

MicroPython for ODROID-GO - Battery

- Make sure that you've followed these guides:
 - [MicroPython setup](#)
 - [MicroPython for ODROID-GO - Hello World](#)

- Refer to the [MicroPython for ESP8266 official documents](#).
 - <http://docs.micropython.org/en/latest/esp8266/>

We will learn how to get the status of the battery with [MicroPython](#) in this guide. The LCD will display how much battery voltage remains in volts.

Get the status of the battery



ODROID-GO has a ~3.7V battery module.

We can read the battery level through one of the **12-bit SAR ADCs** which are integrated in ESP32.

These ADCs are:

- ADC1: 8 channels, attached to GPIOs 32-39.
- ADC2: 10 channels, attached to GPIOs 0, 2, 4, 12-15, 25-27.

There are some restrictions for an application's use of ADC2.

The battery is attached to GPIO pin number **36**, so we should read a value from that using **ADC1**. Fortunately, we can use ADC1 with just putting pin number(32-39) on [MicroPython](#) using **machine.ADC** module.

In this guide, we're going to use that to read the current battery level in volts and display it on the LCD.

First, prepare the code like below to display on the LCD.

We are going to refresh the screen every 1 second so we need to import **time** module too.

```
from odroid_go import G0
import time

G0.lcd.set_font(G0.lcd.fonts.TT24)

def show_battery_voltage():
    # need to fill out

while True:
    show_battery_voltage()

    time.sleep(1)
```

And we defined one function **show_battery_voltage()** for showing current voltage on the screen.

Before filling it out, it's important to know the GPIO battery voltage is divided by 2 due to the input limitation of the integrated ADC.

So, if the original value coming from the battery is 3.7V, then the input value to the GPIO pin will be about 1.85V.

Thus, we have to **multiply the value by 2** to know the actual voltage.

We use **12 bit SAR ADC** for the channel with **11 dB attenuation**. We should use these rates to calculate the result as well.

The width and attenuation can be set with

- **machine.ADC.width()**
- **machine.ADC atten()**

so that we can get a voltage value correctly.

We prepared a battery module for ODROID-GO which has **get_voltage()** function for easy use. So the code using the module will be shown like the below.

```
from odroid_go import G0
import time

G0.lcd.set_font(G0.lcd.fonts.TT24)

def show_battery_voltage():
    G0.lcd.erase()
    G0.lcd.set_pos(, )

    G0.lcd.print("Current Voltage: " + str(G0.battery.get_voltage()))

while True:
    show_battery_voltage()

    time.sleep(1)
```

Note that,

- Each ESP32 chip has its **Vref value for calibrating an ADC value** and has a function for getting correct battery remains. But in [MicroPython](#), we can't use them yet.
- To get an ADC value more correctly, consider sampling a reading value. When **get_voltage()** function in ODROID-GO module called, it reads a ADC value 64 times and returns its average.

Save and overwrite this code as **boot.py** in the **ODROID-GO module installation directory**.

Try it on your ODROID-GO

- To execute this module properly, make sure you've uploaded ODROID-GO module. And you have to upload the written file called **boot.py** using **rshell** or **ampy**.
 - If you uploaded properly, [MicroPython](#) will execute **boot.py** when the device boots automatically.
 - Please refer to setup guide to further information: [Install the ODROID-GO MicroPython module](#).
 - Or you also can do them in **REPL**. Write the codes line by line in order.

Upload the **boot.py** file using **rshell** or **ampy**, enter to **REPL** prompt, and **restart** ODROID-GO.

Then, press any button to show **"Pressed"** string besides that button.

As you noticed, you need to keep pressing it until the LCD is updated due to the **time.sleep(1)** function.

A completed example

The complete example is available in following path:

- [odroid_go/examples/button/battery.py](http://wiki.odroid.com/odroid_go/examples/button/battery.py)

Copy and paste to try the example.

From:

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

Permanent link:

http://wiki.odroid.com/odroid_go/micropython/05_battery

Last update: **2018/07/23 06:05**

