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

WiringPi and Python Wrapper

- Legacy **master** branch has renamed to **master-old** branch. If you face a problem with new master branch, please try again with old one.

WiringPi

The original WiringPi is a PIN based GPIO access library written in C for the BCM2835 used in the Raspberry Pi. It's released under the GNU LGPLv3 license and is usable from C, C++ and RTB (BASIC) as well as many other languages with suitable wrappers (See below) It's designed to be familiar to people who have used the Arduino "wiring" system[1]. Hardkernel provides WiringPi library for ODROID boards forked from original WiringPi.

You can install our WiringPi using our Ubuntu PPA (except for M1). This comes with you to keep it the latest version using **apt** command. Or, clone our Github repository and build it yourself.

Ubuntu PPA

- Please be aware that our package names **odroid-wiringpi**, not just **wiringpi**. Canonical provides the WiringPi package designed for RaspberryPi by default, but that version is not compatible with ours.

target

```
sudo apt install software-properties-common
sudo add-apt-repository ppa:hardkernel/ppa
sudo apt update
sudo apt install odroid-wiringpi
```

If you want to development libraries, install the extra packages.

target

```
sudo apt install libwiringpi-dev
```

Github repository

target

```
sudo apt install git
git clone https://github.com/hardkernel/wiringPi
cd wiringPi
sh autogen.sh
./configure
make
sudo make install
```

ODROID-M1

target

```
sudo apt update
sudo apt install odroid-wiringpi
```

If you want to development libraries, install the extra packages.

target

```
sudo apt install libwiringpi-dev
```

Run **gpio readall** to check all of the expansion GPIO pins. This is the default settings of ODROID-N2.

```
root@odroid:~# gpio readall
 +-----+-----+---------+------+---+--- N2 ---+---+------+---------+-----+---
--+
 | I/O | wPi |   Name  | Mode | V | Physical | V | Mode |  Name   | wPi |
I/O |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+---
--+
 |     |     |    3.3V |      |   |  1 || 2  |   |      | 5V      |     |
|
 | 493 |   8 |   SDA.2 | ALT1 | 1 |  3 || 4  |   |      | 5V      |     |
|
 | 494 |   9 |   SCL.2 | ALT1 | 1 |  5 || 6  |   |      | 0V      |     |
|
 | 473 |   7 |  IO.473 | ALT1 | 0 |  7 || 8  | 1 | ALT1 | TxD1    | 15  |
488 |
 |     |     |      0V |      |   |  9 || 10 | 1 | ALT1 | RxD1    | 16  |
489 |
 | 479 |   0 |  IO.479 |   IN | 1 | 11 || 12 | 1 | IN   | IO.492  | 1   |
492 |
 | 480 |   2 |  IO.480 |   IN | 1 | 13 || 14 |   |      | 0V      |     |
|
 | 483 |   3 |  IO.483 | ALT2 | 1 | 15 || 16 | 1 | IN   | IO.476  | 4   |
476 |
 |     |     |    3.3V |      |   | 17 || 18 | 1 | IN   | IO.477  | 5   |
477 |
 | 484 |  12 |    MOSI |   IN | 1 | 19 || 20 |   |      | 0V      |     |
|
 | 485 |  13 |    MISO |   IN | 1 | 21 || 22 | 1 | IN   | IO.478  | 6   |
478 |
 | 487 |  14 |    SCLK |   IN | 1 | 23 || 24 | 1 | IN   | CE0     | 10  |
486 |
 |     |     |      0V |      |   | 25 || 26 | 0 | IN   | IO.464  | 11  |
464 |
 | 474 |  30 |   SDA.3 | ALT2 | 1 | 27 || 28 | 1 | ALT2 | SCL.3   | 31  |
475 |
 | 490 |  21 |  IO.490 | ALT1 | 1 | 29 || 30 |   |      | 0V      |     |
|
 | 491 |  22 |  IO.491 | ALT1 | 1 | 31 || 32 | 0 | IN   | IO.472  | 26  |
472 |
 | 481 |  23 |  IO.481 |   IN | 1 | 33 || 34 |   |      | 0V      |     |
|
 | 482 |  24 |  IO.482 | ALT2 | 1 | 35 || 36 | 0 | IN   | IO.495  | 27  |
495 |
 |     |  25 |   AIN.3 |      |   | 37 || 38 |   |      | 1V8     | 28  |
|
 |     |     |      0V |      |   | 39 || 40 |   |      | AIN.2   | 29  |
|
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+---
--+
 | I/O | wPi |   Name  | Mode | V | Physical | V | Mode |  Name   | wPi |
I/O |
 +-----+-----+---------+------+---+--- N2 ---+---+------+---------+-----+---
--+
```

```
root@odroid:~# gpio readall -a
 +------+-----+----------+------+---+----+---- Model  ODROID-N2 ----+----+--
-+------+----------+-----+------+
 | GPIO | wPi |   Name   | Mode | V | DS | PU/PD | Physical | PU/PD | DS | V
| Mode |   Name   | wPi | GPIO |
 +------+-----+----------+------+---+----+-------+----++----+-------+----+--
-+------+----------+-----+------+
 |      |     |     3.3V |      |   |    |       |  1 || 2  |       |    |
|      | 5V       |     |      |
 |  493 |   8 |    SDA.2 | ALT1 | 1 |  2 |   P/U |  3 || 4  |       |    |
|      | 5V       |     |      |
 |  494 |   9 |    SCL.2 | ALT1 | 1 |  2 |   P/U |  5 || 6  |       |    |
|      | GND(0V)  |     |      |
 |  473 |   7 | GPIO.473 | ALT1 | 0 |  1 |   P/D |  7 || 8  | P/U   | 1  | 1
| ALT1 | TxD1     | 15  | 488  |
 |      |     |  GND(0V) |      |   |    |       |  9 || 10 | P/U   | 1  | 1
| ALT1 | RxD1     | 16  | 489  |
 |  479 |   0 | GPIO.479 |   IN | 1 |  2 |   P/U | 11 || 12 | P/U   | 1  | 1
| IN   | GPIO.492 | 1   | 492  |
 |  480 |   2 | GPIO.480 |   IN | 1 |  2 |   P/U | 13 || 14 |       |    |
|      | GND(0V)  |     |      |
 |  483 |   3 | GPIO.483 | ALT2 | 1 |  1 |   P/U | 15 || 16 | P/U   | 2  | 1
| IN   | GPIO.476 | 4   | 476  |
 |      |     |     3.3V |      |   |    |       | 17 || 18 | P/U   | 2  | 1
| IN   | GPIO.477 | 5   | 477  |
 |  484 |  12 |     MOSI |   IN | 1 |  1 |   P/U | 19 || 20 |       |    |
|      | GND(0V)  |     |      |
 |  485 |  13 |     MISO |   IN | 1 |  1 |   P/U | 21 || 22 | P/U   | 2  | 1
| IN   | GPIO.478 | 6   | 478  |
 |  487 |  14 |     SCLK |   IN | 1 |  2 |   P/U | 23 || 24 | P/U   | 1  | 1
| IN   | CE0      | 10  | 486  |
 |      |     |  GND(0V) |      |   |    |       | 25 || 26 | P/D   | 1  | 0
| IN   | GPIO.464 | 11  | 464  |
 |  474 |  30 |    SDA.3 | ALT2 | 1 |  3 |   P/U | 27 || 28 | P/U   | 3  | 1
| ALT2 | SCL.3    | 31  | 475  |
 |  490 |  21 | GPIO.490 | ALT1 | 1 |  1 |   P/U | 29 || 30 |       |    |
|      | GND(0V)  |     |      |
 |  491 |  22 | GPIO.491 | ALT1 | 1 |  1 |   P/U | 31 || 32 | P/D   | 2  | 0
| IN   | GPIO.472 | 26  | 472  |
 |  481 |  23 | GPIO.481 |   IN | 1 |  2 |   P/U | 33 || 34 |       |    |
|      | GND(0V)  |     |      |
 |  482 |  24 | GPIO.482 | ALT2 | 1 |  1 |   P/D | 35 || 36 | DSBLD | 1  | 0
| IN   | GPIO.495 | 27  | 495  |
 |      |  25 |    AIN.3 |      |   |    |       | 37 || 38 |       |    |
|      | 1V8      | 28  |      |
 |      |     |  GND(0V) |      |   |    |       | 39 || 40 |       |    |
|      | AIN.2    | 29  |      |
 +------+-----+----------+------+---+----+-------+----++----+-------+----+--
-+------+----------+-----+------+
 | GPIO | wPi |   Name   | Mode | V | DS | PU/PD | Physical | PU/PD | DS | V
| Mode |   Name   | wPi | GPIO |
 +------+-----+----------+------+---+----+---- Model  ODROID-N2 ----+----+--
-+------+----------+-----+------+
```

