

Power and Temperature Modeling of Raspberry Pis

Filip Idzikowski, Krzysztof Klimaszewski, Bartosz Witkowski

Faculty of Computing and Telecommunications, Poznan University of Technology, Poznań, Poland

{filip.idzikowski | krzysztof.klimaszewski}@put.poznan.pl, bartosz.r.witkowski@gmail.com

Abstract—Global electrical energy consumption is constantly increasing as is the number of ubiquitous Single Board Computers (SBCs) used globally. Therefore, the total power consumption of SBCs becomes an issue of increasing importance. The energy consumed by an SBC can be saved by changing its state (on, off, sleep) depending on demands. However, this induces temperature changes, which in turn decrease the device's lifetime. We perform measurements of power and temperature of two Raspberry Pis and use the measured values to develop load-dependent power and temperature models. These models can serve as inputs in future research on power savings and lifetime of SBCs. Our results cover the state-of-the-art Raspberry Pi 5 and Pi 4B with respect to load and taking into account memory transfers on artificial and realistic scenarios. We provide results in terms of power consumed not only by the whole SBCs, but also by separate components of Raspberry Pi 5. Performance of the SBCs in number of operations per Watt is also covered.

Index Terms—Single Board Computers, temperature model, power model, green computing and networking, devices' lifetime

I. INTRODUCTION

Information and Communications Technology (ICT) devices are estimated to consume 916 TWh of energy per year worldwide for 2020 [1]. SBCs contribute to this number even if a single SBC is a relatively small (potentially battery-powered) electricity consumer. The number of SBCs is constantly increasing, with the number of popular Raspberry Pis reaching 68 million sold between 2012 and 2025 [2]. We look at the electricity consumption of SBCs taking into account their wide usage and versatility. Furthermore, we look at the temperatures that SBCs achieve under different conditions focusing on temperature variations that influence the SBCs' lifecycle [3]–[5]. Two artificial scenarios with different Random-Access Memory (RAM) transfers and a realistic scenario performing image processing are used. The main contributions of this work are: (i) power models of the state-of-the-art Raspberry Pi 5 and Raspberry Pi 4B and (ii) temperature models of the SBCs in dependence on their load.

This work is structured as follows. An overview of related work is presented in Section II highlighting our contributions. We describe the methodology, measurement environment, and Devices Under Test (DUTs) in Section III. The two artificial scenarios and corresponding results are presented in Section IV and Section V, respectively. The scenario performing image processing is presented together with corresponding results in Section VI. Conclusions are drawn in Section VII.

This work was supported by the Ministry of Science and Higher Education, Poland.

II. RELATED WORK

We analyze related works on power or energy consumption of SBCs with the following criteria: (i) SBC used, and (ii) presence of device's temperature analysis. The result of our analysis is shown in Table I sorted according to the SBCs and showing measurements and models (power and temperature). The related works are discussed next.

A power model of Raspberry Pi 4 (not 5) is proposed in [6]. It is created using additive and regression models based on measurements of the power and utilization of the SBCs. Temperature is not measured in [6]. No temperature models are proposed there.

Energy consumption of Hybrid Edge Computing systems with embedded devices and Artificial Intelligence (AI) accelerators is addressed in [7]. The authors of [7] introduce a systematic methodology and a set of tools to benchmark AI accelerators and their host devices with interference workloads representing mock Convolutional Neural Network (CNN) models with varying input sizes, network sizes, layer types, and kernel sizes. The scope of [7] is different from our work even if Raspberry Pi 4 (not 5) is used in [7]. Neither temperature measurements nor models are included in [7].

A procedure for creation of an energy consumption model of SBCs is proposed in [8] based on the usage of their components. The proposed procedure is evaluated on a Raspberry Pi 2 (not 4B or 5). Neither temperature measurements nor temperature models are presented in [8].

WattEdge is proposed in [9] as a holistic approach to empirical energy measurements in edge computing devices. It can measure factors of energy consumption such as Central Processing Unit (CPU), memory usage, a combination of them, connectivity, bandwidth usage, and communication protocols. WattEdge is validated on a Smart Agriculture use case. No temperature measurements, no temperature models, and no power models are reported in [9].

