Postprint of the article published in Physics Education: Gergely Makan, Robert Mingesz and Zoltan Gingl 2021 *Phys. Educ.* **56** 043001, DOI: 10.1088/1361-6552/abf1d3

Reliable readings from Arduino voltmeters

Gergely Makan, Robert Mingesz and Zoltan Gingl

Department of Technical Informatics, University of Szeged, Árpád tér 2, 6720, Szeged, Hungary

E-mail: gingl@inf.u-szeged.hu

Abstract

Modern instruments use sensors, analogue and digital electronics, and digital processing to measure almost any kind of physical quantities. Arduino is a very useful and widely used platform to build such systems in a school environment. Since it has analogue voltage inputs, it can be used as a digital voltmeter or even as a chart recorder. However, the ways of ensuring the required level of accuracy or just the right estimation of accuracy are often unknown or misunderstood. In this paper we try to fill this gap to aid reliable school experimentation.

Introduction

Instruments are essential parts of the study in physics experiments. However, their non-ideal behaviour is often overlooked. It is quite easy to believe the numbers displayed on a digital multimeter, it is rarely kept in mind that it can have much larger error than its resolution. It may be unclear too how the accuracy specifications should be used to present and interpret the results [1]. Although in an educational experiment the accuracy requirements are not demanding in most cases, it is a serious problem in STEM education, if the students have no idea about the basics of related error estimation, or if they just do not care about it. In the following we analyse the Arduino-implemented voltmeters frequently used by teachers and students in various physics experiments.

The Arduino voltmeter

The Arduino UNO's microcontroller integrates an analogue-to-digital converter (ADC). It is a digital voltmeter: it has voltage inputs with a certain range, and outputs an integer number as the reading. As any instrument, it compares the input to a standard, a voltage reference V_{REF}. Using the reading x of the ADC the input voltage can be estimated as:

$$V_{in} = x \cdot \frac{V_{REF}}{1024} = x \cdot V_{LSB} \tag{1}$$

where V_{LSB} is the resolution, the weight of the least significant bit (LSB). Note that most members of the Arduino community use 1023 following the false assumption that 1023 corresponds to V_{REF} [2].

The input voltage is somewhere in a range

$$x \cdot V_{LSB} - \frac{V_{LSB}}{2} \leq V_{in} < x \cdot V_{LSB} + \frac{V_{LSB}}{2} \tag{2}$$

so, there is already a quantisation error of $\pm V_{LSB}/2$. The datasheet of the microcontroller discusses other error sources too, it is enough to mention that the absolute total error limit is specified to be $\pm 2 \cdot V_{LSB}$ [3].This voltmeter has similar error specifications like a digital multimeter. The % of reading is associated with the error of the voltage reference, the additional n digits represent the ADC error expressed in V_{LSB} units [1]. The overall error margins can be approximated as the sum of the errors of the components. For a detailed example calculation see the supplementary document [4].

Arduino voltmeter voltage standard options

The chosen voltage reference limits the achievable accuracy.

Power supply voltage as reference

Unfortunately, the most widely used USB supply voltage as reference is the worst choice. It is specified to be in the range of 4.4 V to 5.25 V. What is even worse, its actual value can change in time and on the board it depends on the unpredictably changing load current too (see figure 1). Therefore, this error can't be removed reliably even if someone measures the voltage with an accurate digital multimeter at a certain time. Powering the board via the power jack (e.g. with a 9V adapter) is a better option, since the on-board regulator provides a stable voltage with better tolerance [4], and it does not depend on the load current noticeably. Oddly enough, some boards do not switch properly to external power in certain cases [5].

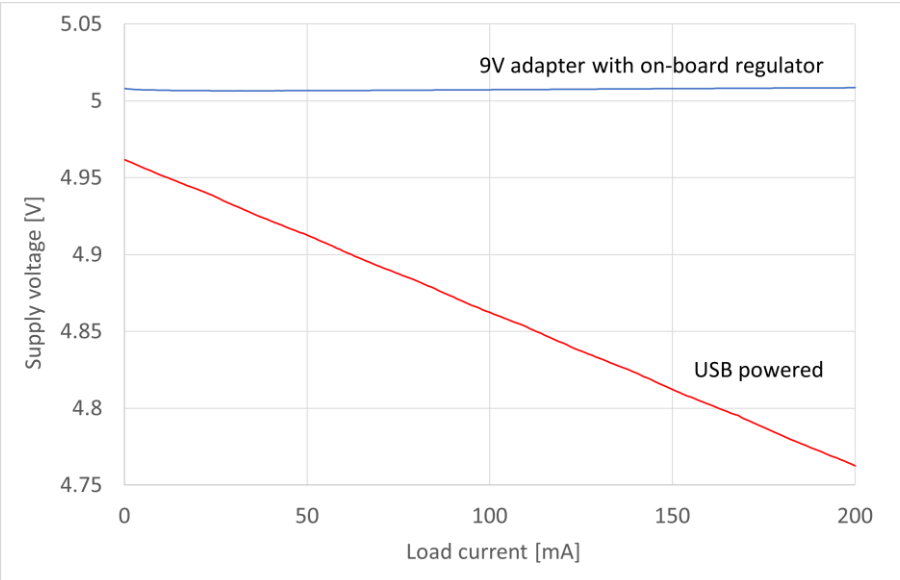

Figure 1 Supply voltage on an Arduino board as a function of the total load current of the port pins in USB and 9 V adapter powered cases.