WiringPi Python Wrapper

Hardkernel also provides WiringPi Python binder for Python programmer called WiringPi-Python. WiringPi-Python is Python-wrapped version of Hardkernel's WiringPi.

Ubuntu PPA

We made a simple wrapper Debian package to install/update automatically for Python wrapper.

target

```
sudo apt install software-properties-common
sudo add-apt-repository ppa:hardkernel/ppa
sudo apt update
sudo apt install odroid-wiringpi-python
```

After installing this package, it installs **odroid-wiringpi PyPI package** using PIP.

This package has two dependency packages; **python3, python3-pip**. If you want to use Python 2, you can install like the below commands.

target

```
sudo apt install python python-pip
sudo odroid-wiringpi-python --install
```

Python PyPI

target

```
sudo apt install python python3 python-pip python3-pip

# Python 2
sudo python -m pip install odroid-wiringpi
# Python 3
sudo python3 -m pip install odroid-wiringpi
```

Github repository

target

```
sudo apt install git python-dev python-setuptools python3-dev python3-
setuptools swig
git clone --recursive https://github.com/hardkernel/WiringPi2-Python
cd WiringPi2-Python

# Python 2
```

```
sudo python setup.py install
# Python 3
sudo python3 setup.py install
```

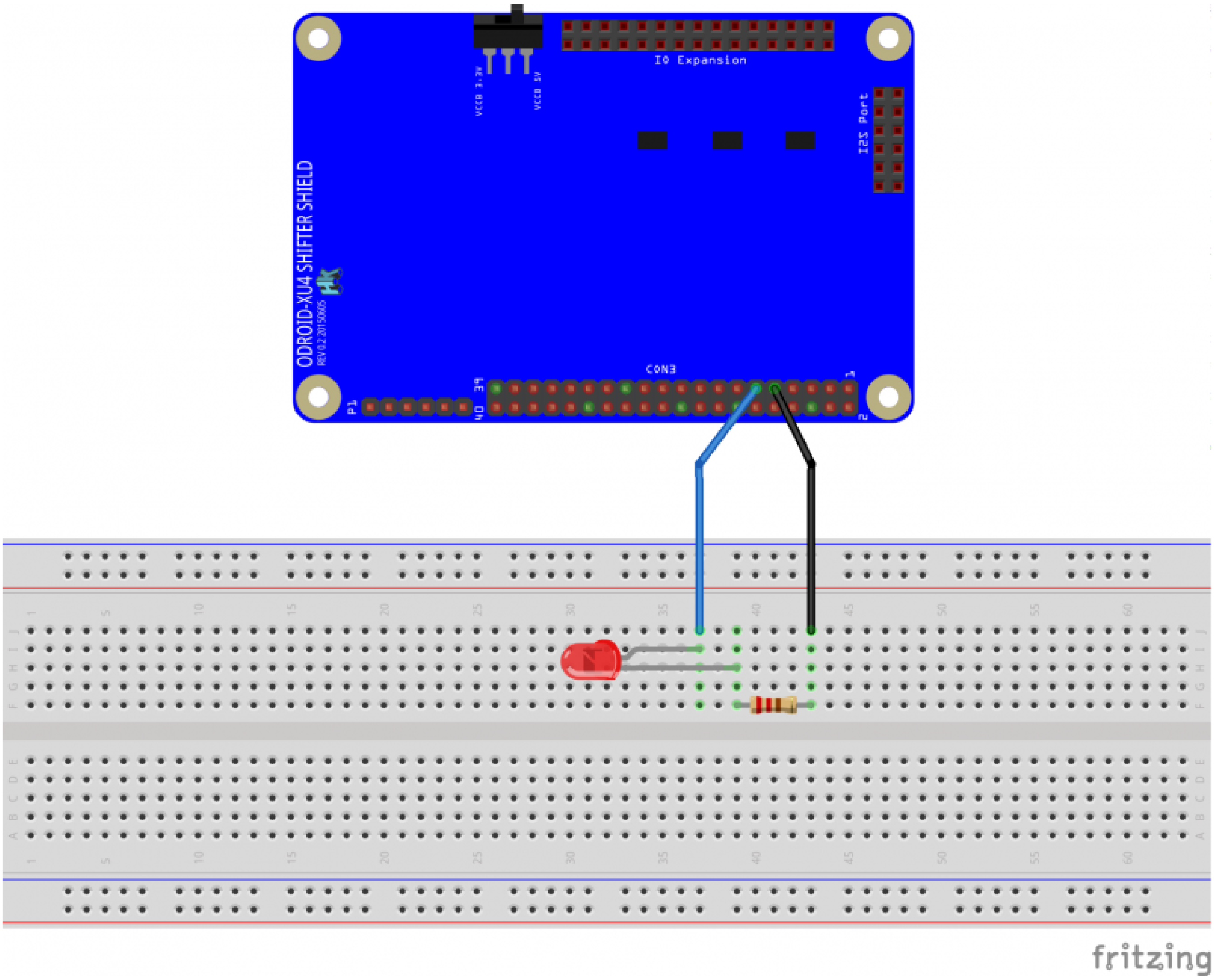
Examples

LED On/Off

You can turn on/off a connected LED using WiringPi.
This example uses **physical pin #11** (#0 on WiringPi) and a ground pin.

