# Analysis of energy consumption for SPI and I2C communications in ultra-low power embedded systems

Josip Zidar*, Ivan Aleksi*, Tomislav Matić*
* Faculty of Electrical Engineering, Computer Science and Information Technology, Osijek
jzidar@ferit.hr; ialeksi@ferit.hr; tmatic1@ferit.hr

***Abstract*—Developing ultra-low-power devices requires low-power components, algorithms, and communication protocols. For environmental monitoring along the supply chain, products may travel a long way from the distributor to the customer. In addition, extreme climatic circumstances, such as temperatures below 0 °C, affect battery capacity. In order to monitor package's environment, it is crucial to ensure that all embedded system parts use as little energy as possible. This paper addresses how much power the SPI and I2C communication protocols use. These protocols are widely used in low-power embedded devices for sensor communication. Based on its results, this research article presented recommendations for developing ultra-low power embedded devices that use the SPI/I2C protocol.**

***Keywords*—I2C, SPI, low power, embedded system, supply chain**

## I. INTRODUCTION

Inter Integrated Circuit (I2C) and Serial Peripheral Interface (SPI) communication protocols are typically used in modern embedded systems to communicate with on-board peripherals at low or medium data transfer rates. Article [1] discusses the different parameters that must be considered while picking a protocol, including features, advantages, disadvantages, the number of devices, data transfer rate, and power consumption.

I2C is a serial communication protocol invented by Philips labs that allows numerous devices to be linked to a single bus, with one device acting as the controller and the others as targets. I2C is commonly used for low-speed devices, like microcontrollers, sensors, actuators, real-time clocks, monitoring devices, power management devices, memories, and so on, to communicate with each other. I2C allows for the connection of up to 128 devices to a single bus, and it is able to support communication speeds of up to 3.4 Mbps. The I2C standard's most recent modifications are in [2].

SPI is a synchronous serial communication protocol originally developed by Motorola that permits several devices to be linked to a single bus, with one device acting as the controller and the others functioning as targets. On an embedded system, SPI is often used for communication between high-speed devices, including control devices, cameras, display controllers, communications, and memory banks. Supported are communication speeds of up to 60 Mbps. The number of target devices on a bus can range from one to several hundred, but is typically restricted to a dozen (e.g., four, eight, or sixteen). Certain specialised hardware implementations may be able to accommodate numerous target devices, each with its own chip-select pin. Consequently, the number of available input/output (I/O) pins on a microcontroller represents significant constraints. It is possible to utilise additional hardware, like an I/O expander or a multiplexer, to overcome the constraint on the number of available I/O pins [3].

I2C is commonly used for connecting low-speed devices, whereas SPI is used for connecting high-speed devices. I2C is simpler, more straightforward, and slower than SPI, which is more complicated, sophisticated, and faster. Although both protocols coexist and are utilised for inter-chip communication, designers of ultra-low power embedded systems should consider the energy consumption of the implemented design.

I2C communication protocol's main advantages are:

- communication requires only two wires, and the microcontroller requires fewer pins
- multi-target functionality enables communication with multiple target devices which are connected to the same bus
- easy to implement and does not require any special hardware or software
- supports data transfer rates of up to 3.4 Mbps
- supports multi-controller systems and is adaptable to a broad range of applications
- flow control and error handling, which results in a more reliable protocol.

I2C communication protocol's main disadvantages are:

- half-duplex communication, in which only one direction of communication is possible at once
- restricted communication range, not suitable for long-distance use
- in a multi-controller system, there is a possibility of address collision between devices
- a maximum of 128 devices are allowed
- comparatively slower than other communication technologies such as SPI or RS-232
- limited data transfer capabilities may not be suitable for applications requiring high speeds.

SPI communication protocol's main advantages are:

- full-duplex communication permits the simultaneous transfer of data in both directions
- well-defined interface and intuitive signalling
- It enables data transfer rates ranging from a few kilobits per second to 60 Mbps, making it suited for high-speed applications
- addressing-free, making it an efficient protocol for communicating with a single peripheral device
- supports numerous target devices, allowing a single controller to communicate with multiple targets.

SPI communication protocol's main disadvantages are:

- restricted communications up to a few metres in range
- chip select lines determine the maximum number of devices connected to the bus
- each extra target device adds four new wires to the circuit board layout
- needs additional circuitry or software to detect and repair faults since it lacks a built-in error detection mechanism.

The current state-of-the-art in low-power I2C and SPI communication focuses on reducing power consumption while preserving or enhancing communication speed and reliability. This is accomplished by combining hardware and software optimisation with low-power FPGA (Field-Programmable Gate Array) components.

