DOI: 10.24427/978-83-66391-69-7_1

Chapter 1
Selection of ESP32 operating parameters in order to reach optimal energy efficiency during calculations

Agnieszka Choroszucho, Jakub Bednarek, Mateusz Sumorek and Jakub Żukowski
Bialystok University of Technology, Faculty of Electrical Engineering

In the era of technological processes automation development, the popularization of house automation, constant limitation of human participation in these processes, as well as the intensification in the field of broadly understood IoT, such devices are increasingly used for remote monitoring and control of various processes. In recent years, the ESP32 platform has gained a significant increase in the popularity of the IoT market. It is often used for logging and remote controlled devices, mostly in smart home solutions. Such devices often have to be small and have long battery work-time. The test results presented in the article will allow for a proper selection of working parameters in order to achieve the most optimal energy efficiency (smallest losses) while performing the assumed task.

Index terms: embedded system, ESP32, Internet of Things, energy efficiency, smart homes, automation.

Introduction

The development of the automation of technological processes and the need for their remote monitoring, a growing popularity of smart home solutions and the collection of data derived from them is associated with the emergence of various needs. Growing interest in the broadly understood IoT results in the emergence of increasingly newer solutions from manufacturers striving to create the most comprehensive system. In recent years, the ESP32 family from Espressif Systems has gained particular popularity. It is a microcontroller based on an efficient 32-bit Xtensa LX6 microprocessor [1] equipped with a Wi-Fi and Bluetooth version 4.2. Thanks to these built-in functionalities, low price, refined documentation and large support of the constructor community, this system is very popular in such solutions as smart homes or remote data collection (e.g. logging of temperature and pressure measurement sensors).

Devices that are created for such purposes must meet certain requirements. The most common ones are small dimensions and the longest possible battery life. Despite the very dynamic development and well-developed battery solutions, it is still difficult to reconcile these two requirements. Thanks to the development of microprocessor systems, they are able to be very energy efficient. Currently, most solutions have a built-in sleep mode. The ESP32 microcontroller in sleep mode is able to limit power consumption to tens or even single μA [1]. This allows to significantly extend the on-battery lifetime of the device. Unfortunately, in this state the device is not able to perform any demanding tasks – only after waking up it can take the assumed actions, which is associated with increased power consumption.

This article presents the results of conducted research. They show with which operating parameters the lowest energy consumption can be obtained during the execution of an algorithm. This is to extend the device's operating time while using battery-based power.

Research assumptions

A proper selection of the device's operating parameters allows to extend its operating time without having to replace the battery or recharge it. Due to this procedure, maintenance work can be carried out less often. This is particularly important when the device is mounted in a hard-to-reach place. The limitation of servicing results in lower operating costs, which is primarily noticeable in large, distributed installations or systems.

In order to select the optimal operating parameters, the measurements of the algorithm execution time and the current consumption of the device were carried out during this process. These measurements concerned different values of the microcontroller clock speed and voltage of different values. As a calculation algorithm, the calculation of the π number was used in accordance with formula (1.1) [7]:

$$\pi = 4 \cdot \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{2n-1} = 4 \cdot \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \ldots\right). \tag{1.1}$$

Figure 1.1 presents the algorithm written in MicroPython language [2], characterized by low computational efficiency [3], especially where the algorithm uses iterative methods [5, 6].

In connection with the fact that formula (1.1) is a slow convergent method [4] (requiring a large amount of iteration to achieve a satisfactory result), a significant load on the processor was forced, resulting in a significant increase in current consumption.

```
count = 0
machine.freq(240000000) #taktowanie zegara

while(count < 1):

  print('start')
  start = time.ticks_ms()
  a = 0
  for i in range (125000):
    a += ((-1)**i)/(2*i+1)
  end = time.ticks_ms()
  print(end - start) #zwróć czas obliczeń
  print(4*a)
  count = 1
```

FIGURE 1.1. Algorithm used to calculate the number of π.

