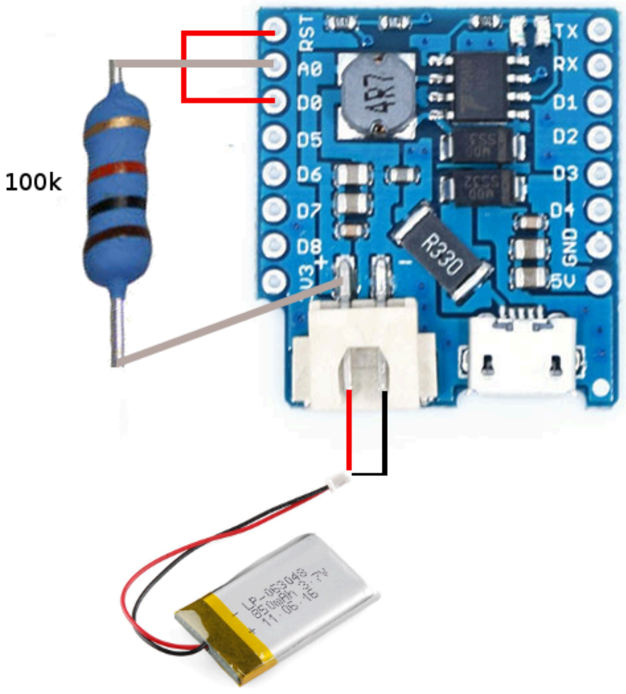
Arduino, ESP8266, ESP32 & Raspberry Pi stuff

Arduino and related stuff (including Attiny and ESP8266) and the Raspberry Pi

Monitoring LiPo battery voltage with Wemos D1 mini-battery shield and Thingspeak

There are a million reasons why you would want to monitor the Battery voltage of your Battery fed ESP8266. I will illustrate it with a Wemos D1 mini and the Battery shield

I am using a small 720 mAh LiPo cel. If I just leave the Wemos access the internet continuously it will last 6.5 hours, but for this example I will put the Wemos in Deepsleep for a minute, then read the battery voltage and upload that to Thingspeak.
You only need to make a few connections:
First, connect RST with GPIO16 (that is D0 on the Wemos D1 mini). This is needed to let the chip awake from sleep.
Then connect the Vbat through a 100k resistor to A0.



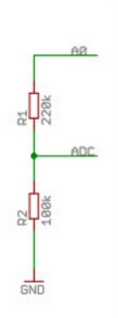
Wemos D1 Mini Battery shield

So why a 100 k resistor?

Well the Wemos D1 mini already has an internal voltage divider that connects the A0 pin to the ADC of the ESP8266 chip. This is a 220 k resistor over a 100 k resistor

By adding a 100k , it will in fact be a total resistance of 100k+220k+100k=420k.
So if the Voltage of a fully loaded Cell would be 4.2 Volt, the ADC of the ESP8266 would get 4.2 * 100/420= 1 Volt

1 Volt is the max input to the ADC and will give a Raw reading of 1023.



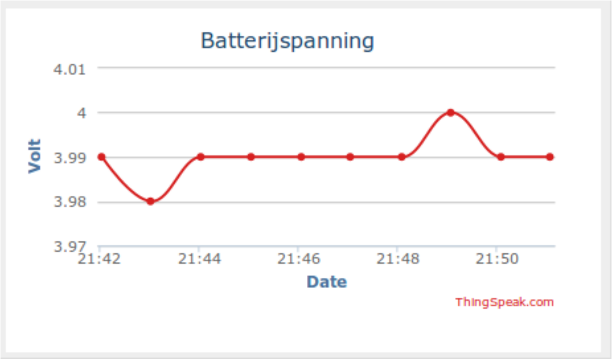
Wemos D1 Internal Voltage divider

The True voltage then can be calculated by:
raw = AnalogRead(A0);voltage =raw/1023;
voltage =4.2*voltage;
Ofcourse you could also do that in one step, but I like to keep it easy to follow.

