

ODROID-C4 with mini DC UPS

This wiki page has almost the same as [ODROID-N2 with mini DC UPS](#). but using ODROID-C4. Some pictures and writing come from the above wiki.

Preparation Material

1. [12V2A 22.2W UPS Uninterrupted Power Supply](#). You can use any other 12Volt DC UPS if its output is 2Amp or higher.

Let's calculate how many hours the battery can endure supplying to the ODROID-C4.

Assume that ODROID-C4 is running while eating 200mA current.

The battery has its specification output 12V and 22.2wH(watt-hour).

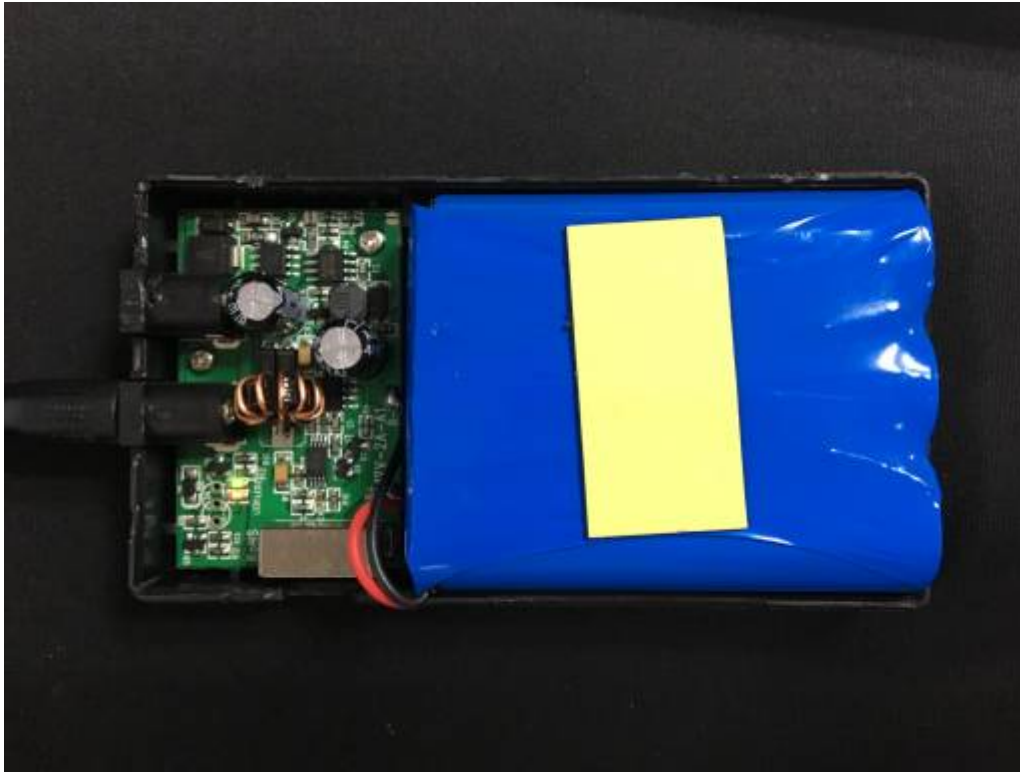
$22.2\text{wH} / (12\text{V} \times 200\text{mA}) = 9.25 \text{ Hours}$ (In principle, but there is lots of environment out there to get decreased it in the real world)