The authors of [10] aim to determine the dependence of power consumption of a Raspberry Pi 2B on the key functionalities that the end-users could perform on the platform. These functionalities are compared with the functionalities of other types of common personal computers such as desktops, laptops, tablets, and smartphones. The main result of [10] is recommendations on techniques and practices that could reduce the power consumption of Raspberry Pis. The authors consider 20 different functionalities, from device start-up through watching a video to system shutdown. Even if [10] is

focused on software, five hardware-related recommendations are made to reduce Raspberry Pi power consumption. Neither temperature measurements nor temperature models nor power models are reported in [10].

A measurement campaign of Raspberry Pi 1B is described in [11]. The authors proposed power models based on CPU usage, Ethernet bandwidth usage (upload and download), WiFi bandwidth usage (upload and download) and combined models. Neither temperature measurements nor temperature models are presented in [11].

The work [11] is extended in [12], where in addition to the Raspberry Pi 1B, also the Raspberry Pi 2B, Raspberry Pi 3B, as well as Cubieboard3, Odroid-C1 and Odroid-C2 are analyzed. Neither temperature measurements nor temperature models are presented in [12].

The only work that tackled temperature of SBCs is [13]. However, it is motivated by a different research question, i.e., can SBC computing clusters consisting of up to 100 units (Raspberry Pi 2) be an alternative to traditional computing clusters? The authors identified the hot spots in the Raspberry Pis using Negative Temperature Coefficient (NTC) thermistors. They calculated the thermal power and experimentally validated the heat transfer coefficients available in the literature. Other conclusions made in [13] include the importance of having stable conditions (including air temperature) for temperature measurements and the use of heatsinks for effective cooling. It must be noted here that Raspberry Pi 2 is not commonly equipped with heatsinks, as is common for more recent models. Neither the temperature nor power models of any Raspberry Pi are proposed in [13].

TABLE I
POWER AND TEMPERATURE MODELS OF SBCs IN THE LITERATURE.

SBC Device	Measurements		Model	
	Power	Temp.	Power	Temp.
Pi 5	ours	ours	ours	ours
Pi 4	[6], B [7], ours	B ours	[6], B [7], ours	B ours
Pi 3	B [12], B+ [9] ^a	no	B [12]	no
Pi 2	[13]; B [8], [10], [12]	[13]	B [8], [12]	no
Pi 1	no	no	no	no
Pi 0	B [11], [12]	no	B [11], [12]	no

^aAn energy model rather than power model is proposed in [9].

As shown in the last column of Table I, there are no temperature models of Raspberry Pis in the related works. Furthermore, we tackle the most recent member of the Raspberry Pi SBC family, i.e., the Raspberry Pi 5, on top of the related works and provide an extensive analysis of results (temperature and power) in comparison to the Pi 4B. At the same time, we deliberately leave out peripherals and analyze pure SBCs.

III. MEASUREMENT ENVIRONMENT

The test setup shown in Fig. 1 is used to measure the parameters in question. Raspberry Pi 5 (Fig. 2(a)) and 4B (Fig. 2(b)) with the most recent version of Raspberry Pi Operating System (OS) at the time of writing [14] in 64-bit version are used as DUT (depicted as 1 in Fig. 1(a)

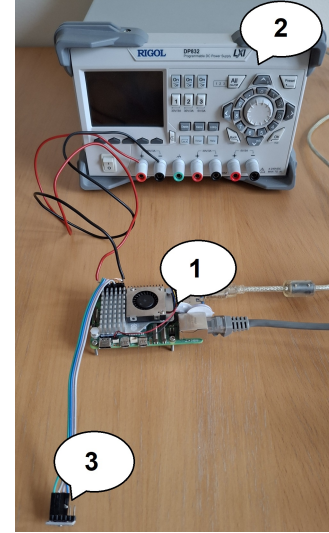
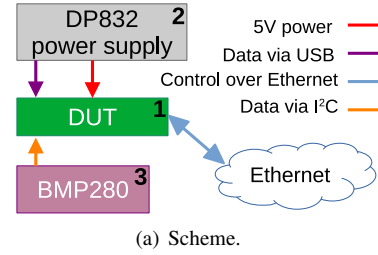


Fig. 1. Setup used in the experiment.

