A SURVEY ABOUT POWER CONSUMPTION FOR ARDUINO

Sabou **SEBASTIAN**
Techical University of Cluj Napoca
sebastian.sabou@cunbm.utcluj.ro

Keywords: Survey, power, Arduino, low-power

Abstract*: The choice of a hardware platform for the implementation of a project is very important also from the perspective of the electricity consumption that it implies. For example, systems that will operate on the basis of energy stored in batteries, or those powered by renewable sources, whose availability is difficult to predict and also monitoring and control systems that are located in hard to reach areas or that are rarely physically checked require increased attention in the selection of the platform to be used. This study aims to provide a comparison, in terms of energy consumption, of the most used Arduino systems, as well as some possible solutions to further reduce their consumption.*

1. INTRODUCTION

Lately, there is a growing need to design and build systems that can be powered by batteries or that can use renewable electricity. This fact implies the use of systems that have a low electricity consumption but on the other hand to be available in large quantities and at the lowest possible price. Some of the most common microcontroller systems are those in the Arduino family.

The purpose of this study is to make a comparative analysis of electricity consumption among the most common models of Arduino development systems: Uno, Nano, Pro Mini and Mega. Most of them use the same microcontroller but are in different formats and some different components. Also, there are presented some hardware and software solutions to reduce the power consumed by these systems.

Being a study, measurements and results obtained by the author are presented, as well as data taken from bibliographic materials.

1.1. Arduino

The Arduino family of development systems contains a wide variety of platforms, using many types of microcontrollers or microprocessors. In this case we will focus on systems that use 8-bit microcontrollers, being among the most used, from CNC or 3D printers to automation systems, to develop intelligent sensors, systems for measuring and monitoring various parameters, having available for use a very large number of sensors, there are an impressive number of implemented projects whose documentation is available free of charge on the Internet.

Table 1. Characteristic of the studied Arduino systems [1-5]

Board	**Arduino Mega**	**Arduino Uno**	**Arduino Nano**	**Arduino Pro Mini 5V**	**Arduino Pro Mini 3.3V**
Processor	ATmega2560	ATmega328P	ATmega328	ATmega328P	Atmega328P
Clock speed	16MHz	16MHz	16Mhz	16MHz	8MHz
SRAM	8kB	2kB	2kB	2kB	2kB
Flash Memory	256kB	32kB	32kB	32kB	32kB
EEPROM	4kB	1kB	1kB	1kB	1kB
Digital I/O pins	54	14	22	14	14
Analog I/O pins	16	6	8	6	6
Operating voltage	5V	5V	5V	5V	3.3V
Input voltage	7-12V	7-12V	7-12V	5-12V	3.35-12V

2. RESEARCH AND MEASUREMENTS

Table 2 shows the measurements performed on several Arduino systems, compared to those taken from [6]. No measurements were performed with USB powered systems because in this case it depends a lot on the USB interface circuit and on the other hand the stand-alone systems are not permanently powered via USB.

Table 2. Power consumption ([6] and personal measure)

Board/VCC	**Arduino Mega**	**Arduino Uno**	**Arduino Nano**	**Arduino Pro Mini 5V**	**Arduino Pro Mini 3.3V**
Reference					
9V	73.2mA [6] 71.2mA	98.4mA[6] 94.2mA	22.1mA[6] 22.0mA	14.6mA[6] 14.3mA	5.1mA[6] 5.2mA
Reduce clock					
9V	61.8mA[6] 62.0mA	42.8mA[6] 42.3mA	18.5mA[6] 18.6mA	10.0mA[6] 10.1mA	3.8mA[6] 3.9mA
Enable Low Power					
9V	26.9mA[6] 27.1mA	27.9mA[6] 28.1mA	4.8mA[6] 4.9mA	3.2mA[6] 3.2mA	3.2mA[6] 3.2mA

The measurements were performed after programming an empty program, so that the systems do not use any peripheral device.

The differences between the measurements can also be explained by the fact that there may have been different versions of Arduino systems or even by different manufacturers. Also, the stabilizer circuit and the USB interface circuit may be different.

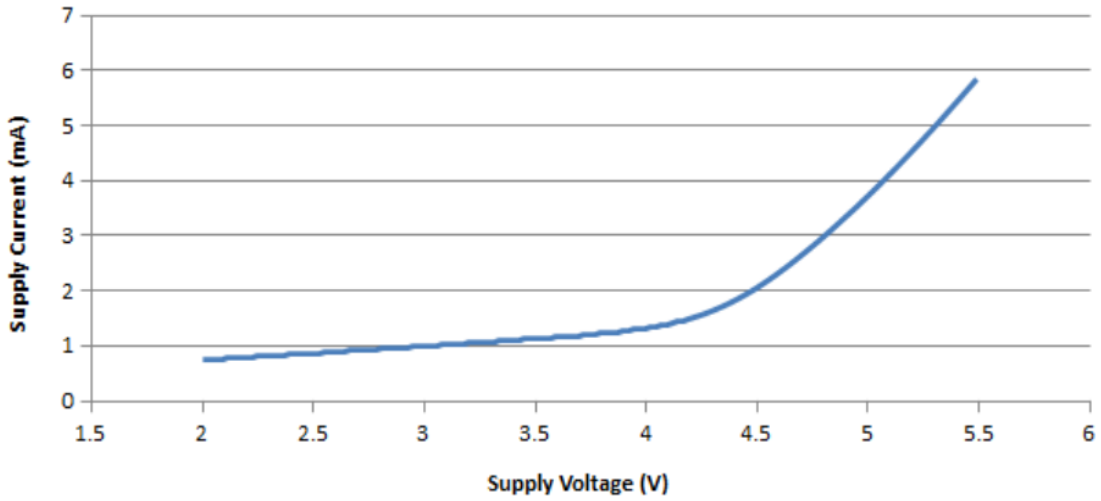

Fig 1. Atmega328P Vcc vs Supply Current (at 1MHz clock)[datasheet]

