Go Down Pages: [1] PRINT (HTTPS://FORUM.ARDUINO.CC/INDEX.PHP?ACTION=PRINTPAGE;TOPIC=616816.0)

Topic: [Solved] Arduino not reading the expected voltage on a voltage divider (Read 2190 times)

x.Php?Topic=616816.0;Prev_next=Prev#New) - Next Topic (Https://Forum.Arduino.Cc/Index.Php?Topic=616816.0;Prev_next=Next#New)

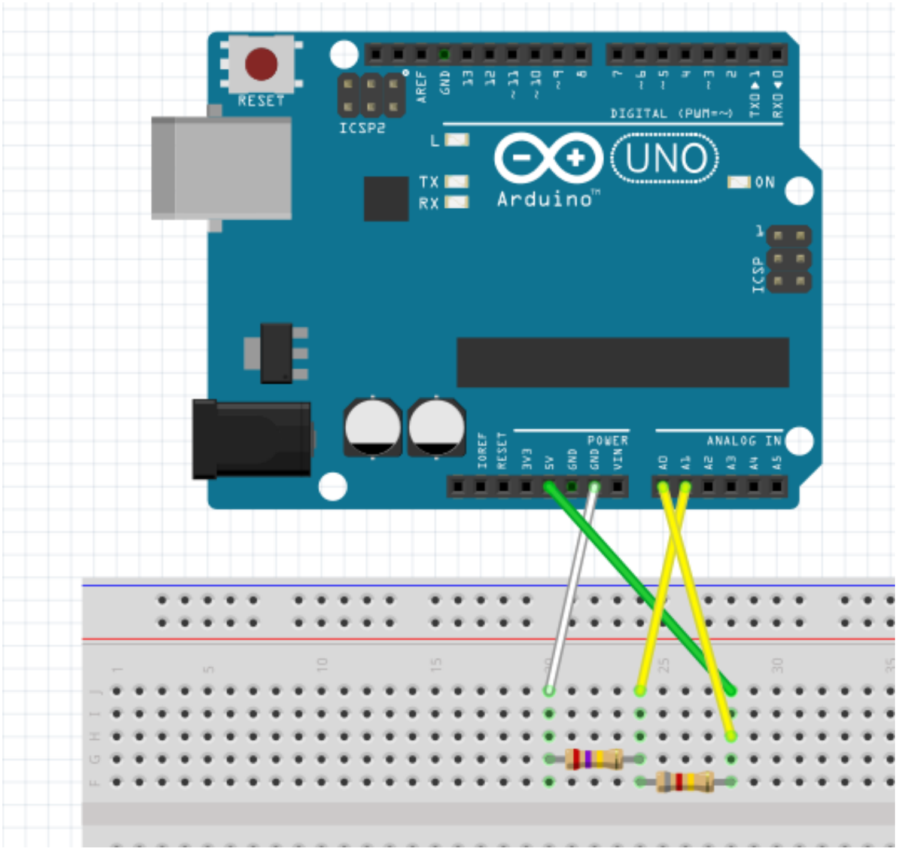
Benur21 (https://forum.arduino.cc/index.php?action=profile;u=1230730)

B (https://forum.arduino.cc/index.php?action=profile;u=1230730)

Newbie
Posts: 6
Karma: 0 [add] (https://forum.arduino.cc/index.php?action=karma;sa=applaud;uid=1230730;f3c5400d6c=3e6417ea98d8c4e1562fcdc3ecd054da)

[Solved] Arduino not reading the expected voltage on a voltage divider (https://forum.arduino.cc/index.php?topic=616816.msg4179656#msg4179656)
May 20, 2019, 03:35 am (https://forum.arduino.cc/index.php?topic=616816.msg4179656#msg4179656) Last Edit: May 21, 2019, 11:39 pm by Benur21

This is my wiring (270k resistor on the left, 820k on the right):



This is my code:

Code: [Select]

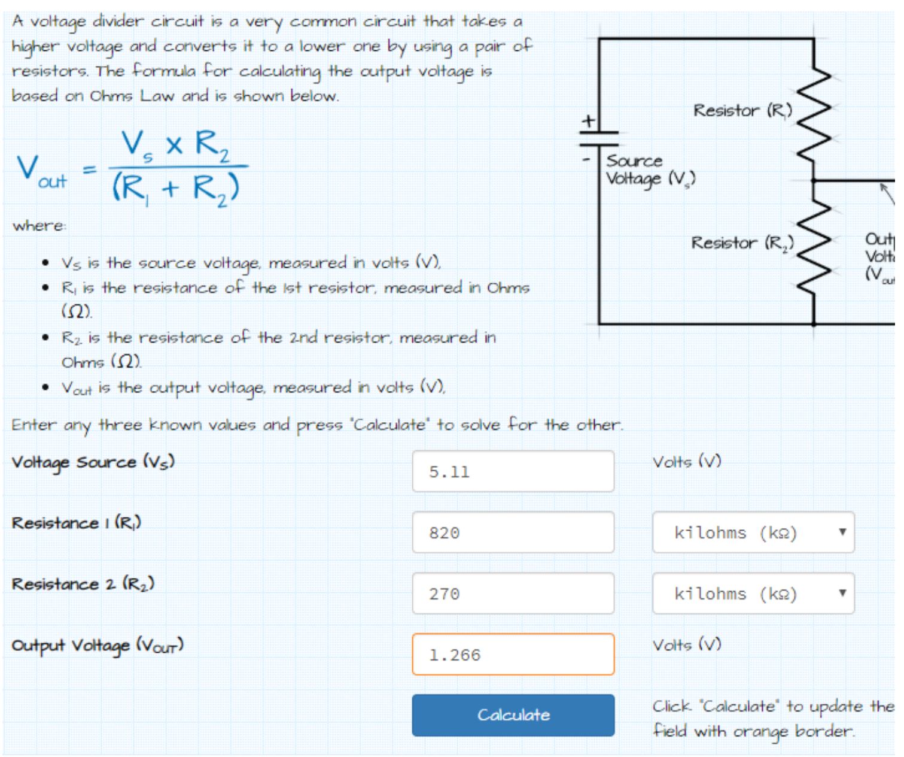
```
void setup() {
  Serial.begin(9600);
}

void loop() {
  analog_A0 = analogRead(A0) / 200.19;
  analog_A1 = analogRead(A1) / 200.19;
  Serial.println(str1 + analog_A0 + str2 + analog_A1 + str3);
}
```