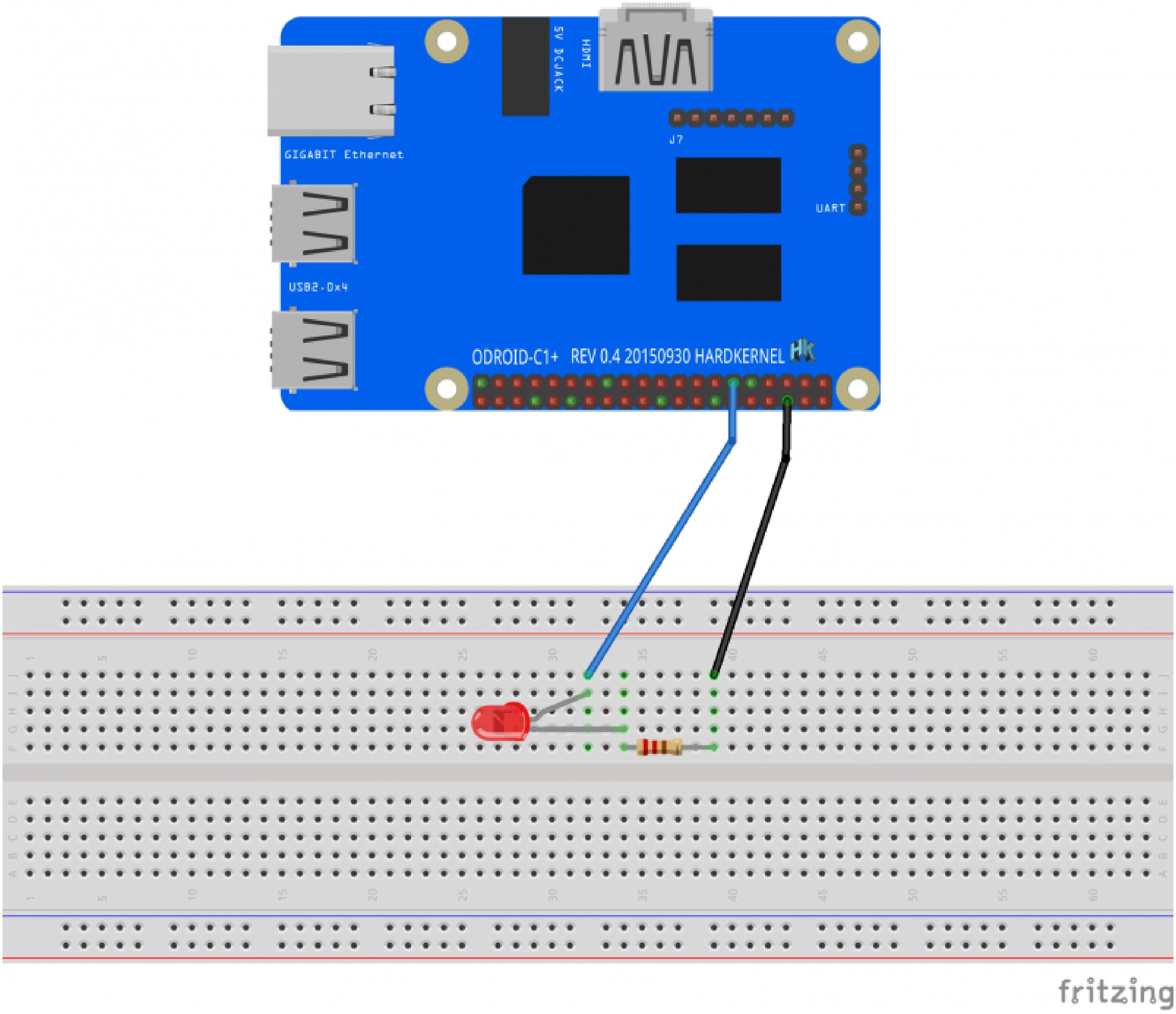
ODROID-XU4 (+Shifter Shield)

Please refer ShiftShield for more detail.