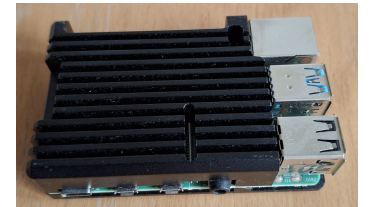
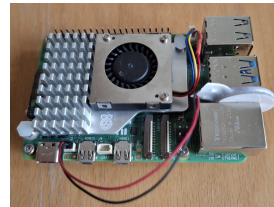


Fig. 2. The Raspberry Pis used in the experiments.

and Fig. 1(b)). All libraries and settings are preserved in original versions provided in the system image – no updates are performed. The OS for both DUTs, Pi 4B and Pi 5, is kept with default settings except for access to Secure SHell (SSH) and I²C and boots only to the command line.

Each DUT is powered with a Rigol DP832 power supply (depicted as 2 in Fig. 1(a) and Fig. 1(b)) at 5 V Direct Current (DC). The power supply is also connected with a Universal Serial Bus (USB) to each DUT to allow real-time power measurement. To measure ambient temperature, a breakout board with a BMP280 sensor [15] (depicted as 3 in Fig. 1(a) and Fig. 1(b)) is connected to each DUT I²C interface. The board and power supply are located on a separate desk in a laboratory with steady ambient temperature and no airflow. Each Raspberry Pi is controlled via an SSH connection provided with an Ethernet connection. No USB devices are connected

to each Raspberry Pi except for the connection to the power supply. No video outputs are connected to any output devices. The WiFi is in idle state, disconnected from any Access Point (AP). We believe that this configuration accurately depicts a significant amount of Raspberry Pis deployed worldwide.

The Raspberry Pi 5 used in the test setup is equipped with a standard heatsink with a fan, as shown in Fig. 2(a). The fan is controlled by a Pulse Width Modulation (PWM) signal provided by the OS. We use the default fan control algorithm with factory settings for trigger temperatures.

The Raspberry Pi 4B used in our experiments does not have a fan. We use a popular model of a Pi 4B case LT-4B03, shown in Fig. 2(b), acting as a passive heatsink for this version of DUT.

Two sets of scripts are executed during the measurements. The first set of scripts gathers data about the power delivered from the power supply, the CPU frequency, the load of the CPU, the CPU temperature, the heatsink fan PWM value, and the ambient temperature. The power management circuit in Raspberry Pi 5 also enables the measurement of currents and voltages for separate power domains on the board. Therefore, we also collect data about the core voltage (varying with the CPU frequency) and current as well as the 1.1 V, 1.8 V and DDR_VCC2 (1.1 V nominal) voltages and currents.

The second set of scripts applies the given set of loads to the computer for a given time, with details depending on the scenario, as described in the next section.

IV. SCENARIOS

We investigate two scenarios. In both scenarios, scripts that apply the load L (expressed as a percentage of the full load) to CPU are executed sequentially. Each script runs 4 parallel threads that start with a certain time offset. Different load values are obtained over the total time T by performing a set of calculations for a given time t_{load} within each thread:

$$t_{load} = T \cdot L / 100 \quad (1)$$

and then keeping the thread asleep for the time t_{sleep} :

$$t_{sleep} = T - t_{load}. \quad (2)$$

The actual algorithm used in the experiment is shown in Algs. 1 and 2. We set $T = 1$ second and vary L between 10% and 100%. Each experiment is repeated $N_R = 5$ times (line 2 in Alg. 1) to ensure reliability of the results. The cycle is repeated for 1000 seconds for each desired load setting. The measurement is repeated by sequentially adjusting the CPU load.

The two scenarios are differentiated by the type of load (line 2 in Alg. 2). The scripts are written in Python and interpreted at runtime. In the first scenario, the software performs simple floating point calculations not requiring a significant amount of RAM nor cache transfers for calculation. A single floating point multiplication, a division, and an addition are performed within within the loop.

In the second scenario, DUT runs software that repeatedly multiplies large matrices using the Numpy library (accelerated

by the BLAS library) to generate a specific load. A 400x400 floating point matrix is multiplied in each pass of the loop. In this way, the script requires a significant amount of data to be transferred to and from RAM. In each loop pass, approximately 64 million floating point multiplications and 64 million floating point additions are performed.

