This is the version of the article before peer review or editing, as submitted by an author to Physics Education. IOP Publishing Ltd is not responsible for any errors or omissions in this version of the manuscript or any version derived from it. The Version of Record is available online at https://doi.org/10.1088/1361-6552/aafa41

Driving with Arduino? Keep the lane!

Zoltan Gingl, Robert Mingesz, Gergely Makan and Janos Mellar

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

Abstract

Arduino is a popular and very useful tool in STEM education. Teacher demonstrations, laboratory and home work of students are supported by an incredibly wide range of application examples at very low cost. Since the heart of the Arduino board is an industrial microcontroller, it is a good chance to teach the basics of the related rules, standards and engineering-like approach of application that is essential in high quality STEM education. We discuss here the output drive capability often misinterpreted in the Arduino community and we also draw the attention to the importance of developing the right attitude.

Output drive characteristics of the Arduino board

The Arduino board is practically a microcontroller whose pins are wired to connectors – so called pin headers – to ease access. Some supporting circuitry is used on the board to allow communication and to provide power. Many pins can be programmed as digital inputs or outputs, and in the latter case they can be used to drive external components including LEDs (see figure1) and other kinds of additional circuits.

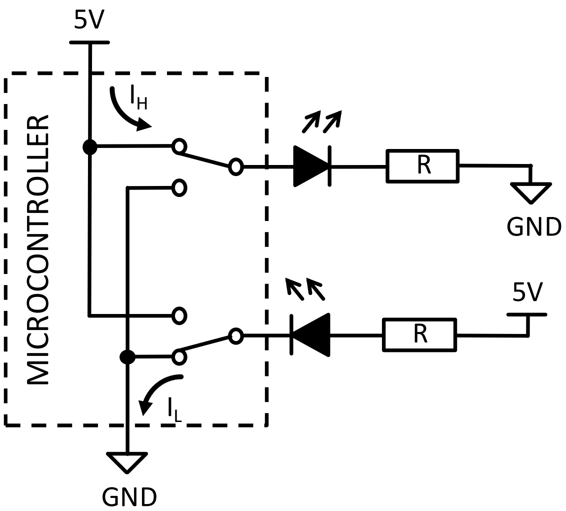


Figure 1. The microcontroller's outputs can be programmed to logic high or logic low state, when they can source I_H or sink I_L current, respectively. Accordingly, LEDs can be operated using positive or negative logic.

Integrated circuits always have specifications including current drive capability and its limitations. If we look in the datasheet of the microcontroller of the Arduino Uno, we can find two sections of specifications: "Absolute Maximum Ratings" and "Common DC Characteristics" [1]. The first one is always explained in the datasheets, since it is extremely important. Let's see how can we understand the three sentences of this.

Possible partial or full damage

"*Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device.* "

This means that the limits must not be exceed in any case for any time even under the worst conditions. What happens otherwise? The device will not necessarily be damaged partially or completely, but the user will not know any more if the device will still be fully functional. Although one may think, that it is possible to test if some degradation has been occurred, it is not the case, since the internal structure of the chip is very complex (and unknown for the user). There is no way to check all the functionality in the case of partial degradation that can even lower the specified tolerance, can make the device more sensitive. Therefore, the reliability – essential in any electronic equipment and system – is lost.

Normal operation is not guaranteed

"*This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied.*"

So, even if we are within these ratings, it does not guarantee normal operation if the recommended conditions are not met. We can't be sure that our device really works as a microcontroller in all aspects. It may look working fine, but it does not mean that full functionality is still available. Unexpected behaviour can occur in slightly changed situations without knowing the reasons.

Operation close to the limits is a bad practice

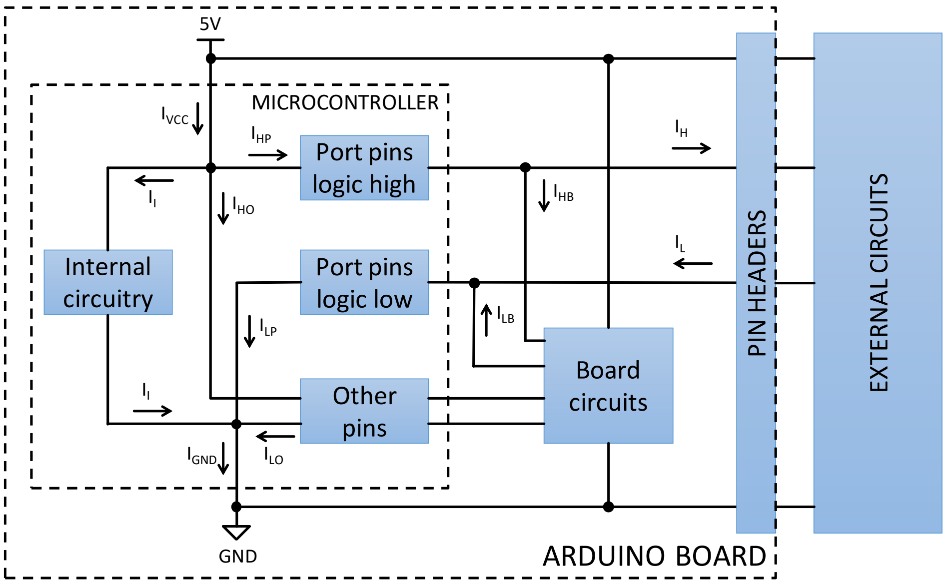
"*Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*"

It speaks for itself, makes clear that it is not allowed to design the operation of the device at the limits. In practice, close to the limits, since it is impossible to stay exactly at these. Just think about the variations in the 5 V supply voltage and loading.