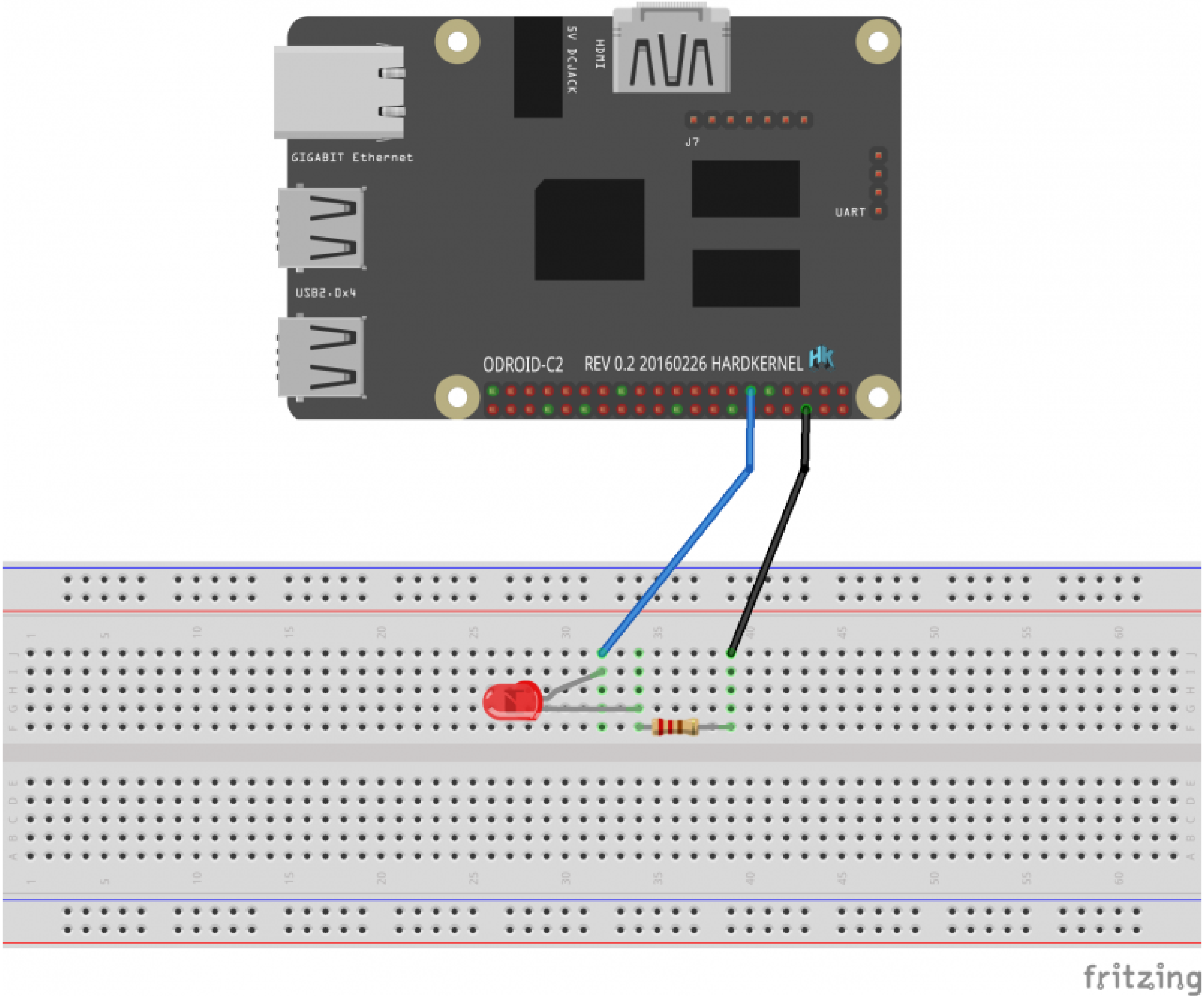
- Download Fritzing Example File: xu4_led.fzz

ODROID-C1+



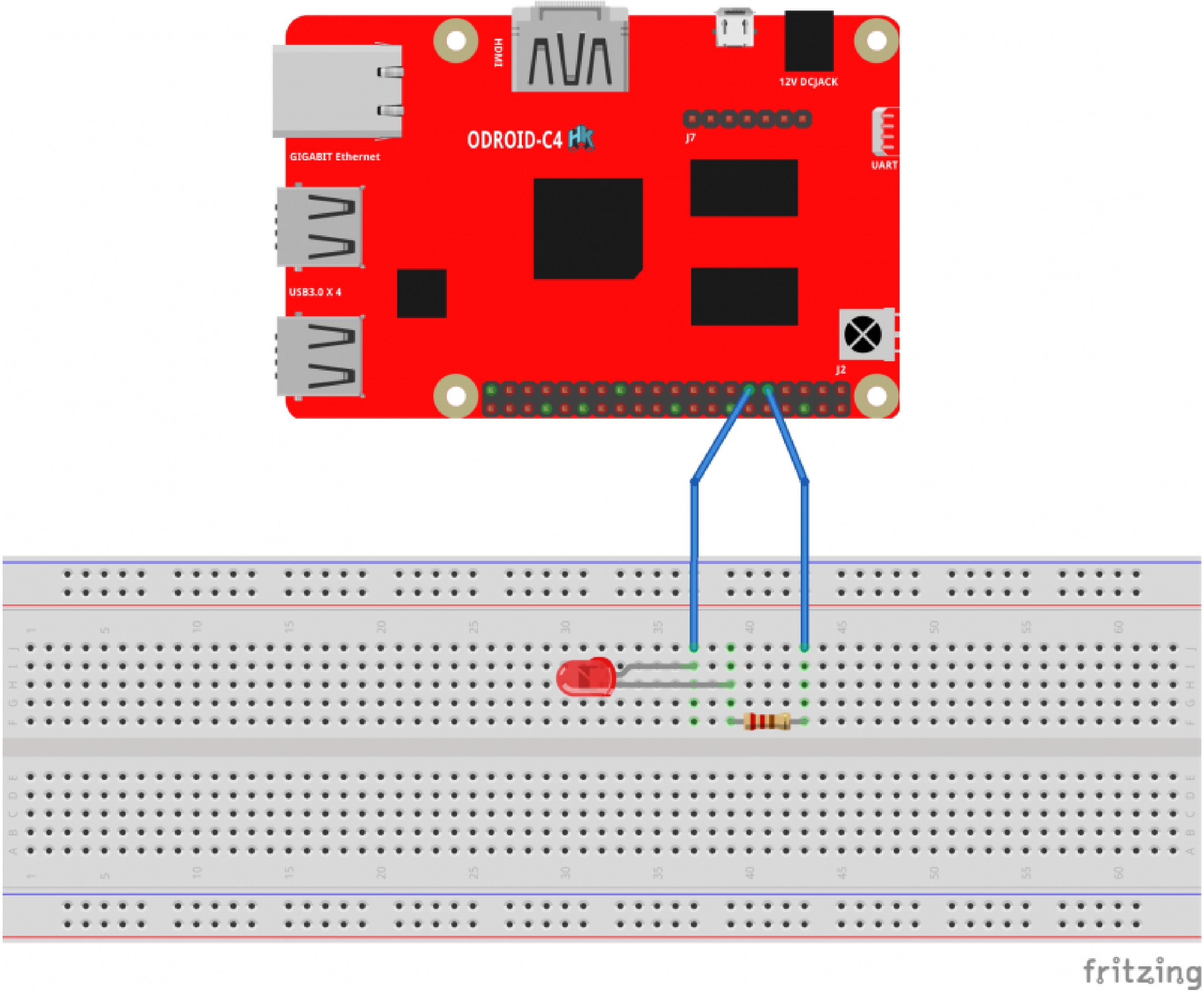
- Download Fritzing Example File: c1_led.fzz