V. RESULTS

Our initial tests show that for each applied load, the state of SBC stabilizes after less than 200 s. Therefore, the results of the experiments are averaged over the selected trailing period of application of the given load L , so that the first 20% of the results for each period (200 s) are discarded and only the last 80% (800 s) are considered for averaging.

Algorithm 1 Algorithm for the single scenario experiment.

Input: $L_{TABLE} = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]$

Input: $N_R = 5$

Output: The set of data for analysis

```

1: Start data recording
2: for  $i = 1$  to  $N_R$  do
3:   for  $L$  in  $L_{TABLE}$  do
4:     Run 4 separate threads (Alg. 2) for 1000 seconds
5:     Sleep for 300 seconds
6:   end for
7:   Sleep for 500 seconds
8: end for
```

Algorithm 2 Algorithm for a single thread.

```

1: for  $k = 1$  to 1000 do
2:   Apply load for  $t_{load}$  time
3:   Sleep for  $t_{sleep}$  time
4: end for
```

A. Supply power

The dependence of the supply power for the entire board on the load L is shown in Fig. 3 for Raspberry Pis 4B and 5 together with the approximations generated with the nonlinear least-squares Marquardt-Levenberg algorithm [16]. The dependency is close to linear for both machines in Scenario I, but Pi 4B draws significantly less power than Pi 5. In contrast to that, a noticeable plateau can be seen at higher loads of Scenario II for both Raspberry Pi 4B and 5. The approximations shown in Fig. 3(a) for Raspberry Pi 4B are given by Eq. (3) and Eq. (4) for Scenarios I and II respectively. The approximations shown in Fig. 3(b) for Raspberry Pi 5 are given by the Eq. (5) and Eq. (6) for Scenarios I and II respectively. Despite our initial tests with various fitting functions that result in higher order load functions of L , we use the first order polynomials in order to obtain the simplest possible models. Apart from simplicity, the first order model avoids problems with non-monotonicity apparent for higher order models without any significant improvements on its quality. For Scenario II, however, a straight line significantly

deviates from the data points. We therefore split the load range into two separate ranges and perform line approximation separately. The load range is split at the value of 56.2% for Pi 4B, while the range is split at the value of 67.7% for Pi 5. These selections are made in order to ensure continuity of the approximation lines for the entire analyzed load range from 10% to 100%.

$$P_{Pi4-SI} = 0.03664 \cdot L + 2.1995 \quad (3)$$

$$P_{Pi4-SII} = \begin{cases} 0.0403 \cdot L + 2.1327 & \text{for } L \in \langle 0, 56.2 \rangle \\ 0.00519 \cdot L + 4.104 & \text{for } L \in (56.2, 100) \end{cases} \quad (4)$$

$$P_{Pi5-SI} = 0.06258 \cdot L + 2.9422 \quad (5)$$

$$P_{Pi5-SII} = \begin{cases} 0.1115 \cdot L + 2.3006 & \text{for } L \in \langle 0, 67.7 \rangle \\ 0.0304 \cdot L + 7.7908 & \text{for } L \in (67.7, 100) \end{cases} \quad (6)$$

B. CPU temperature and frequency

The dependence of CPU temperature on load L is shown in Fig. 4 for both DUTs. We approximate this dependency with linear functions for both scenarios in each of the two DUTs. Eqs. (7) and (8) describe the behavior of Raspberry Pi 4B for Scenarios I and II, respectively. The formula for Raspberry Pi 5 for Scenario I is given by Eq. (9), while the approximation for Scenario II is given by Eq. (10). The ambient temperature varied between 23.5°C and 24.4°C during the experiments, occasionally (twice) reaching 26°C. Thermal throttling is not observed at higher loads, i.e., the CPU frequency remains at the highest possible CPU clock frequency for both Raspberry Pi 4B and 5 as shown in Fig. 5. It can be seen that Scenario II requires a higher frequency compared to Scenario I for the same load for both tested DUTs. The noticeable hump in the temperature for loads of approximately 30-40% for Pi 5 is caused by the fan speeding up (compare with Fig. 6(a)) and reducing the temperature rise.

$$T_{Pi4-SI} = 0.2967 \cdot L + 43.2141 \quad (7)$$

$$T_{Pi4-SII} = 0.1881 \cdot L + 45.0579 \quad (8)$$

$$T_{Pi5-SI} = 0.2039 \cdot L + 44.8761 \quad (9)$$

$$T_{Pi5-SII} = 0.2479 \cdot L + 45.0835 \quad (10)$$

C. SBC components

The new Raspberry Pi 5 provides means of monitoring currents and voltages of power supply lines of most of its components. This is not the case for the Raspberry Pi 4B.