So what I want to do is the arduino Uno be able to read voltages between 0 and ~20V like a voltmeter. I need to divide voltages by 4, so 20V is divided to 5V.

I read the first part of this article (https://startingelectronics.org/articles/arduino/measuring-voltage-with-arduino/) and I understood that I could use a voltage divider to do so, and the sum of its resistors could be about 1Mohm.

Then I calculated two good standard resistor values, 820k and 270k, and bought them.
You can see how much voltage results with a voltage divider like this, at this online calculator (http://www.ohmslawcalculator.com/voltage-divider-calculator):



A voltage divider circuit is a very common circuit that takes a higher voltage and converts it to a lower one by using a pair of resistors. The formula for calculating the output voltage is based on Ohms Law and is shown below.

$$V_{out} = \frac{V_s \times R_2}{(R_1 + R_2)}$$

where:

- V_S is the source voltage, measured in volts (V).
- R_1 is the resistance of the 1st resistor, measured in Ohms (Ω).
- R_2 is the resistance of the 2nd resistor, measured in Ohms (Ω).
- V_{out} is the output voltage, measured in volts (V).

Enter any three known values and press 'Calculate' to solve for the other.

Voltage Source (V_S) 5.11 Volts (V)
Resistance 1 (R_1) 820 kilohms (kΩ)
Resistance 2 (R_2) 270 kilohms (kΩ)
Output Voltage (V_{OUT}) 1.266 Volts (V)

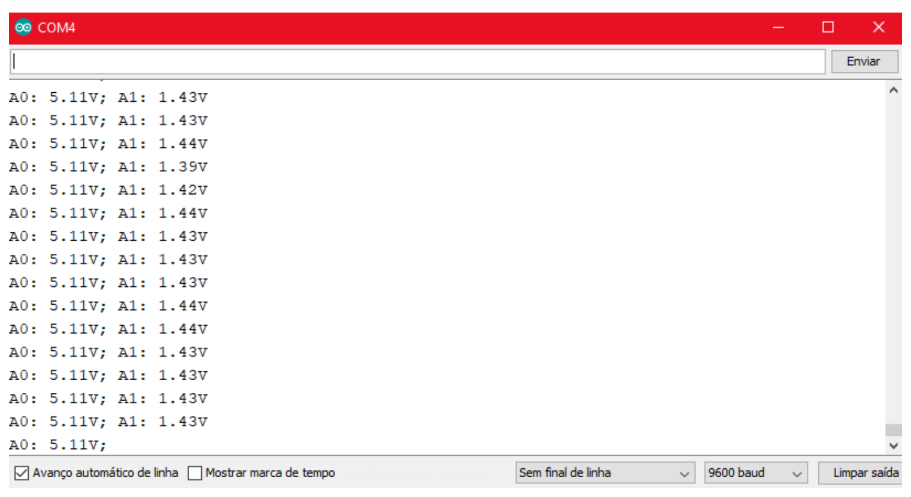
Resistor (R_1)
+
-
Source Voltage (V_s)
Resistor (R_2)
Output Voltage (V_{out})
Calculate
Click 'Calculate' to update the field with orange border.

Then I built the wiring as shown above and wrote the above code. I double checked wiring connections in case of any mistake.

The arduino Uno is connected to the computer via USB.

I measured the 5V pin on the arduino with a multimeter, and it reads ~5.11V, so I use this as the reference for the analogRead().
So the analogRead() returns 0-1023 values, and in case it returns 1023, the real voltage should be 1023 / 200.19 = 5.11V (hence the 200.19 in the code).