ODROID-C2



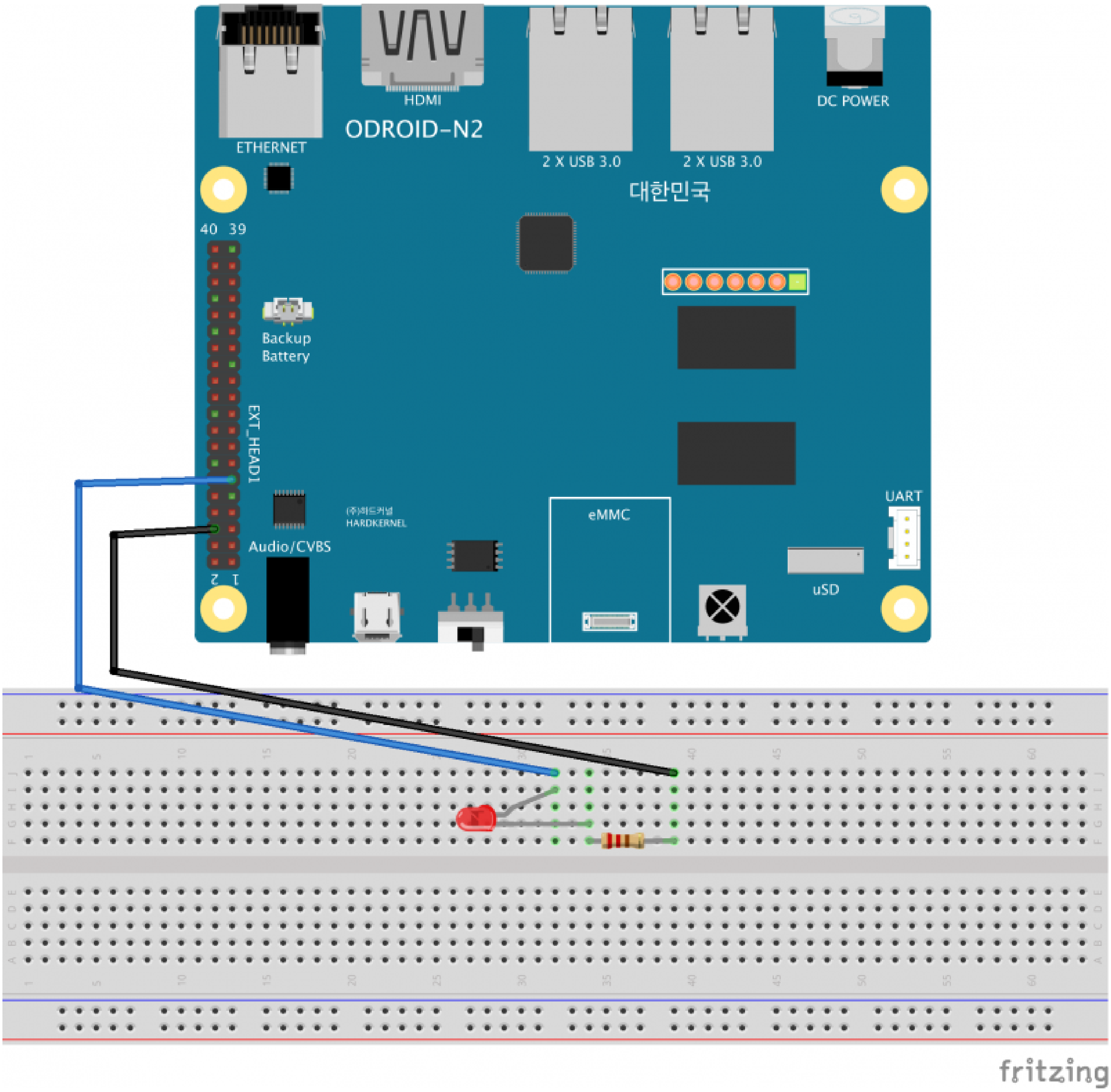
- Download Fritzing Example File: c2_led.fzz

ODROID-C4



- Download Fritzing Example File: odroid-c4.fzpz

ODROID-N2



- Download Fritzing Example File: n2_led.fzz

C

target

```
# C
$ gcc -o wpi_exam_led wpi_exam_led.c $(pkg-config --cflags --libs
libwiringpi2)
$ ./wpi_exam_led
```

wpi_exam_led.c

```
#include <wiringPi.h>

int main(void)
{
    wiringPiSetup();
    pinMode(0, OUTPUT);

    for (;;)
    {
        digitalWrite(0, HIGH);
        delay(1000);
        digitalWrite(0, LOW);
        delay(1000);
    }
    return 0;
}
```

Python 2

target

```
# Python 2
$ sudo python wpi_exam_led.py
```

wpi_exam_led.py

```
#!/usr/bin/env python

import odroid_wiringpi as wpi
import time

wpi.wiringPiSetup()
wpi.pinMode(0, 1)

while True:
    wpi.digitalWrite(0, 1)
    time.sleep(1)
    wpi.digitalWrite(0, 0)
    time.sleep(1)
```