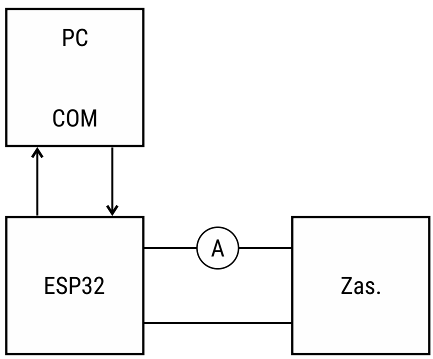

FIGURE 1.2. Measurement diagram, PC, ESP32 – tested microcontroller, Zas – laboratory power supply.

Two factors were selected as set parameters of the ESP32 microcontroller: supply voltage and clock speed of the processor. The value of the supply voltage, according to the manufacturer's documentation, should be in the range <2.7; 3.6> V (current recommendation of the manufacturer is 3.0 V – 3.6 V) [1]. In tests, measurements in the range of <2.7; 3.3> V were conducted. For each given supply voltage, measurements for different clock speeds in the range of 20 – 240 MHz (from available settings options) were performed.

The measurement of the time needed to complete the task and also the time of increased power consumption was made at the level of the algorithm that returned the value to the computer through the serial port (Fig. 1.2). As preliminary current measurements using the oscilloscopic method showed that the current consumed during the calculations is constant during their duration (no fluctuations were noted), it was accepted with satisfactory accuracy that the measurement of the current value using an ammeter would be appropriate (Fig. 1.2). In the test, the supply voltage of the system using a stabilized power supply was set, which in combination with the current measurement, enabled the determination of energy consumption in mWh during calculations in accordance with formula (1.2):

$$E = U \cdot I \cdot \frac{t}{3600}, \tag{1.2}$$

where:
E – used energy in mWh,
U – supply voltage,
I – current consumption under load,
t – time of algorithm execution.

Results

Figure 1.3 shows the dependence of the calculation time on the processor clock frequency. The curves in Fig. 4 show a family of characteristics of the dependence of consumed energy on the processor clock frequency, taking into account the system supply voltage.

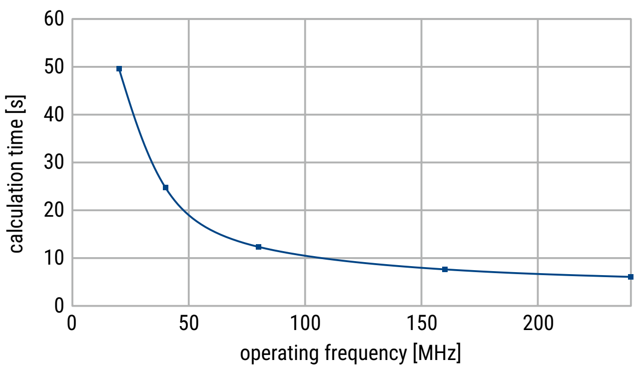

FIGURE 1.3. Characteristics of the dependence of calculation time on clock frequency.

The drawn family of characteristics (Fig. 1.4) allows to observe the linearity of the device's energy consumption along with the reduction of the processor's supply voltage.

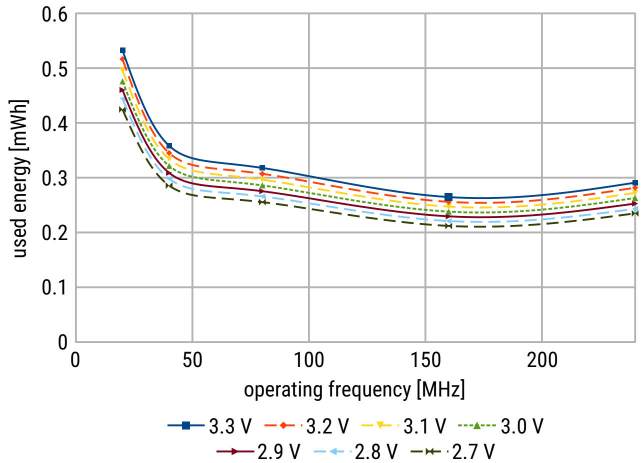

FIGURE 1.4. **Family of characteristics showing energy consumption depending on processor clock speed including supply voltage.**

Figure 1.5 shows the dependence of the used energy on supply voltage on the processor clock frequency.