Do people care these?

Most Arduino descriptions – even the official tutorial [2] – say that the device can drive 40 mA per pin even though it is specified as an absolute maximum. According to the above it is certainly misleading and can encourage users to accept it for normal operation. It would be much better to say that pins can tolerate 40 mA, but one must design using the maximum value specified in the operational section, 20 mA (at V_{CC}=5 V supply voltage) [1]. Fortunately, the official Arduino Uno board page is correct in this sense [3].

It is almost never mentioned that the currents entering/leaving the microcontroller at the supply pins (I_{VCC} and I_{GND}) are limited to an absolute maximum of 200 mA [1]. Figure 2 shows how the supply pin currents are related to the currents flowing between the board and external circuits.



$$I_{VCC}=I_{HP}+I_{HO}+I_I$$
$$I_{HP}=I_H+I_{HB}$$
$$\mathbf{I_{VCC}=I_H+I_{HB}+I_{HO}+I_I}$$

$$I_{GND}=I_{LP}+I_{LO}+I_I$$
$$I_{LP}=I_L+I_{LB}$$
$$\mathbf{I_{GND}=I_L+I_{LB}+I_{LO}+I_I}$$

Figure 2. Main currents flowing in the system. The sum of the sink and source currents of the external circuitry via the port pins are denoted with I_H and I_L, respectively. The board circuits' and the microcontroller's internal circuits contribute to the I_{VCC} and I_{GND} supply pin currents.

It is easy to see, that the 200 mA limitation contains the currents of the board and microcontroller also. These typically fall below 20 mA, leaving an absolute maximum of about 180 mA for the external circuits.

There are further restrictions as well: the sum of the currents of specific groups of pins are also limited as shown in table 1.

Sum of all I_H source currents of the following groups of pins is limited to 150 mA, V_{CC}=5 V			
microcontroller pins	boards pins	estimated maximum internal currents	remaining limit for external circuitry
C0 - C5, D0 - D4 ADC7, RESET	A0 - A5, 0 - 4 RESET	5 mA	**145 mA**
B0 - B5, D5 - D7 ADC6, XTAL1, XTAL2	5 - 13	1 mA	**149 mA**
Sum of all I_L sink currents of the following groups of pins is limited to 100 mA, V_{CC}=5 V			
microcontroller pins	boards pins	estimated maximum internal currents	remaining limit for external circuitry
C0 - C5, ADC7, ADC6	A0 - A5	1 mA	**99 mA**
B0 - B5, D5 - D7 XTAL1, XTAL2	5 - 13	1 mA	**99 mA**
D0 - D4, RESET	0 - 4, RESET	5 mA	**95 mA**

Table 1. Limitations for source/sink currents for groups of pins of the ATmega328P microcontroller [1] of the Arduino Uno board.

Examples – The Good, the Bad and the Ugly

A "LED Bar Graph" example [4] connects 10 LEDs via 220 Ohm resistors to the pins, so the current per pin is about 15 mA for red LEDs. When all LEDs are switched on, the total I_{VCC} current contribution is about 150 mA. It is sufficient, nevertheless a bit close to the limit, leaves only a small room for additions/modifications. The sums of currents flow out via pins 2 - 4 (45 mA) and 5 – 11 (105 mA) are within the specifications also. However, switching to negative logic (see figure 1), the 105 mA would be too high as a sum of sink currents! It would be more safe to reduce the current.

The "Row-column Scanning to control an 8x8 LED Matrix" [5] uses no series resistors at all. Consequently the 40 mA per pin absolute maximum is certainly violated, and if the user switches on more LEDs, other limits may also be exceeded.

Is it costly to damage a board?

Sometimes users say that Arduino is cheap, it is not a big trouble if it gets damaged. In addition, if only an output is damaged, the others can still be used. We think this is a very dangerous approach in teaching and learning. Education and practicing are done within an isolated, protected environment, where one can make mistakes without too serious consequences. However, the role of education is to prepare the students for the real life and conditions, to be careful, well-informed, trustworthy. Just think about the many car recalls due to design mistakes, and that pilots must be serious about safety even during practicing with a flight simulator. Consistent feedback is very important in education as well, violation of the rules must be painful in some sense, otherwise students won't be stimulated to do the things right. High quality STEM education is essential for the engineers, scientists, professionals and teachers of the future, so the cost of damaging an Arduino or part of it can be high!