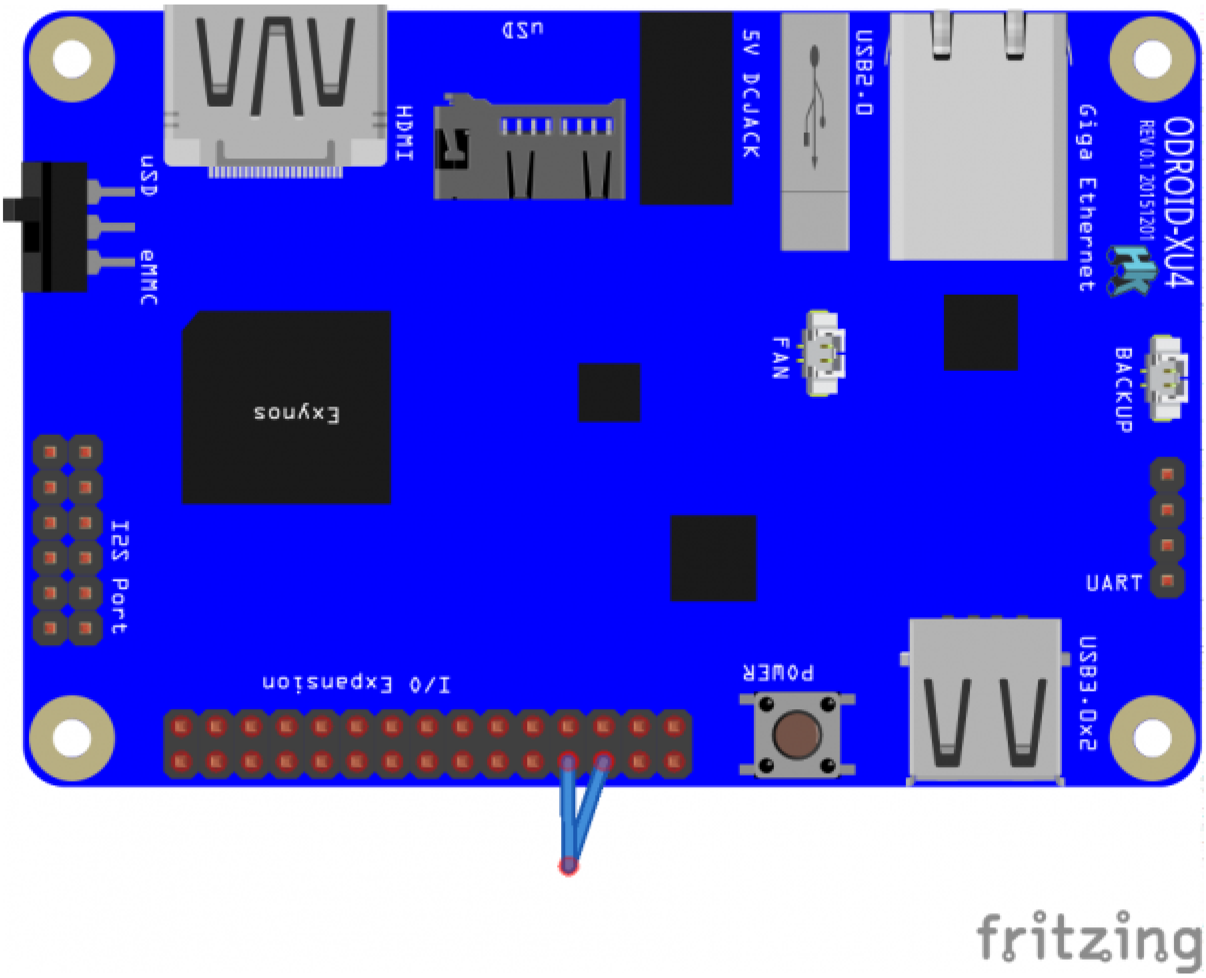
Serial Loopback

A simple serial loopback example.
This example uses **physical pin #8, #10** (#15, #16 on WiringPi).

Connect each other that two pins directly.

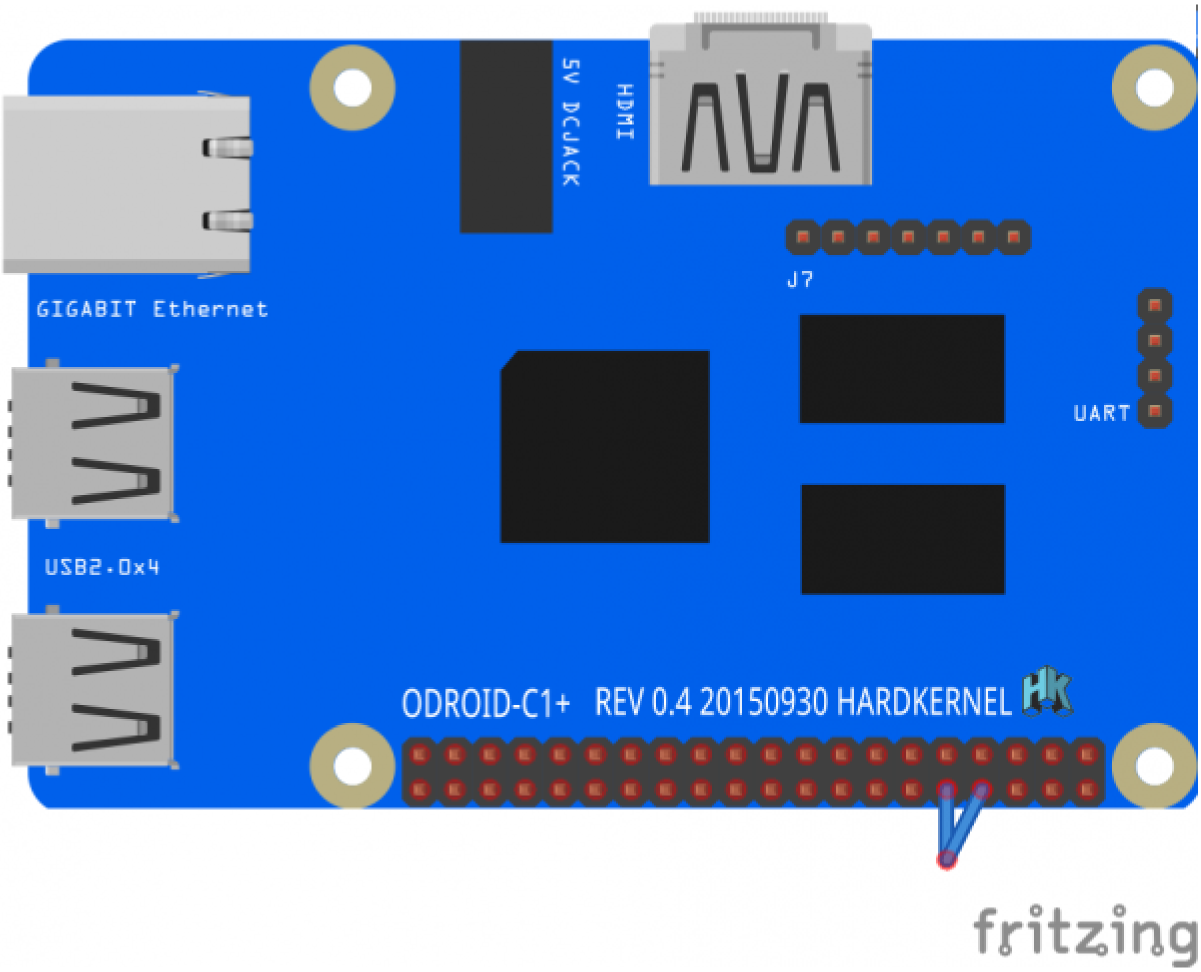
ODROID-XU4 (+Shifter Shield)

Please refer ShiftShield for more detail.