We examine the heatsink fan, CPU, and RAM of the Raspberry Pi 5. The power of the 1.1 V line does not change noticeably during tests and has a value of approximately 0.213 ± 0.004 W for the entire L range. It is therefore not discussed further.

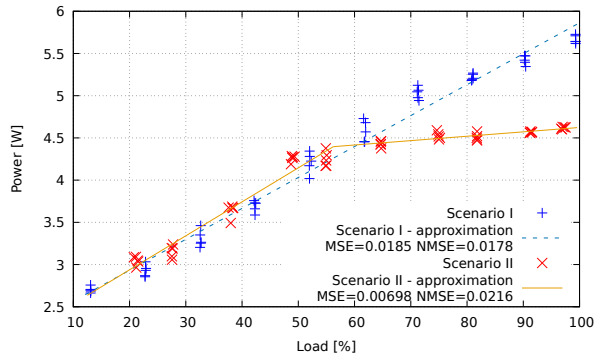
1) *Influence of fan operation on the temperature and power consumption:* The speed of the heatsink fan increases as CPU temperature increases. The speed of the fan is controlled by a PWM signal applied to the fan controller. The PWM values are preset to 0, 75, 125, 175, and 250 depending on the CPU temperature. The power consumed by the fan is shown in Fig. 6(a). It is calculated based on the following values: (i) PWM recorded during the experiments, (ii) current for different PWM values, and (iii) voltage measured on the fan connector. Measurements are performed independently by cutting the power conductor of the fan and using a multimeter. The power of the fan exceeds 0.4 W for the measured 100% load, which corresponds to approximately 4.4% of the power delivered from the power supply for Scenario I and 4.1% for Scenario II. It should be noted that the step nature of the fan operation, which can be observed in Fig. 6(a), influences both the total power and the temperature of the Raspberry Pi 5 shown in Fig. 3(b) and Fig. 4(b).

2) *CPU core power consumption:* CPU is an SBC component that consumes the majority of the SBC power, as can be seen in Fig. 6(b) compared to Fig. 3(b). A similar amount of power with similar load dependency can be observed in Fig. 6(b) for both scenarios. Slightly higher power consumption is observed for Scenario II than for Scenario I at the highest load values. The Scenario I has slightly higher consumption than Scenario II for lower loads. This behavior is consistent for all 5 separate runs of the experiment.

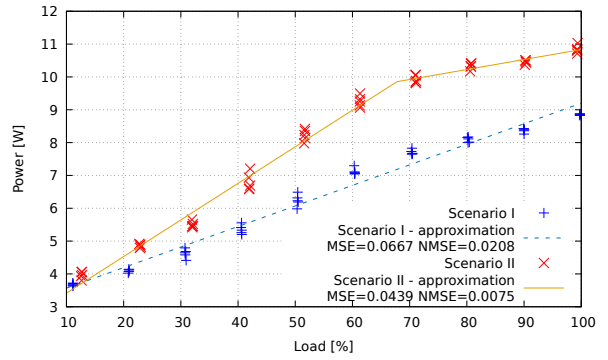
3) *RAM power consumption:* We monitor the two lines that supply power to RAM of the Raspberry Pi 5, i.e., of the 1.8 V rail (this line supplies RAM VCC1 and some other components) and RAM VCC2. The power consumed on these two lines is shown in Fig. 6(c). The power consumed by RAM remains stable in Scenario I, while the power delivered to RAM from the VDD2 and 1.8 V lines increases with increasing load L for Scenario II. However, it must be noted that even for the highest measured loads, the additional power for RAM is lower than the power for the heatsink fan shown in Fig. 6(a).