2. ODROID-C4 at [hardkernel.com](#)
3. [USB IO Board](#)
4. 10K axial resistor
5. 10K Potentiometer
6. Some wires

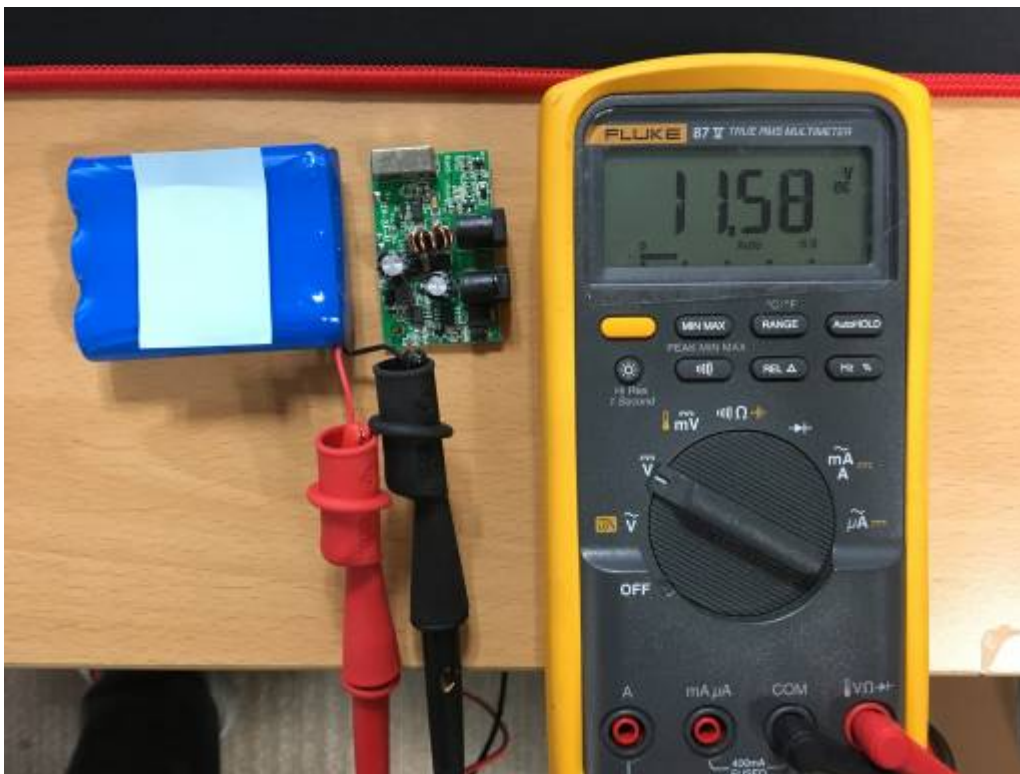
Step 1. Disassemble mini DC UPS and Wiring

There was damaged the part of Power ON/OFF switch on the UPS when I disassembling it.(My fault 😬)





Checked the battery voltage when the battery is fully charged was 11.58V. At the same time, the output voltage on the connector was 12.22V. I guess there is a boost IC placed between the battery and the output connector to make sure over 12V output when it is working.





Soldering a wire at the same pad as the battery on the PCB together to check how level remains in the battery.

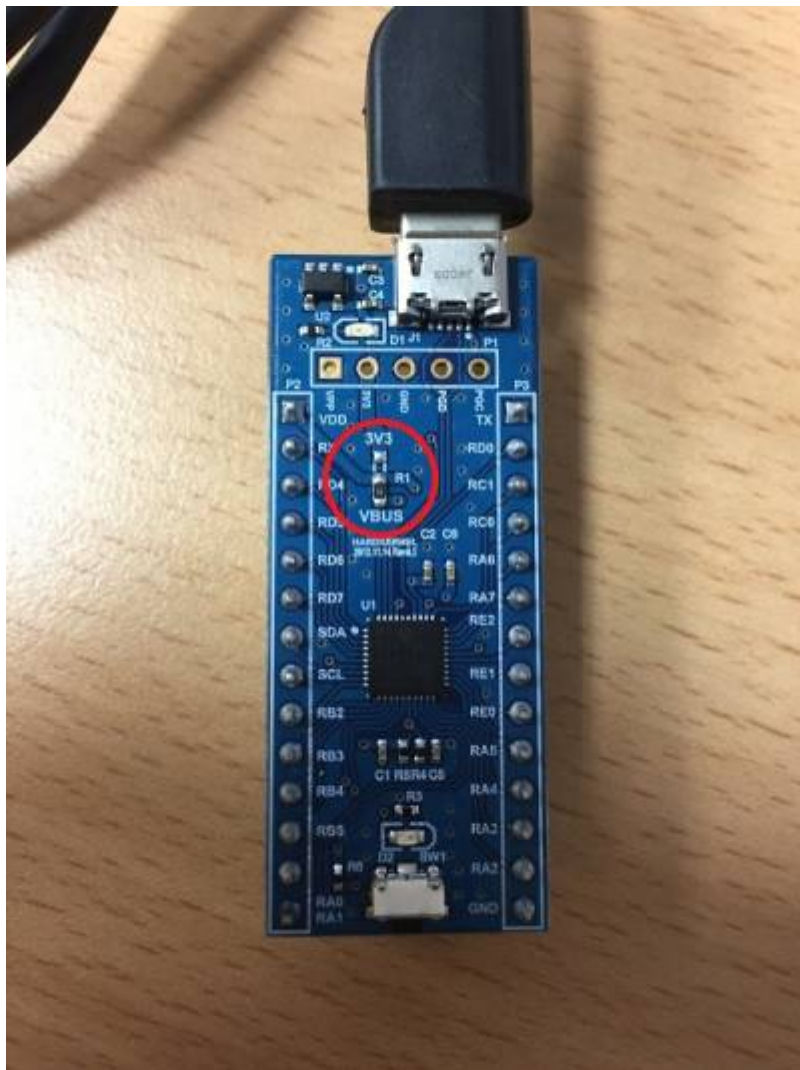
Step 2. ADC Reference voltage changing under USB_IO board