Authors in [4] accurately simulated a reduced power model for low-power systems using peripherals. The method can be applied to various instruction sets and peripherals. Power supply fluctuations and number-rounding inaccuracies generate energy estimating issues, according to the research. This work introduces a new power measuring method using high-precision power measurement hardware and low-level software control to accurately evaluate power consumption.

In [5], the authors proposed a low-power synchronous data line protocol design that attempts to reduce energy consumption by utilising only two data lines, which is beneficial for PCB design. The proposed system consumed 12 mW, while the standard 3-Wire SPI design consumed 19 mW of power, which is a significant improvement. It also enables simplex and half-duplex data transfer at the same rate. The design was validated using FPGA as the controller and the chip as the target.

The authors of [6] provide an exhaustive evaluation of current power efficiency techniques and introduce new ones focused on lowering the quantity of data delivered and processed by devices. It aims to improve the power efficiency of the I2C communication protocol by reducing leakage current. The ıTurbo approach formulates optimal pull-up resistors. With increasing SDA and SCL pull-up resistance values, energy usage is significantly reduced as the current decreases.

Gated and non-gated low-power SPI switch interfaces have been compared by researchers in [7]. The authors intend to reduce the SPI switch interface's power consumption while preserving its performance. The paper describes the design and implementation of the FPGA XP2 low-power SPI switch interface. The results demonstrate that the proposed low-power SPI switch interface consumes much less power than existing methods while maintaining high performance.

The authors of [8] compare the power consumption of various serial interfaces, such as Universal asynchronous receiver-transmitter (UART), SPI, I2C, and others, in order to identify the most power-efficient approach. The report compares the results of a comprehensive analysis of the power consumption of several serial interfaces under various scenarios. The results suggest that the I2C interface is the most power-efficient alternative for low-data-rate applications, but the SPI interface is better suited for high-data-rate applications. Also, the authors present various optimisation strategies to improve the power efficiency of serial interfaces.

The paper [9] describes the implementation of both the I2C and SPI protocols on an FPGA. The authors evaluate these two FPGA-implemented protocols' performance, energy consumption, and design complexity. While implementing these protocols on an FPGA, the authors also discuss the design tradeoffs and difficulties involved. The results indicate that the SPI protocol has superior performance and lower power consumption per Mbps than the I2C protocol, although the I2C protocol has a simpler design.

The framework implementation in [10] involves the implementation of the Flex-SPI protocol into a software architecture, which provides a basis for communication between various devices. The Flex-SPI protocol is characterised by a comprehensive study of its performance, reliability, and power consumption. The energy consumption analysis of the Flex-SPI protocol involves measuring the power utilised by the communication link in various data transfer scenarios. This information is then compared to the I2C standard's energy consumption.

In [11], the energy consumption of SPI and I2C with various pull-up resistors was compared. They determined that a pull-up resistor higher than 47 kΩ would require a reduction in parasitic bus capacitance to function properly. SPI transfer was finished approximately 200 milliseconds faster than predicted, but I2C took 50 ms longer (with accounted NACK and CONTINUE commands on the bus). The influence of bus capacitance on the communication protocol causes the timing disparity.

In [12], the authors suggest a design for a digital circuit that can convert signals between the SPI and I2C communication protocols, presenting a novel approach to SPI and I2C. The control module employs state machines to regulate the data transmission between the two interfaces and guarantees that the communication adheres to the protocol specifications of each interface. The simulation results demonstrate that the conversion circuit can convert signals between the two protocols while preserving the data's integrity.

This article compares the power consumption of I2C and SPI communications on the same chip (eg. register write, register read, ...). This operation can be carried out using a microcontroller connected to a chip supporting the I2C and SPI communication protocols.

The rest of the paper is organised as follows. Section II defines the experimental setup used to perform measurements. Results and discussions are presented in Section III, while Section IV concludes the paper and gives future remarks.

## II. SETUP

Two hardware configurations are compared:

1) microcontroller connected to I2C devices (Fig. 1)
2) microcontroller connected to SPI devices (Fig. 2).

I2C contains two pull-up resistors, the resistance of which influences the energy consumption results. Energy consumption of pull-up resistors decreases with decreasing resistance values adjusted to the level of reliable signal integrity for the I2C connection with the wire length chosen.

I2C bus impedance and signal rise time decrease with lower pull-up resistor values but it might affect signal quality causing voltage drops and bus noise. Lowering the pull-up resistor increases data transmission rate, but it must be done carefully to avoid damaging the bus and devices. Reading bus devices datasheets and specs makes it easier to choose a pull-up resistors value.

Only one device is connected to the SPI or I2C bus when measurements are performed.

Table I gives a list of structural characteristics in order to properly specify the setup. It is important to note that 47 kΩ pull-up resistors are the most energy-efficient option for I2C connections as per [11]. Well estimated pull-up resistors may be a deciding factor in terms of power usage throughout the hardware design phase.