The message

Teacher colleagues, students, developers! Be conscious, careful, respect the rules determined by leading experts, do not let a bad habit to be developed. Figure 3 warns you to always keep enough margin and not to drive blindfolded as Chaplin did in the "Modern times" roller skating scene [6]. Read also an application engineer's very clear notes on the subject [7] and some more about the physics background [8]. Do not forget, that absolute limits are important in other parts of our life as well – elevators, bridges, traffic, economy, social life, …

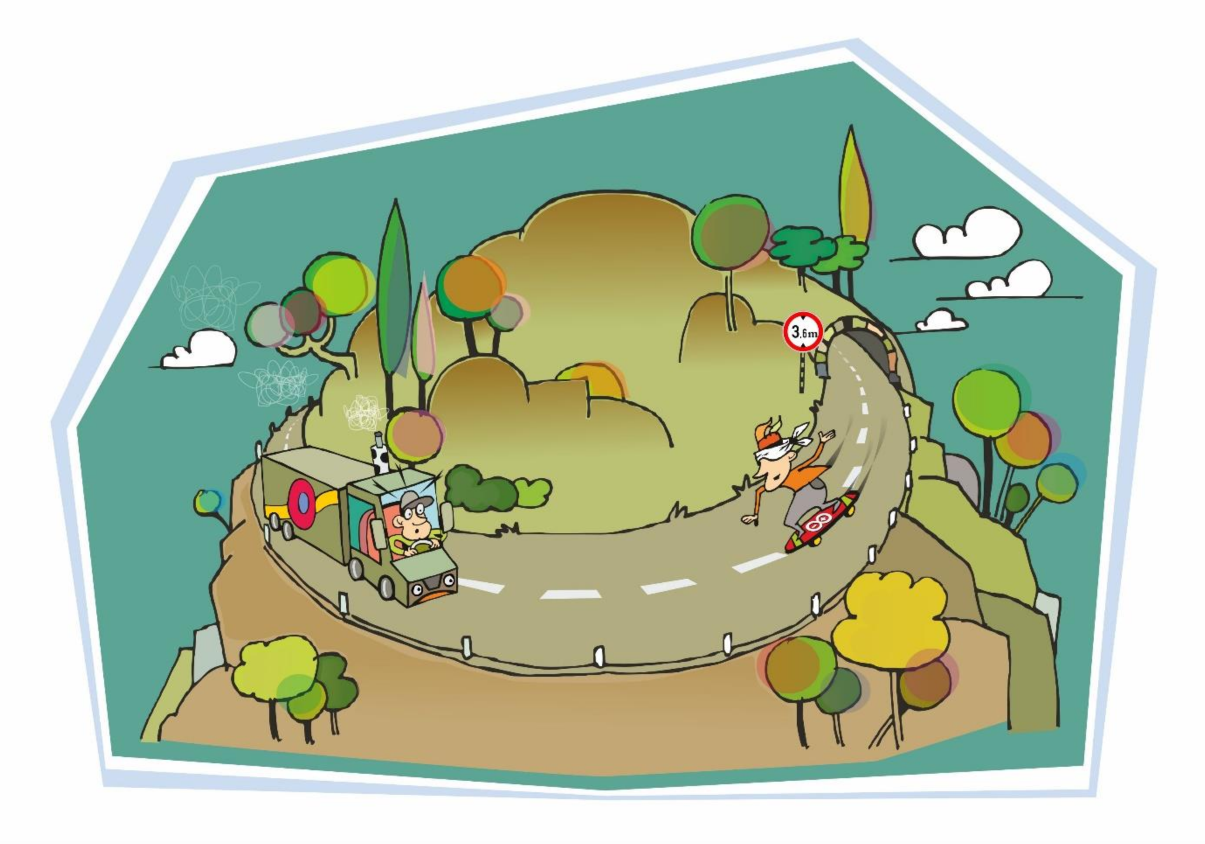


Figure 3. Always keep enough margin and do not drive blindfolded!

Acknowledgments

This study was funded by the Content Pedagogy Research Program of the Hungarian Academy of Sciences. The authors thank Csaba Magyar for his imaginative illustration.

References

[1] ATmega328P, https://www.microchip.com/wwwproducts/en/ATmega328P

[2] Digital pins, In: Arduino tutorials, https://www.arduino.cc/en/Tutorial/DigitalPins

[3] Arduino Uno Rev3, In: Arduino products, https://store.arduino.cc/arduino-uno-rev3

[4] LED Bar Graph, In: Arduino tutorials/Built-In Examples, https://www.arduino.cc/en/Tutorial/BarGraph

[5] Row-columm Scanning to control an 8x8 LED Matrix, In: Arduino tutorials/Built-In Examples, https://www.arduino.cc/en/Tutorial/RowColumnScanning

[6] Chaplin: Modern times, roller skating scene, https://www.youtube.com/watch?v=kPcEFHA3X0c

[7] John Ardizzoni, What's the big deal about ABSOLUTE MAXIMUM RATINGS?, In: Analog Devices, Rarely asked questions, https://www.analog.com/en/analog-dialogue/raqs/raq-issue-50.html

[8] James Bryant , Who Killed the Component?, In: Analog Devices, Rarely asked questions, https://www.analog.com/en/analog-dialogue/raqs/raq-issue-130.html