Now it should read 1.266V at A1 pin, but it is reading 1.43V:



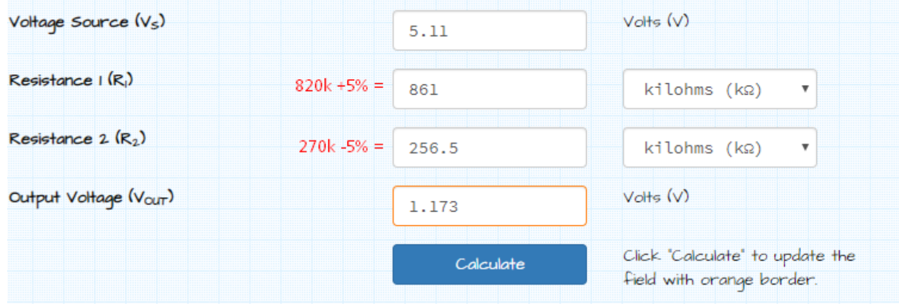
```
A0: 5.11V; A1: 1.43V
A0: 5.11V; A1: 1.43V
A0: 5.11V; A1: 1.44V
A0: 5.11V; A1: 1.39V
A0: 5.11V; A1: 1.42V
A0: 5.11V; A1: 1.44V
A0: 5.11V; A1: 1.43V
A0: 5.11V; A1: 1.43V
A0: 5.11V; A1: 1.43V
A0: 5.11V; A1: 1.44V
A0: 5.11V; A1: 1.44V
A0: 5.11V; A1: 1.43V
A0: 5.11V; A1: 1.43V
A0: 5.11V; A1: 1.43V
A0: 5.11V; A1: 1.43V
A0: 5.11V;
```

☑ Avanço automático de linha □ Mostrar marca de tempo · Sem final de linha · 9600 baud

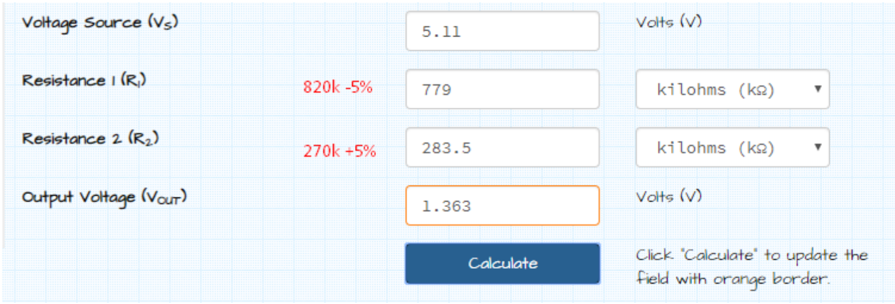
I discovered that there is a "special" effect when reading these voltages with my multimeter: as the multimeter at 20V range doesn't have a much different impedance than the voltage divider (while the voltage divider has 1.09Mohm, the multimeter seems to have 1Mohm, or at least a few Mohms), when I try to read the voltage across the 270kOhm resistor using the multimeter, it influences the circuit so that the serial readings start to be 1.30V at A1 pin, and the multimeter reads 1.09V.

But it seems to me that if this influence of the analog input high impedance was the problem, then the read voltage should be lower than expected, not higher. It's like connecting a big resistor in parallel with the voltage divider resistor. The resistance will decrease, the current increases, and voltage decreases. I can't see why it's reading 1.43V instead of the expected 1.26V.

I also tried including the real measured values of the resistors in the online calculator, but it doesn't seem to make much a difference.



Voltage Source (V_S) 5.11 Volts (V)
Resistance 1 (R_1) 820k +5% = 861 kilohms (kΩ)
Resistance 2 (R_2) 270k -5% = 256.5 kilohms (kΩ)
Output Voltage (V_{OUT}) 1.173 Volts (V)



Voltage Source (V_S) 5.11 Volts (V)
Resistance 1 (R_1) 820k -5% 779 kilohms (kΩ)
Resistance 2 (R_2) 270k +5% 283.5 kilohms (kΩ)
Output Voltage (V_{OUT}) 1.363 Volts (V)

What is wrong? Thank you in advance.

JCA79B
Guest

Re: Arduino not reading the expected voltage on a voltage divider (https://forum.arduino.cc/index.php?topic=616816.msg4179724#msg4179724)
May 20, 2019, 05:31 am (https://forum.arduino.cc/index.php?topic=616816.msg4179724#msg4179724) Last Edit: May 20, 2019, 05:38 am by JCA79B #1

Why such high values for your divider? Try R1 = 22k and R2 = 6.8k. 22k + 6.8k = 28.8k, 6.8k / 22.8k = 0.23611,
5V * 0.23611= 1.18V, 20V * 0.23611 = 4.722V.

jremington (https://forum.arduino.cc/index.php?action=profile;u=196920)

Re: Arduino not reading the expected voltage on a voltage divider (https://forum.arduino.cc/index.php?topic=616816.msg4179738#msg4179738)
May 20, 2019, 06:23 am (https://forum.arduino.cc/index.php?topic=616816.msg4179738#msg4179738) Last Edit: May 20, 2019, 06:31 am by jremington #2

The article you read is giving bad advice. The analog input has very high DC input resistance (>100 Megohm), but it also has a capacitor that must be charged, and the recommended maximum source impedance (the voltage divider) is 10 K.

For your high impedance voltage divider, you must add a 100 nF capacitor from the divider output to ground, in order to reduce the effective impedance of the voltage source. That will slow down the response substantially, but that is fine for infrequent voltage measurements.

Some cheap multimeters have 1 Megohm input impedance. To correct for the meter loading effect in such a case, recalculate the effective voltage divider, which is 820K over 270K in parallel with 1 M (=212.6 K).