The chip select (CS) pin of the controller is set to push-pull GPIO mode. Typically, pull-up resistors on chip select lines are advised for systems where chip select pins of the controller may enter an undefined state. Added pull-up resistor would increase power consumption, which is undesirable for this research.

TABLE I: The characteristics of the measuring setup.

| *Characteristic* | *Value* |
|---|---|
| *MCU model* | STM32L010K4T6 |
| *I2C/SPI device* | LPS22HBTR |
| *Device type* | MEMS pressure sensor |
| *MCU operating frequency* [KHz] | 1048 |
| *Operating voltage* [V] | 3.3 |
| *I2C pull − up resistor values* [kΩ] | 47 |
| *Used wire lenght* [cm] | 15 |
| *SPI clock polarity* (*CPOL*) | High |
| *SPI clock phase* (*CPHA*) | 2 Edge |
| *Power supply* | GW Instek PSB-1400L |
| *Measuring multimeter* | Keysight 34465A |

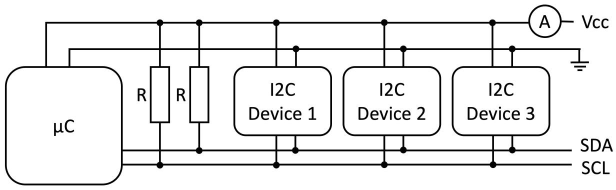


Fig. 1: I2C configuration diagram.

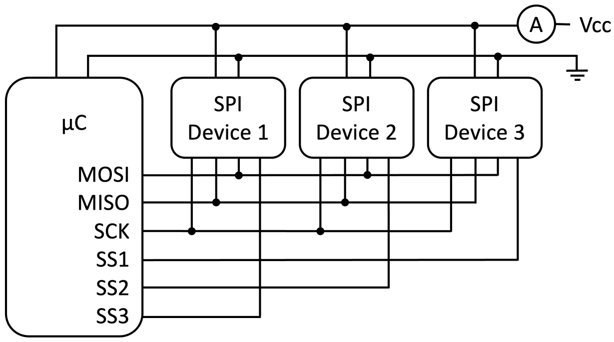


Fig. 2: SPI configuration diagram.

## III. EXPERIMENTAL RESULTS

The power consumption of I2C and SPI interfaces depends on several factors, including clock frequency, voltage level, and implementation details. I2C often consumes less power than SPI due to its slower clock speed and more efficient data transport protocols. However, the actual power consumption of I2C versus SPI can vary considerably based on the application and other variables, such as power control tools, which can have a substantial impact. However, it is recommended to consult the utilised devices' datasheets to ascertain each interface's actual power consumption.

Current measurements were performed with Keysight 34465A 6 ½ digit multimeter (listed in Tab. I) that provides high level of accuracy, speed and resolution. The microcontroller model and clock frequency are shown in Tab. I, as well as the sensor used as the target device.

According to the measured values in Tab. II and Tab. III, the distinction between open drain output and push-pull output is apparent. When the I2C bus idles, both SDA and SCL signals are in the logic high. In that period, the lowest energy consumption is achieved. This is due to the I2C bus consuming energy on every logic low signal, which happens when sending or receiving data.

Because of different data transfer approaches, I2C uses KHz while SPI uses KBit/s as a measure of transfer speed. I2C utilizes synchronous communication, in which a controller handles the timing of data delivery. With I2C, the controller transmits a clock signal to the target device(s), which transmit or receive data. According to the standard, since the frequency of the clock signal directly correlates to the transfer speed, the KHz measure is used. SPI is also a synchronous serial communication protocol,

TABLE II: Current consumption during SPI communication with the device's registers.

| *SPI* | | |
|---|---|---|
| *Communication speed* [KBit/s] | *Read current* [$\mu$A] | *Write current* [$\mu$A] |
| idle | 458,92 | 458,92 |
| 4,093 | 269,57 | 606,45 |
| 8,187 | 270,07 | 686,12 |
| 16,375 | 271,44 | 867,54 |
| 32,75 | 273,59 | 1.124,30 |
| 65,5 | 274,02 | 1.231,91 |
| 131 | 275,26 | 1.001,59 |

TABLE III: Current consumption during I2C communication with the device's registers.

| *I2C* | | |
|---|---|---|
| *Communication speed* [KHz] | *Read current* [$\mu$A] | *Write current* [$\mu$A] |
| idle | 270,05 | 270,05 |
| 4 | 523,49 | 539,17 |
| 8 | 505,96 | 519,05 |
| 16 | 475,77 | 486,67 |
| 32 | 441,18 | 456,16 |
| 65 | 403,80 | 417,72 |
| 131 | 376,97 | 404,45 |

but it is designed to operate at much higher speed rates compared to I2C. The duplex property of the SPI protocol enables data to be sent simultaneously. Read and write operations do not require equal clock signal cycles. Hence the transfer rate is measured in KBit/s.