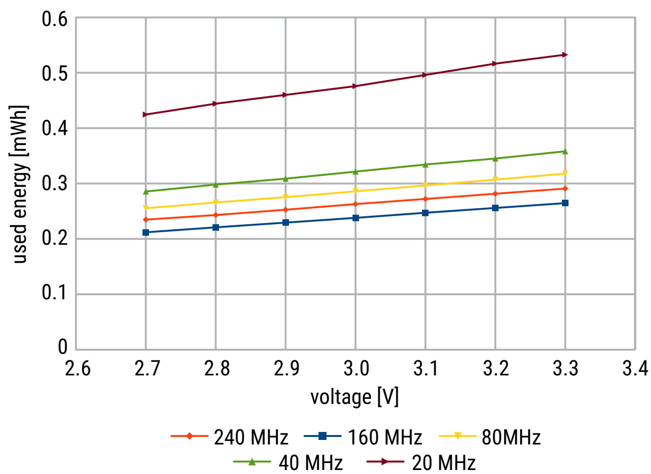

FIGURE 1.5. **Family of characteristics showing energy consumption as a function of supply voltage.**

Based on the conducted measurements, it can be concluded that supply voltage does not affect the duration of the performed calculations (Table 1.1).

TABLE 1.1. Measurement results

Power supply [V]	Frequency [MHz]	Calculation time [ms]	Used energy [mWh]
3.3	20	49664	0.5326
	40	24729	0.3582
	80	12339	0.3178
	160	7635	0.2646
	240	6068	0.2909
3.2	20	49664	0.5165
	40	24729	0.3451
	80	12339	0.3071
	160	7635	0.2559
	240	6068	0.2816
3.1	20	49664	0.4961
	40	24729	0.3343
	80	12339	0.2964
	160	7635	0.2472
	240	6068	0.2722
3	20	49664	0.4759
	40	24729	0.3215
	80	12339	0.2859
	160	7635	0.238
	240	6068	0.2629
2.9	20	49664	0.4601
	40	24729	0.3088
	80	12339	0.2753
	160	7635	0.2294
	240	6068	0.2527
2.8	20	49664	0.4442
	40	24729	0,2981
	80	12339	0,2658
	160	7635	0.2209
	240	6068	0.2431
2.7	20	49664	0.4246
	40	24729	0.2856
	80	12339	0.2554
	160	7635	0.2119
	240	6068	0.2348

Discussion

Analyzing the processor's documentation, it can be observed that in its structure at the power input there are linear stabilizers responsible for supplying individual sections of the microcontroller, such a decrease is directly related to them (Fig. 1.6). The losses are proportional to the voltage drop across the voltage regulator. This means that bringing the supply voltage closer to the minimum value that allows an element to work correctly will result in the smallest energy loss on it.

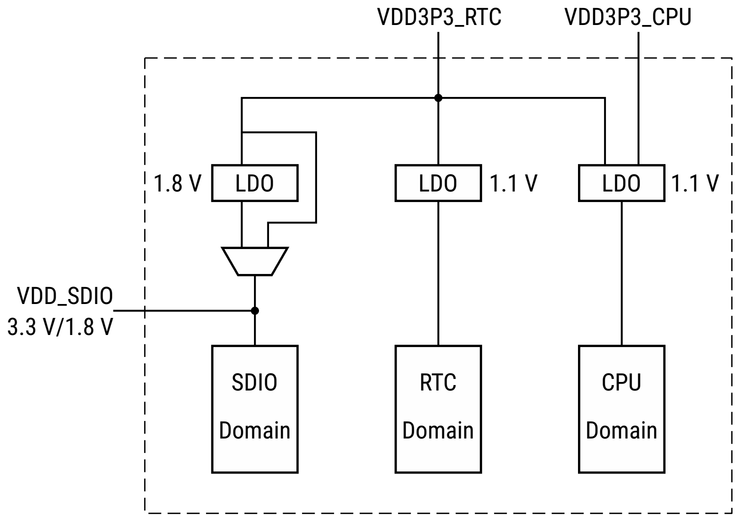

FIGURE 1.6. **ESP32 processor power input structure from the manufacturer's documentation [1]**