So the effective division ratio is 212.6/(820+212.6) = 0.205, instead of 270/(820 + 270) = 0.248, so the meter reading will seem too low.

You also need to calibrate the ADC for the actual ADC reference voltage. Use your multimeter to measure that.

(https://forum.arduino.cc/index.php?action=profile;u=196920)
Brattain Member
Posts: 28,562
Karma: 1945 [add] (https://forum.arduino.cc/index.php?action=karma;sa=applaud;uid=196920;f3c5400d6c=3e6417ea98d8c4e1562fcdc3ecd054da)

Keyboard not found: press F12 to continue.

wvmarle (https://forum.arduino.cc/index.php?action=profile;u=680606)

(https://forum.arduino.cc/index.php?action=profile;u=680606)
Brattain Member
Posts: 15,660
Karma: 787 [add] (https://forum.arduino.cc/index.php?action=karma;sa=applaud;uid=680606;f3c5400d6c=3e6417ea98d8c4e1562fcdc3ecd054da)

Re: Arduino not reading the expected voltage on a voltage divider (https://forum.arduino.cc/index.php?topic=616816.msg4179758#msg4179758)
May 20, 2019, 07:21 am (https://forum.arduino.cc/index.php?topic=616816.msg4179758#msg4179758) #3

I would use a much smaller capacitor, no need for 100 nF. 1-2 nF should be more than enough, as it's about two orders of magnitude larger than the internal one.

Nonetheless it should work without the external capacitor, but you have to read A1 twice. The first reading sets the MUX to that input, and as it takes a moment to charge the internal cap the reading is bound to be off (a bit too high, likely, as the voltage has to drop) Then the second reading will be much more accurate as the ADC has had time to settle.

The better method is of course to lower the resistor values. The recommended input impedance for the ADC is <10k, you have over 200k (calculate the parallel resistance of the two resistors).

For overall more accurate measurements: bring down your input voltage to <1.1V, and use the internal reference. You have to calibrate this reference voltage once (it has a 10% tolerance) but it's stable, usually far more stable than your 5V supply, and independent from the supply voltage. E.g. a 1k2 and a 22k resistor combination will do fine for your 20V input.

Quality of answers is related to the quality of questions. Good questions will get good answers. Useless answers are a sign of a poor question.

Benur21 (https://forum.arduino.cc/index.php?action=profile;u=1230730)

B
(https://forum.arduino.cc/index.php?action=profile;u=1230730)
Newbie
Posts: 6
Karma: 0 [add] (https://forum.arduino.cc/index.php?action=karma;sa=applaud;uid=1230730;f3c5400d6c=3e6417ea98d8c4e1562fcdc3ecd054da)

Re: Arduino not reading the expected voltage on a voltage divider (https://forum.arduino.cc/index.php?topic=616816.msg4180955#msg4180955)
May 20, 2019, 11:48 pm (https://forum.arduino.cc/index.php?topic=616816.msg4180955#msg4180955) Last Edit: May 20, 2019, 11:56 pm by Benur21 #4

Quote from: JCA79B on May 20, 2019, 05:31 am (https://forum.arduino.cc/index.php?msg=4179724)

> *Why such high values for your divider? Try R1 = 22k and R2 = 6.8k. 22k + 6.8k = 28.8k, 6.8k / 22.8k = 0.23611, 5V * 0.23611= 1.18V, 20V * 0.23611 = 4.722V.*

I wanted to interfere as little as possible with the circuit that is being read. So when I read that article, I thought the bigger resistance, the better. It says multimeters usually have 10MOhms impedance, and then gives an example with about 1MOhm impedance. So I thought it was a good number, as it would just draw a few microamps.

That seems to me also a good combination of resistors. But I only bought 820k and 270k resistors, I can buy others, but as of now I still don't know what's going on with the wrong reads, and it would be better if I could use these I already have.

Quote from: jremington on May 20, 2019, 06:23 am (https://forum.arduino.cc/index.php?msg=4179738)

> *The article you read is giving bad advice. The analog input has very high DC input resistance (>100 Megohm), but it also has a capacitor that must be charged...*

When does it must be charged?

Quote from: jremington on May 20, 2019, 06:23 am (https://forum.arduino.cc/index.php?msg=4179738)

> *For your high impedance voltage divider, you must add a 100 nF capacitor from the divider output to ground, in order to reduce the effective impedance of the voltage source. That will slow down the response substantially, but that is fine for infrequent voltage measurements.*

I'd like to be able to read voltage quickly and frequently. Also right now I only have a 10uF capacitor I took of a broken device. Does not fit, correct?

Quote from: jremington on May 20, 2019, 06:23 am (https://forum.arduino.cc/index.php?msg=4179738)

> *Some cheap multimeters have 1 Megohm input impedance. To correct for the meter loading effect in such a case, recalculate the effective voltage divider, which is 820K over 270K in parallel with 1 M (=212.6 K).*
>
> *So the effective division ratio is 212.6/(820+212.6) = 0.205, instead of 270/(820 + 270) = 0.248, so the meter reading will seem too low.*

Yes, it looks like my cheap multimeter really has 1Mega ohm impedance.
Now, any explanation about why the arduino reads 1.43V normally and about 1.30V when I connect the multimeter?

