo	J2 (2*20 pins)	10	8	489 (GPIOX.13)	488 (GPIOX.12)	/dev/ttyS1
UART_EE_B	uart1.dtbo	J2 (2*20 pins)	15	35	483 (GPIOX.7)	482 (GPIOX.6)	/dev/ttyS2

```
; Device Tree Overlay
overlay_resize=16384
overlay_profile=
overlays="uart0 uart1"
```

Odroid-C4

UART	DTBO	Pin header	Phy Pin # (RX)	Phy Pin # (TX)	GPIO Pin # (RX)	GPIO Pin # (TX)	Device File
UART_EE_A	uart0.dtbo	J2 (2*20 pins)	10	8	489 (GPIOX.13)	488 (GPIOX.12)	/dev/ttyS1
UART_EE_B	uart1.dtbo	J2 (2*20 pins)	15	33	483 (GPIOX.7)	482 (GPIOX.6)	/dev/ttyS2
UART_EE_C	uart2.dtbo	J2 (2*20 pins)	26	32	23 (GPIOH.6)	24 (GPIOH.7)	/dev/ttyS3
UART_AO_B	uart3.dtbo	J7 (1*7 pins)	4	6	505 (GPIOAO.9)	504 (GPIOAO.8)	/dev/ttyS4

```
; Device Tree Overlay
overlay_resize=16384
overlay_profile=
overlays="uart0 uart1 uart2 uart3"
```

Odroid-M1

UART	DTBO	Pin header	Phy Pin # (RX)	Phy Pin # (TX)	GPIO Pin # (RX)	GPIO Pin # (TX)	Device File
UART1_M1	uart0.dtbo	J1 (2*20 pins)	10	8	127 (GPIO3 D.BIT7)	126 (GPIO3 D.BIT6)	/dev/ttyS0
UART0	uart1.dtbo	J1 (2*20 pins)	11	13	16 (GPIO0 C.BIT0)	17 (GPIO0 C.BIT1)	/dev/ttyS1

```
[generic]
overlay_resize=16384
overlay_profile=
overlays="uart0 uart1"
```

In Odroid-C4, all the serial device files will be listed like,

target

```
$ ls -al /dev/ttyS?
crw--w---- 1 root tty     239, 0 Apr 26 05:45 /dev/ttyS0
crw-rw---- 1 root dialout 239, 1 Mar 17 21:36 /dev/ttyS1
crw-rw---- 1 root dialout 239, 2 Mar 17 21:36 /dev/ttyS2
crw-rw---- 1 root dialout 239, 3 Mar 17 21:36 /dev/ttyS3
crw-rw---- 1 root dialout 239, 4 Mar 17 21:36 /dev/ttyS4
```

- **/dev/ttyS0** in Odroid-N2 and Odroid-C4 is the default debugging UART interface.
- **/dev/ttyFIQ0** in ODROID-M1 is the default debugging UART interface.

UART module using the UART pins

- This guide is for **ODROID-N2 and C4** but you can use **/dev/ttyS1** (**/dev/ttySAC0** in ODROID-XU4, **/dev/ttyFIQ0** in ODROID-M1) on the other board as well since that first UART pin locates with the same pin numbers.

Connect wires

The pin map of onboard 4-pin CON5 header for our debugging UART (Shown as /dev/ttyS0) is,

```
_____UART____
|           |
|Pin 4 - GND|
|Pin 3 - RXD|
|Pin 2 - TXD|
|Pin 1 - VCC|
|_          |
  |__________|

3.3V LVTTL
```

You can connect the UART pins on the pin header to your host PC.

See the pin maps.