USB IO Board can choose either 3.3v(default) or 5v supply by the position of R1. It also becomes ADC reference voltage.

Check it out [USB IO Board](#)

I have decided to use 5V ADC reference voltage. It has better get resolution than 3.3V and gets easy calculating.

so, I soldered the R1 resistor to VBUS 5V from 3V3 on the PCB.



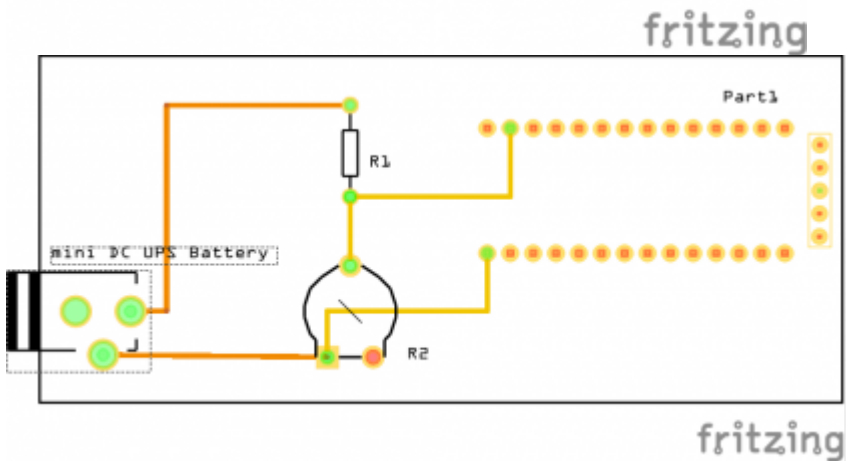
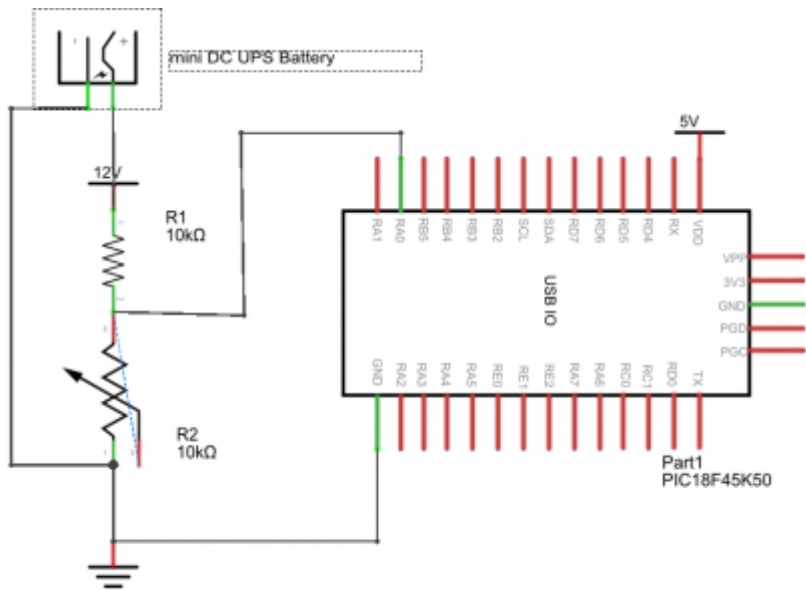
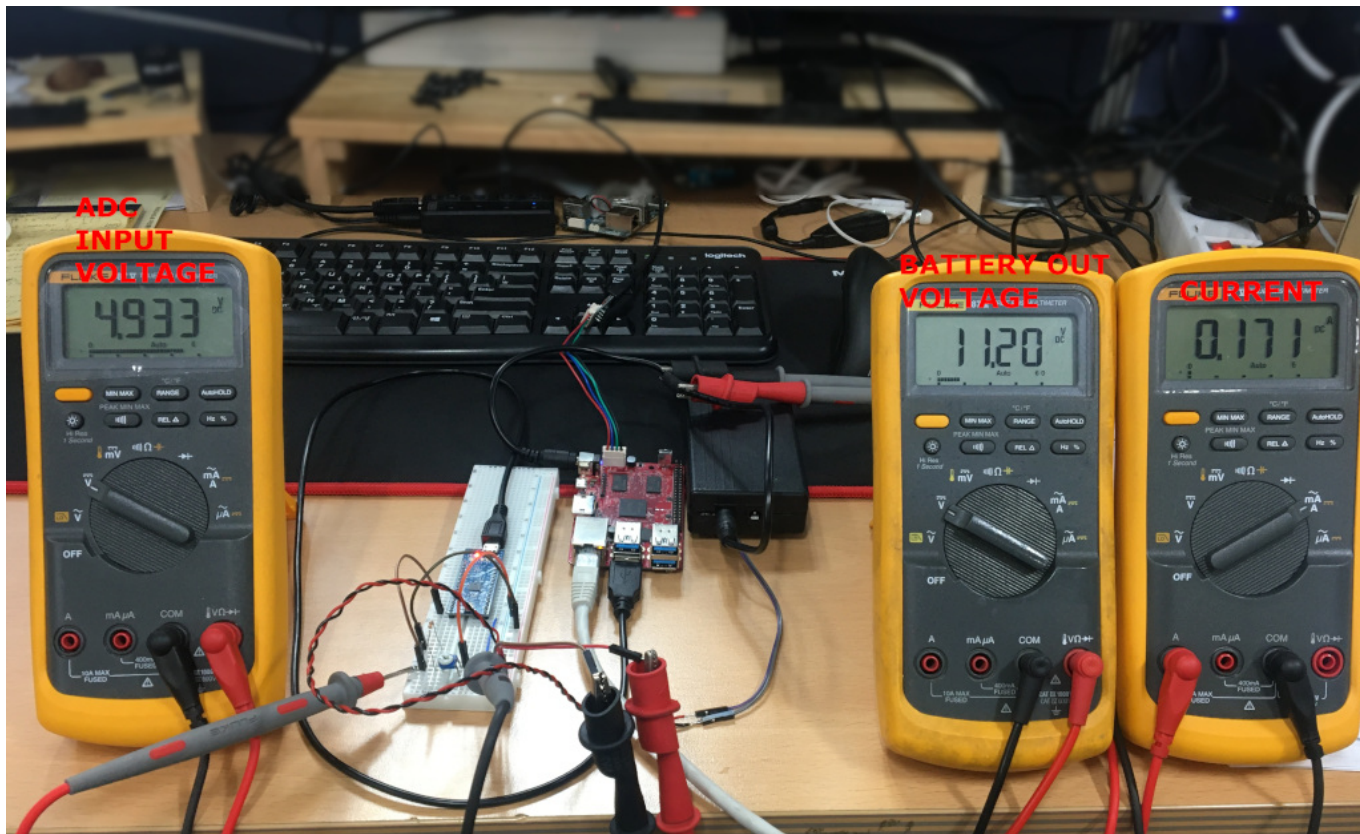
Step 3. Define value of resistors

