

Lesson

7

Voltage Measure



Introduction:

There are three parts in this lesson. In order to prevent battery damage caused by over discharge, it is necessary to monitor the voltage. So we will teach you how to measure the battery voltage.

Tumbler Voltage Measure

1.1 Hardware Design

1.2 Software Design

1.3 Upload Validation

Preparations:

one car (with a battery)

one USB cable

1.1 Hardware Design

Our external power supply is to drive the motor, so the power cord is connected to the motor drive chip VM. (As shown in figure 1.1.1~1.1.2)

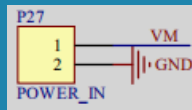


Figure 1.1.1 The schematic of the power

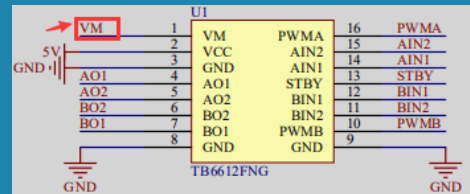


Figure 1.1.2 The schematic of the TB6612FENG

So how do we measure the voltage?

We set the pin of Arduino to collect battery voltage as A2, because the internal reference voltage of Arduino is 1.1V. That is to say, no matter how large the power supply voltage is, the voltage corresponding to the value of 0 ~ 1023 sampled by ADC is 0V ~ 1.1V, and it is still 1023 when the voltage exceeds 1.1V. The AVR document also indicates that there is a certain deviation in this benchmark value. However, the reference voltage of 1.1V is too low, so the measured voltage must be smaller than it. So let's divide the voltage to be tested. I used a 10k resistor and a 1.5k resistor in series, and then connected A2 pin to the middle of the two resistors to measure, then the voltage value of 1 / 11 must be below 1.1V. (As shown in figure 1.1.3~1.1.4)

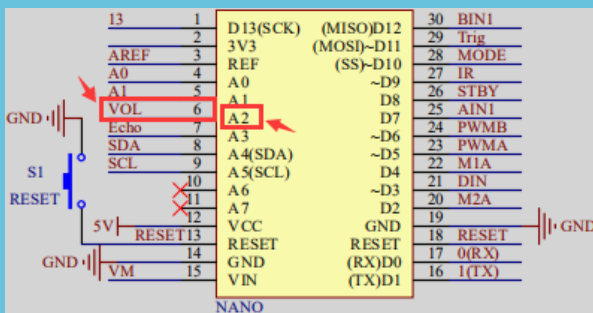


Figure 1.1.3 The schematic of the A2 on Nano



Figure 1.1.4 The schematic of Voltage divider circuit

1.2 Software Design

Please open [ELEGOO Tumbler Self-Balancing Car Tutorial -> Lesson 7 Voltage Measure -> VoltageMeasure-> VoltageMeasure.ino](#) in the current folder.

First, we need to set the reference voltage.

```
void voltageInit()
{
  analogReference(INTERNAL);
}
```

Then calculate the power supply voltage according to the formula.

$$V_{out} = (V_{in} \times R2) / (R1 + R2)$$

Then the `analogread()` function is used to collect the value of A2. We know that the Arduino controller has multiple 10 digit A / D channels. This means that Arduino can map the voltage input signal of 0-5V to the value of 0-1023. In other words, we can divide 5V signal into 1024 parts. The input signal of 0V corresponds to the value 0, and the input signal of 5V corresponds to 1023. In order to realize voltage monitoring, it is not enough to use the function of `analogread()` to read the A4 value. We need to convert the reading value to the actual voltage value. Therefore, we should use the following formula to convert:

$$\text{Battery voltage } V = \text{A4 read value} * (5.00 / 1024.00) * \text{multiplier}$$

Sum up:

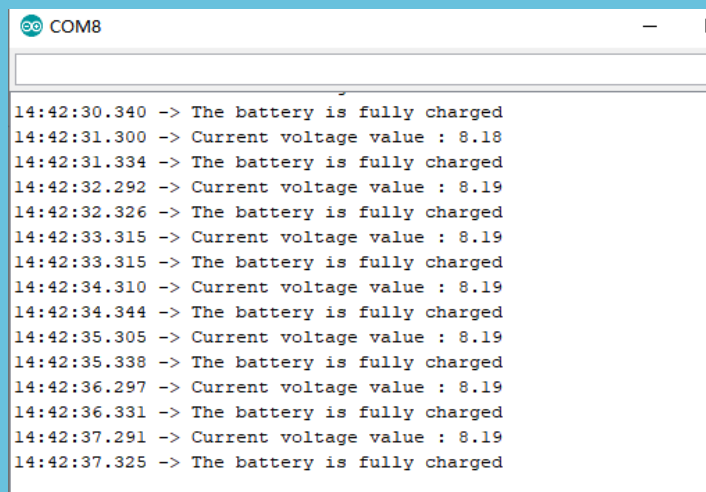
$$\text{Battery voltage } V = \text{A4 read value} * (5.00 / 1024.00) * ((R1 + R2) / R2)$$

We set it to measure once per second.

```
void Voltage_Measure()
{
  if (millis() - vol_measure_time > 1000) //Measured every 1000 milliseconds
  {
    vol_measure_time = millis();
    double voltage = (analogRead(VOL_MEASURE_PIN) * 1.1 / 1024) * ((10 + 1.5) / 1.5); //Read voltage value
    Serial.print("Current voltage value : ");
    Serial.println(voltage);
    if(voltage>7.8)
      Serial.println("The battery is fully charged");
    else
      Serial.println("Low battery");
  }
}
```

1.3 Upload Validation

After uploading the program and turning on the power, you can see the measured voltage value in the serial port. (As shown in figure 1.3.1)



```
COM8
14:42:30.340 -> The battery is fully charged
14:42:31.300 -> Current voltage value : 8.18
14:42:31.334 -> The battery is fully charged
14:42:32.292 -> Current voltage value : 8.19
14:42:32.326 -> The battery is fully charged
14:42:33.315 -> Current voltage value : 8.19
14:42:33.315 -> The battery is fully charged
14:42:34.310 -> Current voltage value : 8.19
14:42:34.344 -> The battery is fully charged
14:42:35.305 -> Current voltage value : 8.19
14:42:35.338 -> The battery is fully charged
14:42:36.297 -> Current voltage value : 8.19
14:42:36.331 -> The battery is fully charged
14:42:37.291 -> Current voltage value : 8.19
14:42:37.325 -> The battery is fully charged
```

Figure 1.3.1 Serial port display