Odroid-N2/N2+

=== UART_EE_A (/dev/ttyS1) ===

J2 Header Pin #	UART Module Pin #	Role
9	4	GND
10	3	RxD
8	2	TxD
38	1	VCC

=== UART_EE_B (/dev/ttyS2) ===

J2 Header Pin #	UART Module Pin #	Role
9	4	GND
15	3	RxD
35	2	TxD
38	1	VCC

Odroid-C4

=== UART_EE_A (/dev/ttyS1) ===

J2 Header Pin #	UART Module Pin #	Role
9	4	GND
10	3	RxD
8	2	TxD
38	1	VCC

=== UART_EE_B (/dev/ttyS2) ===

J2 Header Pin #	UART Module Pin #	Role
9	4	GND
15	3	RxD
33	2	TxD
38	1	VCC

Odroid-M1

=== UART1_M1 (/dev/ttyS0) ===

J2 Header Pin #	UART Module Pin #	Role
9	4	GND
10	3	RxD
8	2	TxD
38	1	VCC

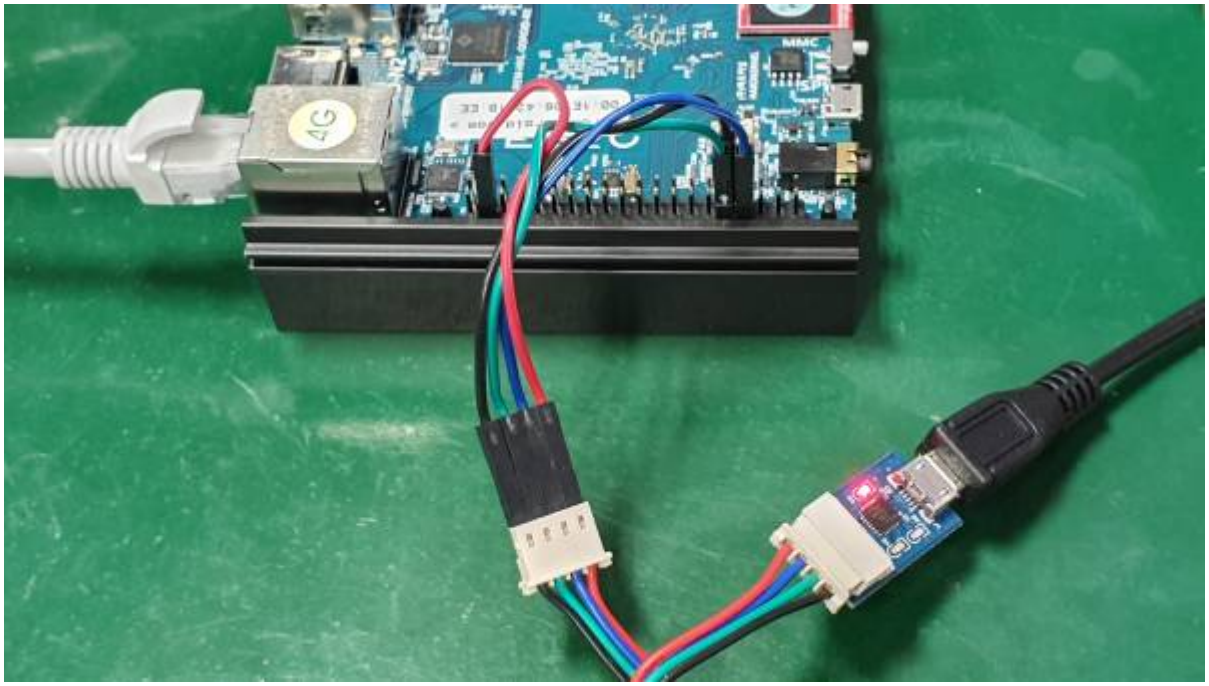
=== UART0 (/dev/ttyS1) ===

J2 Header Pin #	UART Module Pin #	Role
9	4	GND
11	3	RxD
13	2	TxD
38	1	VCC

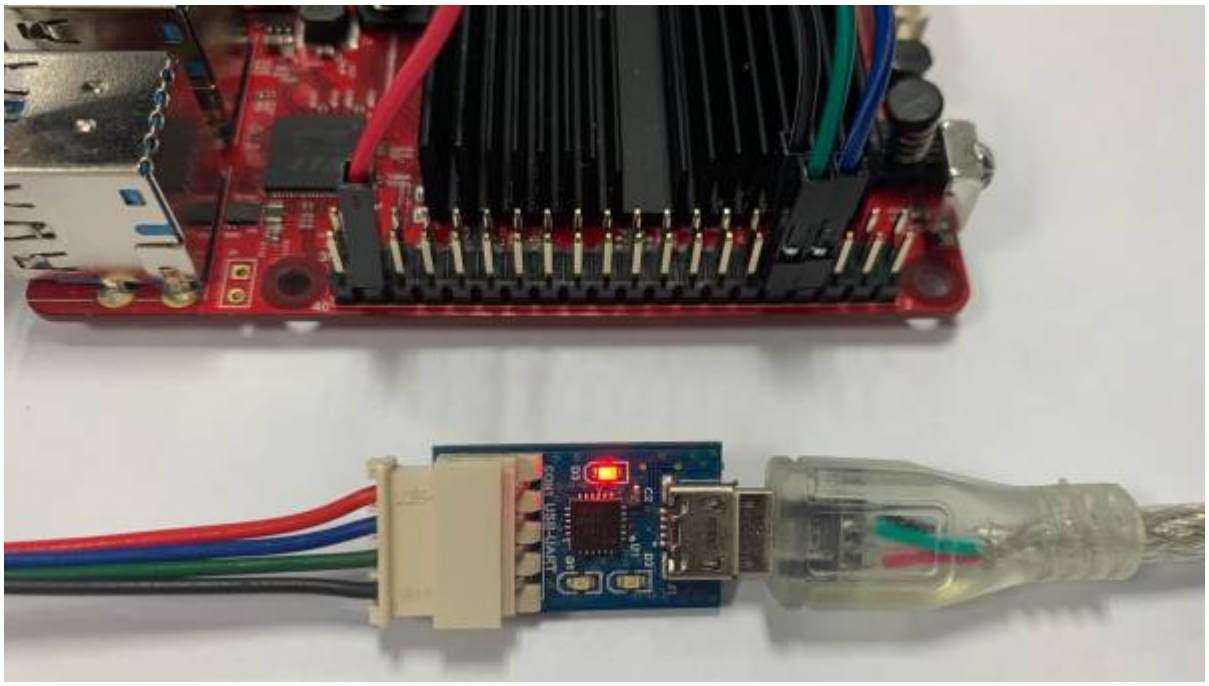
Connect the cables like the following picture, which shows when using **UART_EE_A**.

The RxD and TxD pins must be twisted together.

ODROID-N2



ODROID-C4



Connect the opposite side of the UART module to your host PC.

Test drive

- Assume that your host PC is Debian/Ubuntu system.

Enter the following commands to set **/dev/ttyS1** up.

target

```
$ sudo stty -F /dev/ttyS1 115200
```

On your host PC, install **minicom** to serial communication.

host

```
$ sudo apt install minicom
```

Check the UART module installed as **/dev/ttyUSB***.

host

```
$ ls /dev/ttyUSB?
/dev/ttyUSB0
```

Then open **minicom** with the following options.

host

```
$ minicom -b 115200 -D /dev/ttyUSB0
```

At your target board, put **something** to **/dev/ttyS1** device.

target

```
$ echo "something" | sudo tee /dev/ttyS1
```

Then you can see the message on the **minicom** screen on your host PC.

2021/04/26 15:01 · joshua