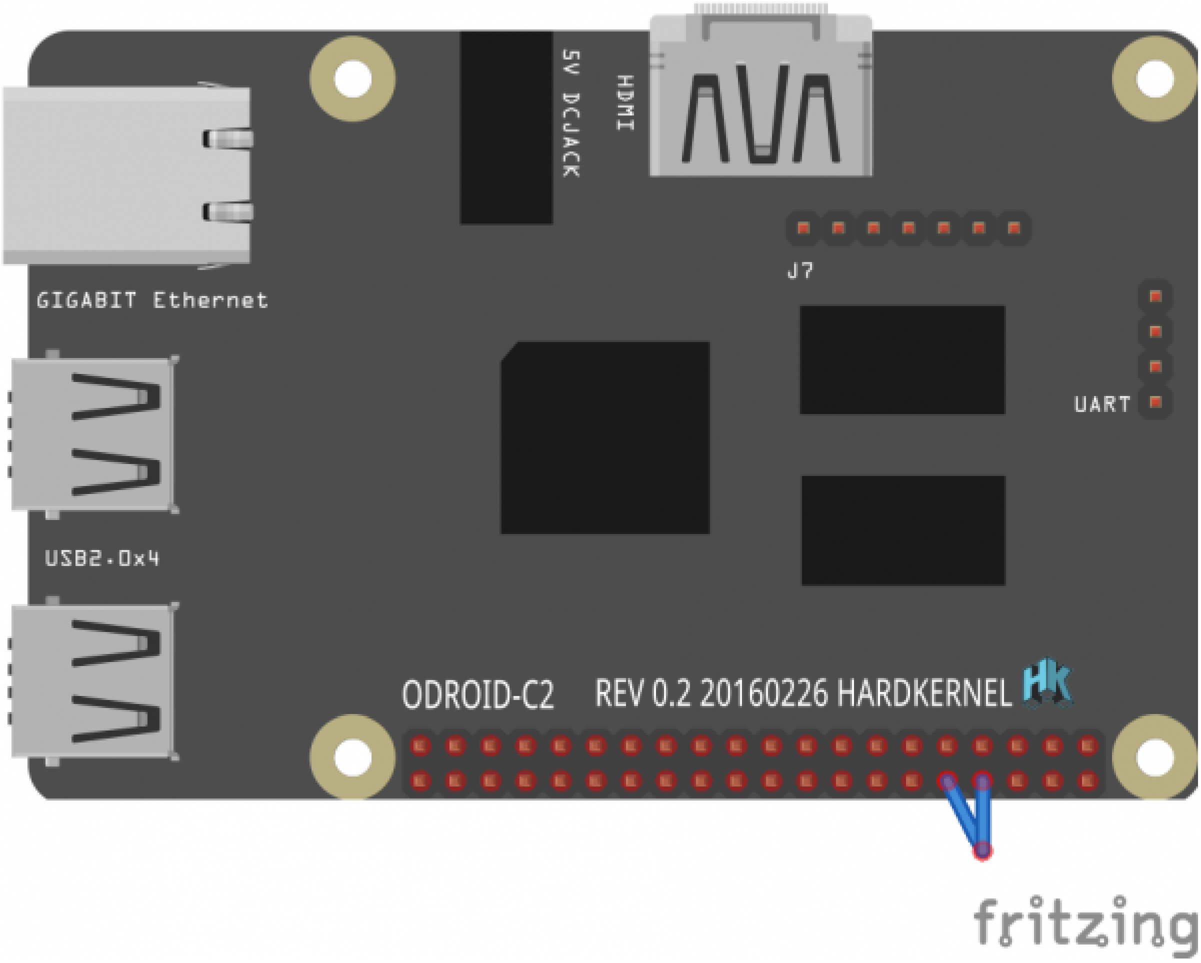
- Download Fritzing Example File: xu4_serial_loopback.fzz

ODROID-C1+



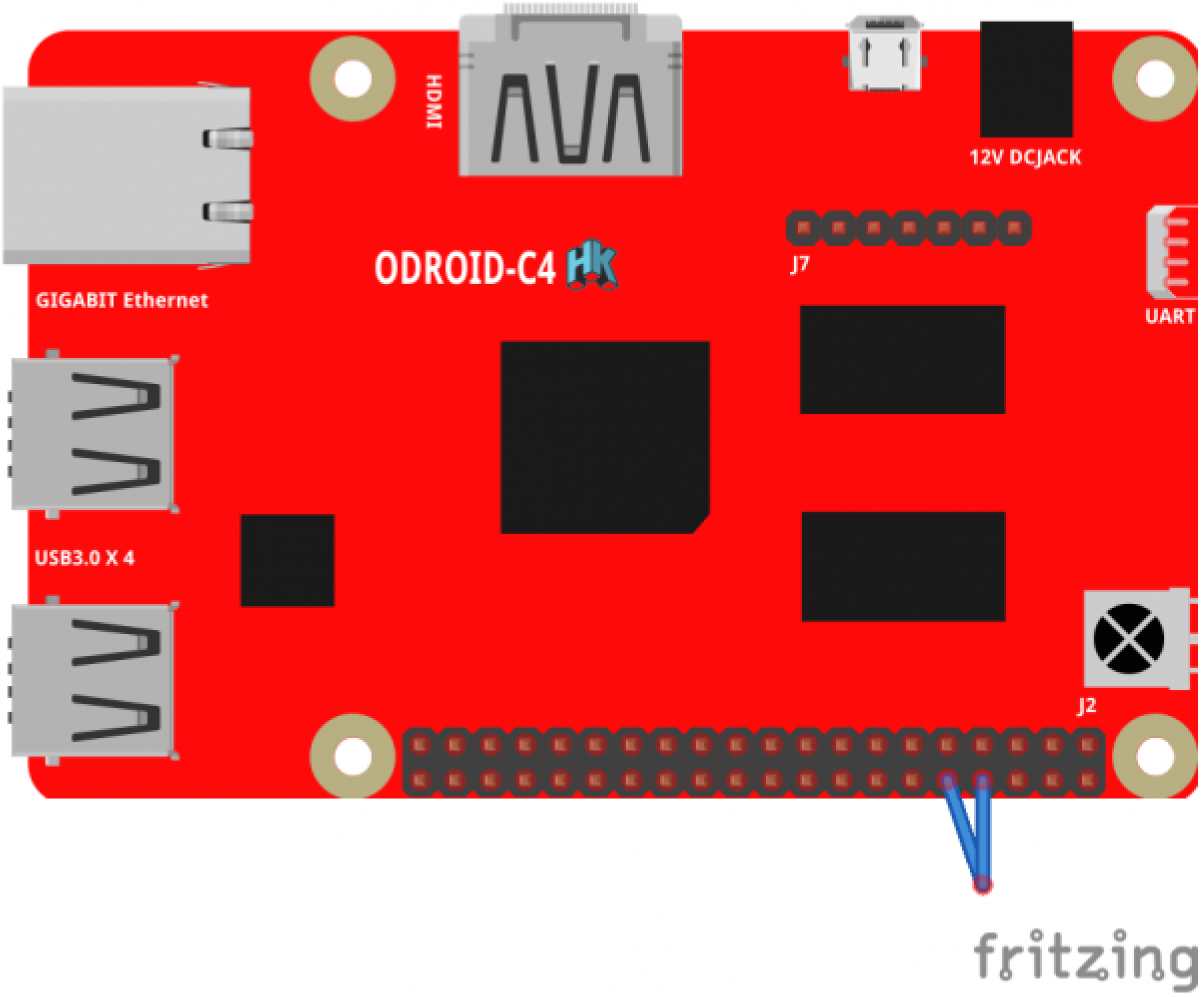
- Download Fritzing Example File: c1_serial_loopback.fzz