A seven-segment display can modulate the USB-delivered supply voltage considerably depending on the displayed number. Therefore, time-multiplexed multi-digit displays can generate large noise on the supply and consequently in the readings of the ADC (see figure 2).

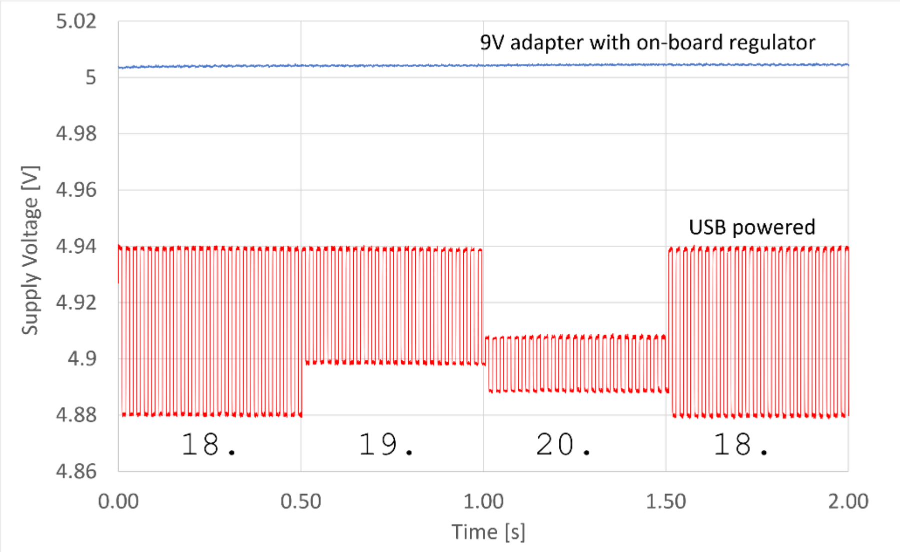

Figure 2 Supply voltage on an Arduino board during displaying three different numbers on a time-multiplexed two-digit 7 segment display for USB and 9 V adapter powered cases. The displayed numbers are shown below the signal.

In a ratiometric arrangement the V_{in} input signal's magnitude is proportional to V_{REF} [3,6], therefore, V_{REF} can be removed from equation (1). However, its value may still matter due to the different dynamics of the reference and voltage inputs of the ADC and the injected noise.

On-chip voltage reference

The microcontroller has an on-chip voltage reference specified to output 1.1 V±0.1 V. The tolerance is moderate, but if the output voltage is known with the required accuracy, it can be a good choice. The main advantage is the better resolution of 1.1 V/1024, about 1 mV. The voltage appears at the AREF pin, but it should not be loaded. The safest is to leave the AREF pin unconnected while the on-chip voltage reference is in use.

External voltage reference

An external reference connected to the AREF pin can provide the highest accuracy and best reliability. There are several voltage reference chips with various voltage levels and tolerances, for an example see [4,7].

Since the Arduino's default configuration connects the supply voltage to the AREF pin inside the microcontroller, wiring it to another output creates a dangerous short circuit, the maximum ratings can be violated [8]. The best practice is to download and run the initial version of the program that selects external reference before connecting anything to the AREF pin.

Student experiments

Teachers can give the students rather instructive related experiments. The various Arduino voltage reference sources can be measured by digital multimeters, and the specifications of the multimeter could be considered too [1]. Different voltages can be generated by resistive voltage dividers and can be measured by the Arduino voltmeter and a multimeter, and the results can be compared. The data can also be used to get more accurate measurements from the Arduino voltmeter by a simple adjustment of the reading. However, one should be careful, this can only be an option, if the voltage reference is stable. Therefore, the USB-powered option should be avoided, at least an external supply source should be used in any Arduino voltmeter application. We believe that it should be taught that in most cases it is the safest to use a sensor and a voltage reference with the desired accuracy, since the calibration needs expertise beyond the school level [7] while accuracy estimation is easier and should be an important part of experimenting STEM education. It can also be instructive to choose a reference and sensor that fits to a given accuracy requirement. This is what experts do in practice.

Acknowledgments

This study was funded by the Content Pedagogy Research Program of the Hungarian Academy of Sciences.

References

1. Gingl Z and Mingesz R 2019 Comment on ‘Resistance of a digital voltmeter: teaching creative thinking through an inquiry-based lab’ *Phys. Educ.* **54** 056502
2. Gingl Z, Makan G, Mellár JZ 2018 Arduino data conversion – 1023 or 1024? *figshare* https://doi.org/10.6084/m9.figshare.7434074
3. Makan G, Mingesz R and Gingl Z 2019 How accurate is an Arduino Ohmmeter? *Phys. Educ.* **54** 033001
4. Makan G, Mingesz R and Gingl Z 2021 Error estimation of the Arduino voltmeter *figshare* https://doi.org/10.6084/m9.figshare.14166728
5. Makan G, Mingesz R, Gingl Z 2020 Plugging in the external supply may leave your Arduino USB-powered *figshare* https://doi.org/10.6084/m9.figshare.13011959
6. Kinchin J 2018 Arduino and cheap thermistor to make a simple temperature sensor *Phys. Educ.* **53** 063008
7. Gingl Z and Mingesz R 2021 Comment on ‘A low-cost cryogenic temperature measurement system using Arduino microcontroller’ *Phys. Educ.* **54** 056502
8. Gingl Z, Mingesz R, Makan G and Mellar J 2019 Driving with an Arduino? Keep the lane! *Phys. Educ.* **54** 025010