According to SPI communication measurements, results have shown that reading data consumes less energy than writing data. For a read operation, the controller sends a data request, and the target device sends back the requested data by shifting it out on the MISO (Master In Slave Out) line. For the write operation, the controller sends both the address of the register and the data which will be written into the register. Since the target device has to write received data into the register, power consumption for the write operation rises. As a result, a read operation typically consumes less power than a write operation.

## IV. CONCLUSION

When designing ultra-low power embedded systems, it is recommended that all on-board peripherals use minimum energy. One of the aspects that affect energy consumption is the communication protocol used. This paper analyses the energy consumption of SPI and I2C communication protocols with speeds in the range of 100 KHz.

Based on the results, a conclusion can be made for three scenarios which can occur when utilizing SPI or I2C.

- Communication occurs sometimes, and the communication line spends in the idle mode most of the time. In this case I2C is more suitable when it comes to energy consumption.
- SPI is used to continuously retrieve data (i.e. measurements from sensor) without frequent register writes. In this case SPI is more suitable when it comes to energy consumption.
- Communication is balanced, a number of register reads and register writes are equally represented. In this case I2C is more suitable when it comes to energy consumption.

In applications with multiple I2C devices, results may vary due to the calculated resistance of pull-up resistors. It is also worth noting that number of read and write operations can also impact energy consumption when using SPI communication protocol, which isn't the case for I2C communication protocol.

Due to SPI not having a formal standard, and possible variations and options of SPI configuration, the results of this study are not uniform across the board. More experimental cases should be evaluated for modes which are not assessed in this research. Options such as SPI clock polarity (CPOL), clock phase (CPHA), and other types of SPI configurations, such as daisy chain or expander configuration should be considered.

Future work will include research for developing new ultra-low power communication protocol and developing methods for improving energy consumption in embedded systems.


## ACKNOWLEDGMENT

This work was funded by the European Union through the European Regional Development Fund, under project "Smart sticker for measuring and monitoring storage and transportation conditions of products" KK.01.1.1.04.0116.


## REFERENCES


[1] F. Leens, "An introduction to i2c and spi protocols," *IEEE Instrumentation & Measurement Magazine*, vol. 12, no. 1, pp. 8–13, 2009.

[2] *I2C-bus specification Rev 7*, NXP Semiconductors.

[3] *SPI Block Guide*, Motorola, Inc. [Online]. Available: https://www.nxp.com/files-static/microcontrollers/doc/ref_manual/S12SPIV4.pdf

[4] G. Berthou, K. Marquet, T. Risset, and G. Salagnac, "Accurate power consumption evaluation for peripherals in ultra low-power embedded systems," in *2020 Global Internet of Things Summit (GIoTS)*, 2020, pp. 1–6.

[5] S. L. Chen, T.-K. Chi, M.-C. Tuan, C.-A. Chen, L.-H. Wang, W.-Y. Chiang, M.-Y. Lin, and P. A. Abu, "A novel low-power synchronous preamble data line chip design for oscillator control interface," *Electronics*, vol. 9, p. 1509, 09 2020.

[6] Y. Song, S. Li, C. Zhang, S. Li, and L. Lu, "Rethinking power efficiency for next-generation processor-free sensing devices," *Sensors*, vol. 22, no. 8, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/8/3074

[7] S. Kaur and N. Sharma, “Designing and implementation of low power spi-switch interface on fpga,” *International Journal of Engineering Research and*, vol. V4, 05 2015.

[8] K. Mikhaylov and J. Tervonen, “Evaluation of power efficiency for digital serial interfaces of microcontrollers,” in *2012 5th International Conference on New Technologies, Mobility and Security (NTMS)*, 2012, pp. 1–5.

[9] A. Oudjida, M. L. Berrandjia, R. Tiar, L. Ahmed, and K. Tahraoui, “Fpga implementation of i2c & spi protocols: A comparative study,” 12 2009.

[10] P. Visconti, G. Giannotta, R. Brama, P. Primiceri, R. De fazio, and A. Malvasi, “Framework implementation, firmware development and characterization of flex-spi communication protocol: Energy consumption analysis and comparison with i2c standard,” *International Journal on Smart Sensing and Intelligent Systems*, vol. Vol.10, pp. pp. 754 – 792, 12 2017.

[11] T. Solheim and M. Grannæs, “A comparison of serial interfaces on energy critical systems,” in *2015 Nordic Circuits and Systems Conference (NORCAS): NORCHIP & International Symposium on System-on-Chip (SoC)*, 2015, pp. 1–4.

[12] D. Trivedi, A. Khade, K. Jain, and R. Jadhav, “Spi to i2c protocol conversion using verilog,” in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 2018, pp. 1–4.