Quote from: jremington on May 20, 2019, 06:23 am (https://forum.arduino.cc/index.php?msg=4179738)

> *You also need to calibrate the ADC for the actual ADC reference voltage. Use your multimeter to measure that.*

If I understand you correctly, I did that. I measured 5.11V between GND and 5V pins with the multimeter and am using it in the calculations. So Vref = 5.11V.

Quote from: wvmarle on May 20, 2019, 07:21 am (https://forum.arduino.cc/index.php?msg=4179758)

> *I would use a much smaller capacitor, no need for 100 nF. 1-2 nF should be more than enough, as it's about two orders of magnitude larger than the internal one.*

What is the advantage of a 1-2 nF instead of a 100nF?

Quote from: wvmarle on May 20, 2019, 07:21 am (https://forum.arduino.cc/index.php?msg=4179758)

> *Nonetheless it should work without the external capacitor, but you have to read A1 twice. The first reading sets the MUX to that input, and as it takes a moment to charge the internal cap the reading is bound to be off (a bit too high, likely, as the voltage has to drop) Then the second reading will be much more accurate as the ADC has had time to settle.*

Do you mean like this:

Code: [Select]

```
void loop() {
  analog_A0 = analogRead(A0) / 200.19;
  analog_A1 = analogRead(A1) / 200.19;
  Serial.println(str1 + analog_A0 + str2 + analog_A1 + str3);
  analog_A0 = analogRead(A0) / 200.19;
  analog_A1 = analogRead(A1) / 200.19;
  Serial.println(str1 + analog_A0 + str2 + analog_A1 + str3);
}
```

or this?

Code: [Select]

```
void loop() {
  analog_A0 = analogRead(A0) / 200.19;
  analog_A1 = analogRead(A1) / 200.19;
  analog_A0 = analogRead(A0) / 200.19;
  analog_A1 = analogRead(A1) / 200.19;
  Serial.println(str1 + analog_A0 + str2 + analog_A1 + str3);
}
```

None of these fix the issue, though. The same result.

Quote from: wvmarle on May 20, 2019, 07:21 am (https://forum.arduino.cc/index.php?msg=4179758)

> *The better method is of course to lower the resistor values. The recommended input impedance for the ADC is <10k, you have over 200k (calculate the parallel resistance of the two resistors).*

So between every analog input I use and the GND I should not have more than 10kOhms?

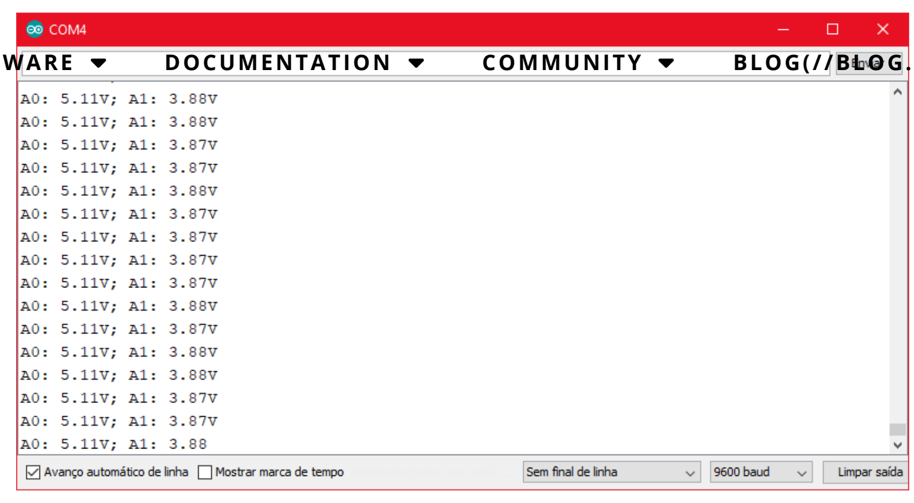
Quote from: wvmarle on May 20, 2019, 07:21 am (https://forum.arduino.cc/index.php?msg=4179758)

> *For overall more accurate measurements: bring down your input voltage to <1.1V, and use the internal reference. You have to calibrate this reference voltage once (it has a 10% tolerance) but it's stable, usually far more stable than your 5V supply, and independent from the supply voltage. E.g. a 1k2 and a 22k resistor combination will do fine for your 20V input.*

Do you mean using analogReference(INTERNAL)? Being max input 1.1V won't it be less accurate?