In Figure 1.4 the presented characteristics clearly show the optimal operating point of the microcontroller, which determines the supply voltage of 2.7 V and the processor clock speed of 160 MHz. This assumption works when the processor uses only basic peripherals for mathematical calculations. With this assumption, energy consumption for carrying out the task equaled 0.211 mWh. The system working with standard parameters, which are 3.3 V and 240 MHz, used 0.263 mWh. A decrease in energy consumption is classified at around 20% relative to the standard operating parameters of the controller. Taking into account the updated recommendations of the ESP32 manufacturer (supply voltage range), it should be in the range of 3.0 V – 3.6 V. The optimal operating point is the following: supply voltage of 3.0 V and clock speed equal to 160 MHz. ESP32 at these parameters consumed 0.238 mWh, which means a decrease in energy consumption by about 10% compared to standard parameters. The characteristics show a significant increase in energy consumption by the microcontroller when operating at low clock frequencies (Fig. 1.3). Despite a significant decrease in the controller's current consumption at low clock speeds, the calculation time lengthened significantly, resulting in an increase in energy consumption.

Additional characteristics of the dependence of the system's energy consumption are presented in Fig. 4. It was also observed that the exponential increase in calculation time along with a decreased calculation time on the clock operating frequency were drawn up in the article to determine the main impact on the in-clock speed is visible. Based on both characteristics, a significant correlation can be made between them in the low clock frequency range.

Conclusions

The article presents the results of testing energy consumption of the ESP32 microcontroller manufactured by Espressif Systems. The purpose of the measurements was to determine the optimal operating parameters of the system in order to most efficiently use energy to perform a given task, and evaluate the extension of the potential operating time of the device in battery supply conditions. According to the measurements made, the reduction of the clock frequency reduces the value of the current consumed by the microcontroller, but significantly extends the calculation time. The value of the current consumed and the calculation time allow for determining the system's operating parameters at which it is possible to reduce energy consumption by up to 20% as compared to standard operating parameters. The research will be continued using other research methods that take into account different operating conditions of the device.

This work was supported by Ministry of Science and Higher Education in Poland under work No. MB/WE/5/2018.

Authors: A. Choroszucho (e-mail: a.choroszucho@pb.edu.pl), J. Bednarek (e-mail: j.bednarek@student.pb.edu.pl), M. Sumorek (e-mail: mateusz.sumorek@gmail.com) and J. Żukowski (e-mail: jakubzzukowski@gmail.com), Bialystok University of Technology, Faculty of Electrical Engineering, Wiejska 45D Str., 15-351 Bialystok, Poland.

References

[1] Espressif Systems. (September 2020). Espressif.com [Online]. Accessed: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf.

[2] Tollervey N. H., "Programming with MicroPython: Embedded Programming with Microcontrollers and Python," O'Reilly Media (2017).

[3] A. Choroszucho, P. Golonko, J. Bednarek, M. Sumorek, J. Żukowski, "Comparison of high-level programming languages efficiency in embedded systems," *SPIE 11176, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2019*, (6 November 2019); doi: 10.1117/12.2537143, 2019

[4] Borwein J.M., Borwein P.B., Bailey D.H., "The American Mathematical Monthly," 96(3), 201–219, Mathematical Association of America (March 1989).

[5] A. Steckiewicz, A. Choroszucho, "Optimization-based synthesis of a metamaterial electric cloak using nonhomogeneous composite materials," *Journal of Electromagnetic Waves and Applications*, Vol. 33, No. 14, 1933–1941 (2019).
[6] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms," The MIT Press (1989), pp. 15–38.
[7] Debnath, Lokenath, "The Legacy of Leonhard Euler: A Tricentennial Tribute," World Scientific (2009).