A series of useful information can also be obtained from the datasheets, thus, in *figure 1* you can see the change in consumption compared to the supply voltage, at a certain clock frequency, and in *figure 2* the variation of the electric current depending on the clock frequency at a constant supply voltage of 5V

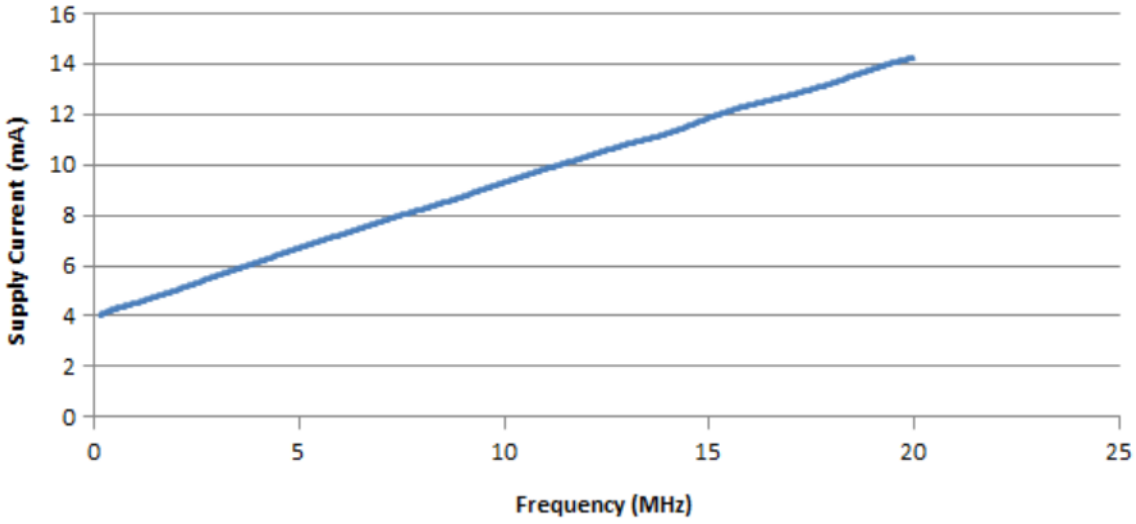

Fig. 2 Atmega328P Frequency vs Supply Current (5V) [datasheet]

Reducing the frequency of the clock signal is another method of reducing power consumption, software can be implemented, reducing the frequency from 16MHz to 8MHz can reduce the current from 12mA to 8.5mA, over an extended period of time can mean greater autonomy. Reducing the clock frequency means even fewer calculations performed over a period of time, often not the best method, being preferred methods to put the system into deep sleep, for example, being more efficient that way.

On the other hand changing clock speed can cause boot loader issues and your Arduino may get damaged.

In certain situations, hardware changes can be made that allow a significant reduction in consumption, as in [7] where some changes that require certain knowledge of electronics, such as modifying the circuit to power the USB interface circuit from this USB interface. , resulting in a reduction of 44.7%, the reduction by 7mA of the current by disconnecting the power led and a major change is the use of a switching voltage stabilizer instead of the linear voltage stabilizer. All these involve quite important changes on the system, changes that are not the subject of this study but are mentioned as a possible way to reduce consumption.

There is also another way to reduce electricity consumption, namely by using the library of low power functions [8]. The advantage of this method is that it does not require hardware changes, it requires only a more detailed knowledge of the programming process and microcontroller architecture, the reduction of consumption can be significant (according to Table 2) depending very much on the requirements to implement software.

3. CONCLUSIONS

There are several solutions to use an Arduino system that is best suited to the requirements of low power consumption. The first and easiest solution is to choose a system that is as close to the design requirements as possible, so if you need a circuit that requires only a few digital inputs or outputs and a few analog inputs, you can opt for an Arduino Nano (22mA) or Arduino Pro Mini (15mA) compared to an Arduino Uno (98mA). A second solution, which requires only the use in the programming of function libraries that allow the use of low power facilities, involves the development of programs that are designed to make the most of these functions. A third option, with more effort, is one that involves hardware changes, eliminating any consumers that are not needed but are present on these systems.

REFERENCES

[1] https://store.arduino.cc/arduino-mega-2560-rev3.

[2] https://store.arduino.cc/arduino-uno-rev3.

[3] https://store.arduino.cc/arduino-nano.
[4] https://store.arduino.cc/arduino-pro-mini.
[5] https://learn.sparkfun.com/tutorials/reducing-arduino-power-consumption/all.
[6] https://diyi0t.com/arduino-reduce-power-consumption/.
[7] https://www.defproc.co.uk/tutorial/how-to-reduce-arduino-uno-power-usage-by-95/.
[8] https://www.arduino.cc/en/Reference/ArduinoLowPower.
[9] E. White, *Making Embedded Systems*, O'Reilly Media; Inc. USA, 2011.
[10] J. Henkel, S. Parameswaran, *Designing Embedded Processors. A Low Power Perspective*, Springer Netherlands, 2007.