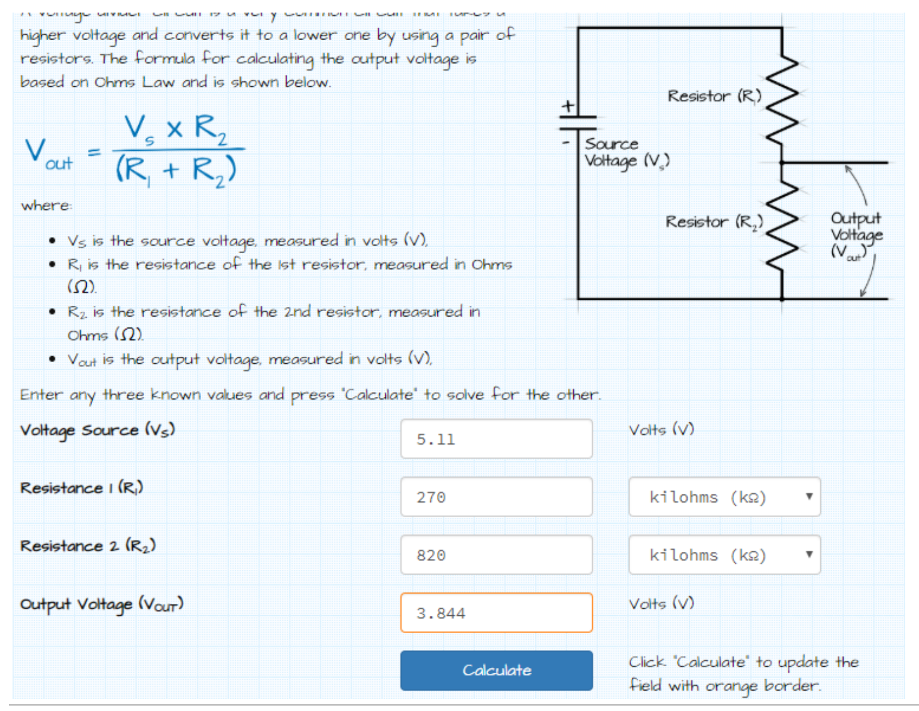
I also still don't understand how a resistor in parallel (like the multimeter or ADC impedance) can increase the read voltage instead of decrease it.

EDIT: Just saying that when I switch the resistor positions, then it reads 'correctly':



```
A0: 5.11V; A1: 3.88V
A0: 5.11V; A1: 3.88V
A0: 5.11V; A1: 3.87V
A0: 5.11V; A1: 3.87V
A0: 5.11V; A1: 3.88V
A0: 5.11V; A1: 3.87V
A0: 5.11V; A1: 3.87V
A0: 5.11V; A1: 3.87V
A0: 5.11V; A1: 3.87V
A0: 5.11V; A1: 3.88V
A0: 5.11V; A1: 3.87V
A0: 5.11V; A1: 3.88V
A0: 5.11V; A1: 3.88V
A0: 5.11V; A1: 3.87V
A0: 5.11V; A1: 3.87V
A0: 5.11V; A1: 3.88
```

☑ Avanço automático de linha □ Mostrar marca de tempo | Sem final de linha | 9600 baud | Limpar saída



A voltage divider circuit is a very common circuit that takes a higher voltage and converts it to a lower one by using a pair of resistors. The formula for calculating the output voltage is based on Ohms Law and is shown below.

$$V_{out} = \frac{V_s \times R_2}{(R_1 + R_2)}$$

where:

- V_S is the source voltage, measured in volts (V),
- R_1 is the resistance of the 1st resistor, measured in Ohms (Ω).
- R_2 is the resistance of the 2nd resistor, measured in Ohms (Ω).
- V_{out} is the output voltage, measured in volts (V),

Enter any three known values and press 'Calculate' to solve for the other.

Voltage Source (V_S)	5.11	Volts (V)
Resistance 1 (R_1)	270	kilohms (kΩ) ▾
Resistance 2 (R_2)	820	kilohms (kΩ) ▾
Output Voltage (V_{OUT})	3.844	Volts (V)

Calculate

Click 'Calculate' to update the field with orange border.

jremington (https://forum.arduino.cc/index.php?action=profile;u=196920)

(https://forum.arduino.cc/index.php?action=profile;u=196920)

Brattain Member
Posts: 28,562
Karma: 1945 [add] (https://forum.arduino.cc/index.php?action=karma;sa=applaud;uid=196920;f3c5400d6c=3e6417ea98d8c4e1562fcdc3ecd054da)

Keyboard not found: press F12 to continue.

Re: Arduino not reading the expected voltage on a voltage divider (https://forum.arduino.cc/index.php?topic=616816.msg4181067#msg4181067)
May 21, 2019, 03:31 am (https://forum.arduino.cc/index.php?topic=616816.msg4181067#msg4181067) Last Edit: May 21, 2019, 03:33 am by jremington #5

Quote

> *I'd like to be able to read voltage quickly and frequently*

How frequently? State the reading rate.

Quote

> *Now, any explanation about why the arduino reads 1.43V normally and about 1.30V when I connect the multimeter?*

Same explanation. The meter resistance changes the voltage divider ratio.

Quote

> *What is the advantage of a 1-2 nF instead of a 100nF?*

The voltage divider will respond more rapidly to input voltage changes, but the voltage readings will show more variation (be noisier).