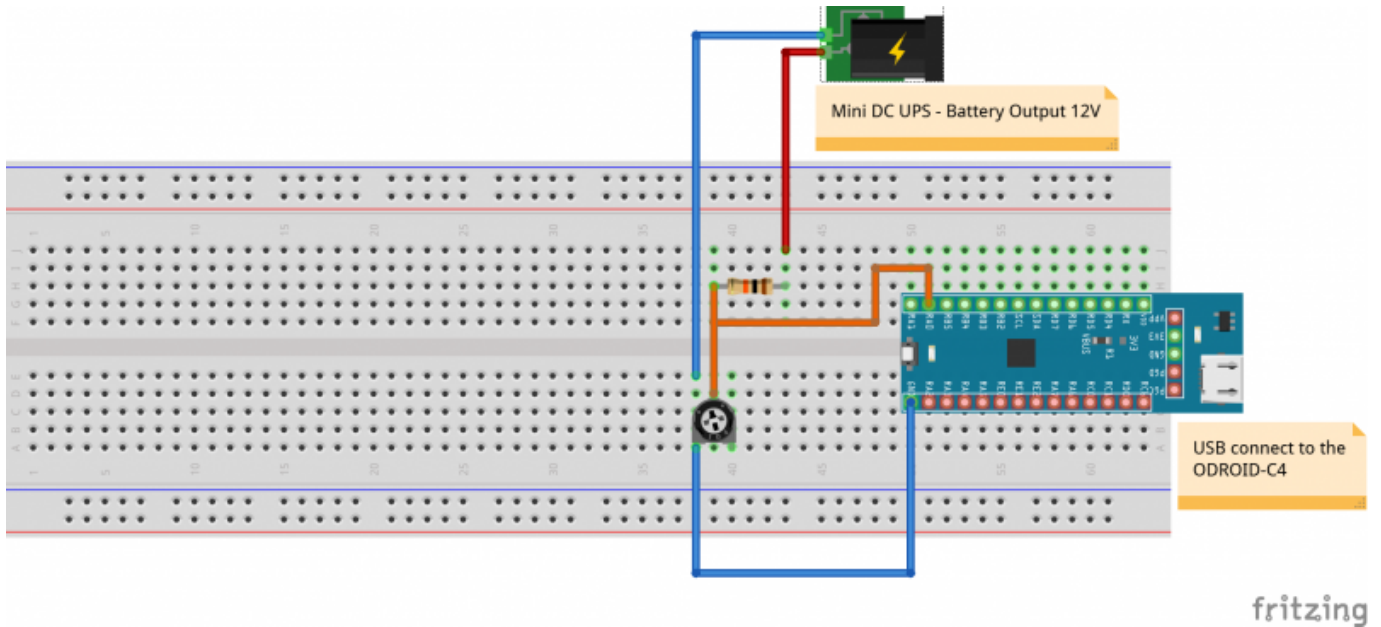
Using two resistors 10KOhm(R1) and 7.143KOhm(R2), We can be divided 5V and 7V from 12V in principle.



$12V \times (R2 / (R1 + R2)) = 5V$
If R1 is 10,000 Ohm, R2 is about 7,143 Ohm.

However, there is not 7,143 Ohm out there, so I am using a 10K potentiometer
Now I've measured when I got the divided 5V, my R1 is 9.98 KOhm and R2 is 7.04 KOhm that means the battery of mini DC UPS is a few under 12V when I charged fully.
I manipulated to increase the R2 value a little more to get ADC value full 10bit value 1024.





[Fritzing download](#)

minidcups_usbio_c4.fzz

}}}

Step 4. Define the Maximum & Minimum ADC value

Build software as following the below instruction on ODROID-C4.

```
$ sudo apt-get install libusb-1.0-0-dev git rdate
$ git clone https://github.com/hardkernel/Odroid-USBIO
$ cd Odroid-USBIO/usbio/linux
$ make
$ sudo ./usbio
```

- Toggle LED
- AN0/POT Value
- Button status
- Get Register
- Set Register
- Get Register Bit
- Set Register Bit
- Exit

```
msb = 512, lsb = 209
potentiometerValue = 721
```

I've set the maximum ADC value is 1023(10 bits all high) by manipulating R2 when mini DC UPS is charged fully.