If you do use this possibility, do realise that the resistors drain the battery as well with a constant 10uA (4.2V/420 000ohm). The powerconsumption of an ESP8266 in deepsleep is about 77uA. With the battery monitor this would be 87uA, which is a sizeable increase. A solution could be to close off the Vbat to the A0 with a transistor, controlled from an ESP8266 pin

A program could look like this:

```
/*
 * Wemos battery shield, measure Vbat
 * add 100k between Vbat and ADC
 * Voltage divider of 100k+220k over 100k
 * gives 100/420k
```



Wemos Battery monitoring

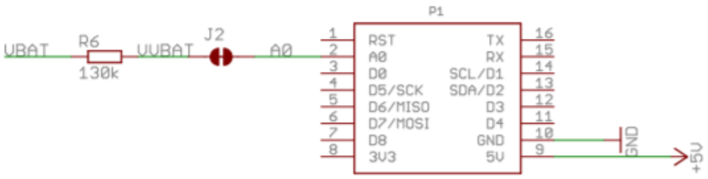
```
 * ergo 4.2V -> 1Volt
 * Max input on A0=1Volt ->1023
 * 4.2*(Raw/1023)=Vbat
 */

// Connect RST en gpio16 (RST and D0 on Wemos)
#include <ESP8266WiFi.h>
unsigned int raw=0;
float volt=0.0;
// Time to sleep (in seconds):
const int sleepTimeS = 60;

void setup() {
  Serial.begin(115200);
  Serial.println("ESP8266 in normal mode");
  const char* ssid     = "YourSSID";
  const char* password = "YourPW";
  const char* host = "api.thingspeak.com";
  const char* writeAPIKey="YourAPIkey";
  // put your setup code here, to run once:
  pinMode(A0, INPUT);
  raw = analogRead(A0);
  volt=raw/1023.0;
  volt=volt*4.2;
//  Connect to WiFi network
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  }
  String v=String(volt);// change float into string
  // make TCP connections
  WiFiClient client;
  const int httpPort = 80;
  if (!client.connect(host, httpPort)) {
    return;
  }
  String url = "/update?key=";
  url += writeAPIKey;
  url += "&field6=";// I had field 6 still free
that's why
  url += String(volt);
  url += "\r\n";

// Send request to the server
  client.print(String("GET ") + url + " HTTP/1.1\r\n"
+
                 "Host: " + host + "\r\n" +
                 "Connection: close\r\n\r\n");

 //Sleep
   Serial.println("ESP8266 in sleep mode");
   ESP.deepSleep(sleepTimeS * 1000000);
}

void loop() {
  //all code is in the Setup
}
```

The new battery shield

There now is a new version (V1.2.0) of the battery shield that has an inbuilt resistor connecting A0 to the battery, through Jumer J2 ('Jumper' being a big word for 2 solderpads), so if you want to measure the battery voltage, all you need to do is to put some solder connecting J2

However, rather than a 100k resistor, a 130k resistor was used. The voltage divider thus becomes 100/(130+220+100), so for a full reading of 1023 (=1Volt on A0) a total of (1/100k)*(130+220+100)=4.5Volt would be necessary.

In reality the Lipo cell will not give off 4.5Volt.



E / December 25, 2016 / Uncategorized

84 thoughts on “Monitoring LiPo battery voltage with Wemos D1 minibattery shield and Thingspeak”

Jeroen

December 26, 2016 at 14:47

Depending of course on what your project is required to do, my advice would be to put the thing in the deepest possible sleep once it hits the low end of the battery voltage, maybe only to sound a piezo buzzer for say 200 ms when it shortly wakes up. That will be the safest bet to keep your LiPo alive as long as possible. The cell shown probably has a low voltage protection circuit build in though. Still, the warning is helpful.

Sidenote: The other day I accidentally miswired a cell like that (600 mA version), reversed polarity. In a few ms, the magic smoke appeared and the cell seemed to have lost quite a bit of it’s capacity. Hard to believe things going south that fast. Turned out the protection circuit had fried a bit. Without it, the cell runs fine with 550 mAh betweem 3.0 and 4.2 volt.

Arduino

December 29, 2016 at 03:02

I agree. It is quite easy to do that. My intention ofcourse is that it will not get too low because of a solar cell that is to provide enough input.
esp.deepsleep is as far as i know the deepest sleep, but when the battery voltage hits 3.0 it might be wise to let the deepsleep last as long as possible and do no uploads anymore.
I am not sure at what voltage the esp stops working, but it does still work at 3Volt, and will proceed to discharge the cell. After 3.5 days my cell has discharged till 3.7 Volt. Obviously that still can be better, but the battery shield is not the most efficient. I will see how long it will take for the last 0.7Volt. Will repeat it with a bare ESP8266-12

Yes, some components really dont like reversed potential, but seems you still had some luck

ronaldogabrielrapado

October 21, 2018 at 02:54

Hello friend sketch is giving error in line volt of float = 0,0; help

E

October 23, 2018 at 12:13

as you seem to direct this to Jeroen, I am not sure what sketch you refer to?

Jeroen

February 7, 2017 at 10:56