Quote

> *I also still don't understand how a resistor in parallel (like the multimeter or ADC impedance) can increase the read voltage instead of decrease it.*

It doesn't. You must be doing something else wrong. Study this tutorial on a loaded voltage divider (http://hyperphysics.phy-astr.gsu.edu/hbase/electric/voldiv.html).

wvmarle (https://forum.arduino.cc/index.php?action=profile;u=680606)

(https://forum.arduino.cc/index.php?action=profile;u=680606)

Brattain Member
Posts: 15,660
Karma: 787 [add] (https://forum.arduino.cc/index.php?action=karma;sa=applaud;uid=680606;f3c5400d6c=3e6417ea98d8c4e1562fcdc3ecd054da)

Re: Arduino not reading the expected voltage on a voltage divider (https://forum.arduino.cc/index.php?topic=616816.msg4181133#msg4181133)
May 21, 2019, 06:05 am (https://forum.arduino.cc/index.php?topic=616816.msg4181133#msg4181133) #6

Quote from: Benur21 on May 20, 2019, 11:48 pm (https://forum.arduino.cc/index.php?msg=4180955)

> *When does it must be charged?*

The ADC has a small capacitor built in, that must be charged up to the voltage you want to measure. If you don't give it time to settle to the (new) voltage or to follow a changing voltage, you get a wrong reading.

Quote

> *I'd like to be able to read voltage quickly and frequently. Also right now I only have a 10uF capacitor I took of a broken device. Does not fit, correct?*

That'll slow down your readings badly. Your RC constant is about 2 seconds in that case, so voltage takes about 10 seconds (5*RC) to settle.

Quote

> *What is the advantage of a 1-2 nF instead of a 100nF?*

It settles 50-100 times faster than the 100 nF, but may also have more noise. The faster your signal changes, the least you want to slow this down, so you want a smaller capacitor. The noisier your electrical environment, the more noise you have, and you'll want a larger capacitor for better noise suppression. The final value is as always a compromise.

Quote

> *Do you mean like this:*
>
> Code: [Select]
>
> ```
> void loop() {
>   analog_A0 = analogRead(A0) / 200.19;
>   analog_A1 = analogRead(A1) / 200.19;
>   Serial.println(str1 + analog_A0 + str2 + analog_A1 + str3);
>   analog_A0 = analogRead(A0) / 200.19;
>   analog_A1 = analogRead(A1) / 200.19;
>   Serial.println(str1 + analog_A0 + str2 + analog_A1 + str3);
> }
> ```

No: you're still switching between every reading.

Code: [Select]

```
  analog_A0 = analogRead(A0); // Dummy reading - to set the MUX to A0 and allow input to set
  analog_A0 = analogRead(A0) / 200.19; // Actual reading.
  analog_A1 = analogRead(A1);
  analog_A1 = analogRead(A1) / 200.19;
```

Quote

> *Do you mean using analogReference(INTERNAL)? Being max input 1.1V won't it be less accurate?*

No, more accurate (more stable reference voltage), same resolution (1024 ADC points on the full scale of 0-1023). After all you're working with ratios. Absolute values don't matter for the measurement itself.

When buying resistors don't get the one or two values you think you need, get one of those kits with 20 pcs of 30 values. It's dirt cheap, I got a set of 1% such resistors for less than USD 1 equivalent. You'll end up not needing some values but you never know which one you need when... they come in useful when you least expect it. Cheap enough and small enough to just keep on hand, and a great convenience when you need them, like now. Same for capacitors, a set of multiple values ceramics is the same price. Electrolytics cost a little more, still not much. Very common values I buy in bulk, couple hundred a time, like the 10k, that's used all over the place. Same for the 100 nF capacitor.

Quality of answers is related to the quality of questions. Good questions will get good answers. Useless answers are a sign of a poor question.

Benur21 (https://forum.arduino.cc/index.php?action=profile;u=1230730)

B

(https://forum.arduino.cc/index.php?action=profile;u=1230730)

Newbie
Posts: 6
Karma: 0 [add] (https://forum.arduino.cc/index.php?action=karma;sa=applaud;uid=1230730;f3c5400d6c=3e6417ea98d8c4e1562fcdc3ecd054da)

Re: Arduino not reading the expected voltage on a voltage divider (https://forum.arduino.cc/index.php?topic=616816.msg4182261#msg4182261)
May 21, 2019, 11:34 pm (https://forum.arduino.cc/index.php?topic=616816.msg4182261#msg4182261) #7

Quote from: jremington on May 21, 2019, 03:31 am (https://forum.arduino.cc/index.php?msg=4181067)

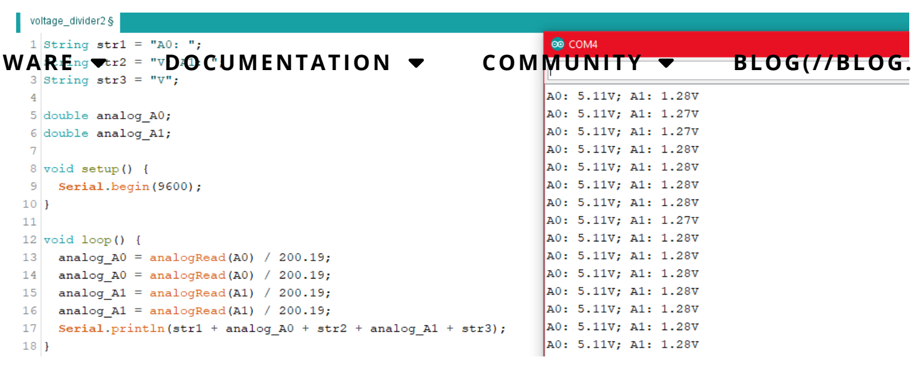
> *How frequently? State the reading rate.*

Everytime I want the analogRead to read the value on a pin. Something like I did in the code I posted. I don't know how fast it runs. I think serial makes it slower, but probably at the actual speed (with the serial) it's already good.