We have to know the minimum ADC value to see the remaining battery level.

I've acquired the minimum ADC value **721(About 3.52 volt)** by giving some load like stress app to the system until ODROID-C4 turned off supplied power using mini DC UPS only.

This script helps me to get minimum ADC value.

```
root@odroid:~/usbio/Odroid-USBIO/usbio/linux# cat -n batCheck.sh
 1 #!/bin/bash
 2 sleepSec=2
 3 for i in {1..100000}
 4 do
 5 ./usbio << endl >> ./adcValue.log
 6 b
 7 q
 8 endl
 9 # elapsedTime=`expr ${i} \* ${sleepSec}`
10 echo -n "${i} : "
11 /usr/bin/rdate time.bora.net # Need network connecting
12 sleep ${sleepSec}
13 # echo "`date +%Y/%m/%d-%H:%M:%S` : ${i}"
14 done
root@odroid:~/usbio/Odroid-USBIO/usbio/linux# nohup ./batCheck.sh &
```

I've got the minimum ADC value **721**.

And the extra that how many hours the full charged UPS can supply to the ODROID-C4 without power input is about **1 hour and 30 minutes**.

(It was fully loaded running the batchcheck.sh script and the command "# stress-ng -cpu 4 -cpu-method matrixprod" simultaneously)

```
root@odroid:~/Odroid-USBIO/usbio/linux# grep -rn "potentiometerValue"
adcValue.log | awk -F"=" '{print $2}' | sort -nr | tail
730
729
729
727
727
726
725
724
723
721

root@odroid:~/Odroid-USBIO/usbio/linux# head -n 1 nohup.out
1 : Tue Mar 31 05:24:18 UTC 2020
root@odroid:~/Odroid-USBIO/usbio/linux# tail -n 2 nohup.out
2541 : Tue Mar 31 06:53:31 UTC 2020
```

Step 5. Remaining battery level and TO DO more

Now, We figured out the maximum and the minimum ADC value. Therefore, we can calculate the remaining battery level.



**Remaining battery level(%) =
(ADC value - minimum ADC value) x 100 / (1023 - minimum ADC value)**

And I want to show you that before arriving the minimum ADC value, it is going to shut down ODROID-C4 safely.

This is an example shell script how to make an application with it.

As I said before, My minimumADC value is 721 by experimentation, so I set the minimumADC value is 800 by a wide margin in this script.

If remaining battery level of mini DC UPS is lower than 10 percent as I set `${minRemainBat}` in the script, it is going to invoke shutdown procedure.

```

1 #!/bin/bash
2
3 logFile=/var/log/batADC.log
4 minimumADC=800
5 minRemainBat=10 # 10% percent
6
7 while true
8 do
9
10 ./usbio << endl >> ${logFile}
11 b
12 q
13 endl
14
15 if [ -f ${logFile} ];
16 then
17     ADCValue=`cat /var/log/batADC.log | grep "potentiometerValue" |
awk -F" " '{print $3}' | tail -1`
18
19     if [ ${ADCValue} -gt 0 ];
20     then
21
22         # remainBat=`expr \( \(${ADCValue} - ${minimumADC} \) / \(
1023 - ${minimumADC} \) \) '*' 100`
23         remainBat=`expr \( \(${ADCValue} - ${minimumADC} \) '*' 100 /
\ ( 1023 - ${minimumADC} \) `
24
25         echo "ADC value : ${ADCValue}"
26         echo "Minimum ADC value$ : ${minimumADC}"
27         echo "Remaining Battery level : ${remainBat}%"
28
29         if [ ${remainBat} -lt ${minRemainBat} ];
30         then
31             echo "Shut down system now"
32             sleep 1
33             /sbin/poweroff
34         else

```

```
35      echo "Delete batteryADC log"
36      rm -f ${logFile}
37      sleep 1
38          fi
39      fi
40  fi
41  sleep 60
42
43  done
```

From:
<https://wiki.odroid.com/> - **ODROID Wiki**

Permanent link:
https://wiki.odroid.com/odroid-c4/application_note/odroid-c4_with_mini_dc_ups

Last update: **2020/05/26 14:04**