ODROID-C2



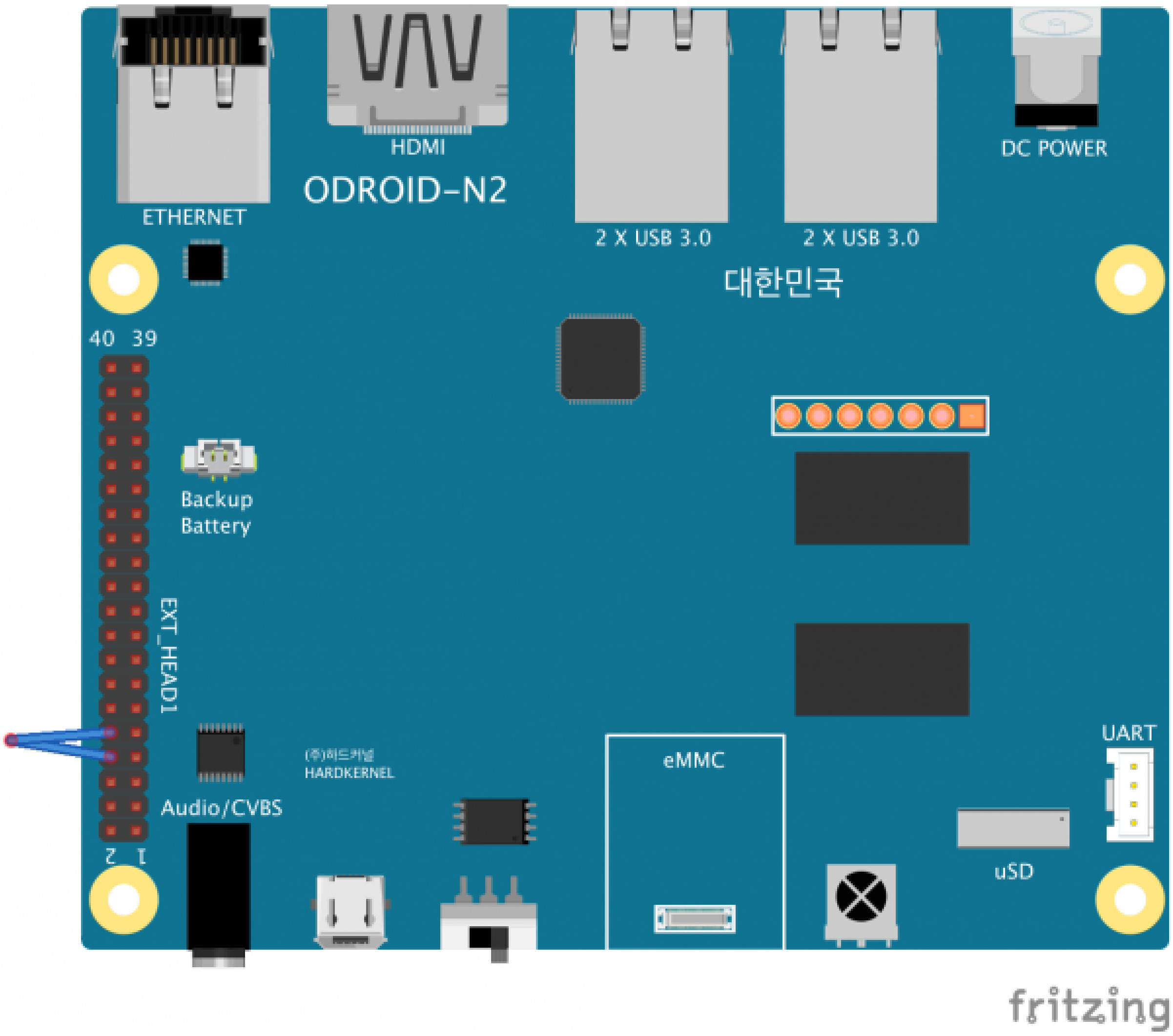
- Download Fritzing Example File: c2_serial_loopback.fzz

ODROID-C4



- Download Fritzing Example File: odroid-c4.fzpz

ODROID-N2



- Download Fritzing Example File: n2_serial_loopback.fzz

- The serial device file name is different on each boards.
  - ODROID-XU4: **/dev/ttySAC0**
  - ODROID-C1/C1+: **/dev/ttyS2**
  - ODROID-C2: **/dev/ttyS1**
  - ODROID-N2: **/dev/ttyS1**
  - ODROID-C4: **/dev/ttyS1**

C

target

```
# C
$ gcc -o wpi_exam_serial_loopback wpi_exam_serial_loopback.c $(pkg-config --cflags --libs libwiringpi2)
$ ./wpi_exam_serial_loopback
```

wpi_exam_serial_loopback.c

```
#include <wiringSerial.h>
#include <wiringPi.h>
#include <stdio.h>

int main(int argc, char ** argv)
{
        int fd = serialOpen("/dev/ttyS0", 115200);
        char buf[255];

        while (1) {
                printf("Serial Input> ");
                fgets(buf, 255, stdin);
                serialPuts(fd, buf);
                delay(10);
                printf("Serial Output> ");
                while (serialDataAvail(fd))
                        putchar(serialGetchar(fd));
        }
}
```

Python 2

target

```
# Python 2
$ sudo python wpi_exam_serial_loopback.py

# Will result like..
$ sudo python wpi_exam_serial_loopback.py
Serial Input> test
Serial Output> test
Serial Input>
```

wpi_exam_serial_loopback.py

```
#!/usr/bin/env python
import odroid_wiringpi as wpi
import time

serial = wpi.serialOpen('/dev/ttyS0', 115200)

while True:
        input_str = raw_input('Serial Input> ')
        wpi.serialPuts(serial, input_str)
        time.sleep(0.1)
        output_str = 'Serial Output> '
        while wpi.serialDataAvail(serial):
                output_str += chr(wpi.serialGetchar(serial))
        print output_str

wpi.serialClose(serial)
```

References

[1] http://wiringpi.com/

2020/03/18 14:27 · luke.go