Quote from: wvmarle on May 21, 2019, 06:05 am (https://forum.arduino.cc/index.php?msg=4181133)

> *No: you're still switching between every reading.*
>
> Code: [Select]
>
> ```
>   analog_A0 = analogRead(A0); // Dummy reading - to set the MUX to A0 and allow input to settle.
>   analog_A0 = analogRead(A0) / 200.19; // Actual reading.
>   analog_A1 = analogRead(A1);
>   analog_A1 = analogRead(A1) / 200.19;
> ```

Oh, I wouldn't have thought like that.
IT WORKS! Thank you!!!



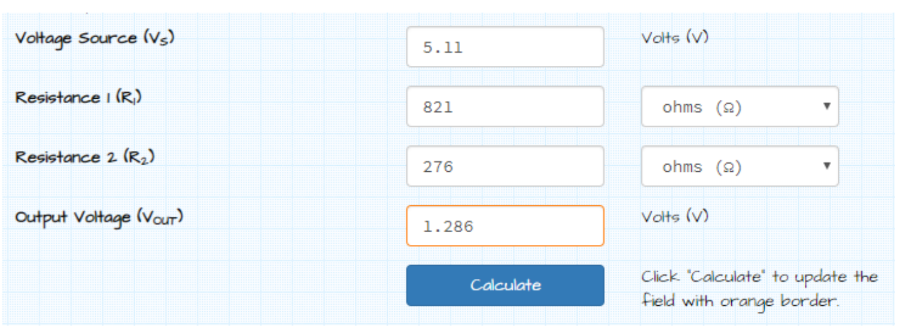
```
String str1 = "A0: ";
String str2 = "V; A1: ";
String str3 = "V";

double analog_A0;
double analog_A1;

void setup() {
  Serial.begin(9600);
}

void loop() {
  analog_A0 = analogRead(A0) / 200.19;
  analog_A0 = analogRead(A0) / 200.19;
  analog_A1 = analogRead(A1) / 200.19;
  analog_A1 = analogRead(A1) / 200.19;
  Serial.println(str1 + analog_A0 + str2 + analog_A1 + str3);
}
```

```
A0: 5.11V; A1: 1.28V
A0: 5.11V; A1: 1.27V
A0: 5.11V; A1: 1.27V
A0: 5.11V; A1: 1.28V
A0: 5.11V; A1: 1.28V
A0: 5.11V; A1: 1.28V
A0: 5.11V; A1: 1.28V
A0: 5.11V; A1: 1.27V
A0: 5.11V; A1: 1.28V
A0: 5.11V; A1: 1.28V
A0: 5.11V; A1: 1.28V
A0: 5.11V; A1: 1.28V
A0: 5.11V; A1: 1.28V
A0: 5.11V; A1: 1.28V
A0: 5.11V; A1: 1.28V
```



Voltage Source (V_S): 5.11 Volts (V)

Resistance 1 (R_1): 821 ohms (Ω)

Resistance 2 (R_2): 276 ohms (Ω)

Output Voltage (V_{OUT}): 1.286 Volts (V)

Now I know the problem was something with the capacitor, so it would not read accurately.
Are there any limitations / disadvantages of doing like this instead of using a 1-2nF capacitor? Other than the normal time to read an analog input being doubled?

> Quote from: wvmarle on May 21, 2019, 06:05 am (https://forum.arduino.cc/index.php?msg=4181133)
>
> *No, more accurate (more stable reference voltage), same resolution (1024 ADC points on the full scale of 0-1023). After all you're working with ratios. Absolute values don't matter for the measurement itself.*

Thanks for the suggestion, then!

wvmarle (https://forum.arduino.cc/index.php?action=profile;u=680606)

(https://forum.arduino.cc/index.php?action=profile;u=680606)

Brattain Member

Posts: 15,660

Karma: 787 [add] (https://forum.arduino.cc/index.php?action=karma;sa=applaud;uid=680606;f3c5400d6c=3e6417ea98d8c4e1562fcdc3ecd054da)

Re: [Solved] Arduino not reading the expected voltage on a voltage divider (https://forum.arduino.cc/index.php?topic=616816.msg4182418#msg4182418)
May 22, 2019, 05:13 am (https://forum.arduino.cc/index.php?topic=616816.msg4182418#msg4182418) #8

A small cap on the pin should help it to settle faster, and you probably don't need the second reading method.

Try it out! See what works best for you.

Quality of answers is related to the quality of questions. Good questions will get good answers. Useless answers are a sign of a poor question.

Go Up Pages: [1]

PRINT (HTTPS://FORUM.ARDUINO.CC/INDEX.PHP?ACTION=PRINTPAGE;TOPIC=616816.0)

Jump to:
=> Sensors Go

ARDUINO® Back to top

Trademark (//www.arduino.cc/en/Trademark/HomePage) Help Center (//support.arduino.cc)
Contact Us (//www.arduino.cc/en/contact-us)
Distributors (//store.arduino.cc/distributors)
Careers (//careers.arduino.cc)

NEWSLETTER
Enter your email to sign up SUBSCRIBE

FOLLOW US
(https://