D. Computational performance

Due to significant differences in power consumption and computing power between Raspberry Pi 4B and Raspberry Pi 5, a comparison of computational performance is made for these devices. During the previous experiments, we measured the amount of time needed to perform a certain number of load loops. This experiment is performed separately. Our measurements are performed for a single thread and full load. The results show that Pi 5 is able to perform the same amount of load loops approximately 2.6 times faster than Pi 4B for Scenario I and approximately 2.5 faster for Scenario II. Table II shows calculated results in terms of performance index defined as the performance (number of operations) per Watt normalized to results obtained for Pi 4B. Pi 5 is more energy efficient than Pi 4B, especially for Scenario I, for which the calculations are performed without significant transfers to and from memory. In such a case, Pi 5 is able to

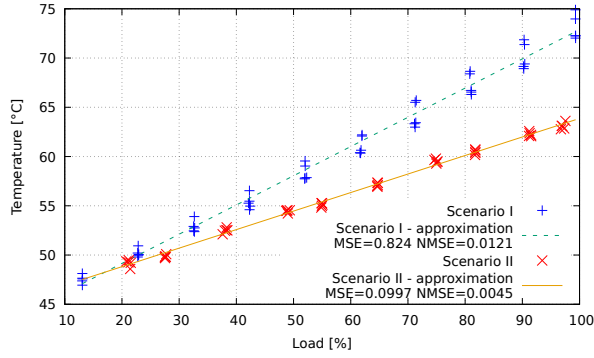


(a) Pi 4B.

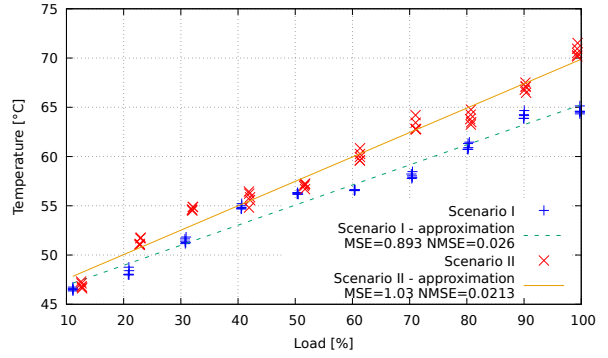


(b) Pi 5.

Fig. 3. Power dependency on load for DUTs together with Mean Squared Error (MSE) and Normalized Mean Squared Error (NMSE) values.



(a) Pi 4B.



(b) Pi 5.

Fig. 4. CPU temperature dependency on load for DUTs together with MSE and NMSE values.

TABLE II
COMPUTATIONAL PERFORMANCE INDEX COMPARISON (HIGHER BETTER).

SBC model	Scenario	# of loops / second	Performance index
Pi 4B	I	2213560	1.0
Pi 4B	II	96	1.0
Pi 5	I	4987818	1.46
Pi 5	II	239	1.06

perform approximately 1.46 times the number of calculations per Watt of consumed power compared to Pi 4B. The results are given for interpreted Python code exactly as used for the loading scripts described in Section IV.

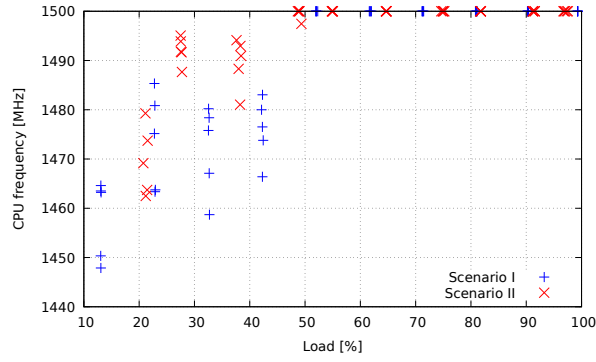
VI. PRACTICAL APPLICATION

We perform additional tests with realistic workload, where Pis run a recent YOLO26 object detector [17] that is optimized for edge devices. The Small version of the detector is chosen for both Pi 4B and Pi 5, and additionally, due to the much greater computing power of Pi 5, we test the Large version on Pi 5. The detector works on images with resolution 640x384 pixels for Pi 5, while the resolution is reduced to 320x192 pixels on Pi 4B. The images used for the tests are captured by a camera connected to the Pi in order to simulate the actual probable use case. The same images are used for the duration

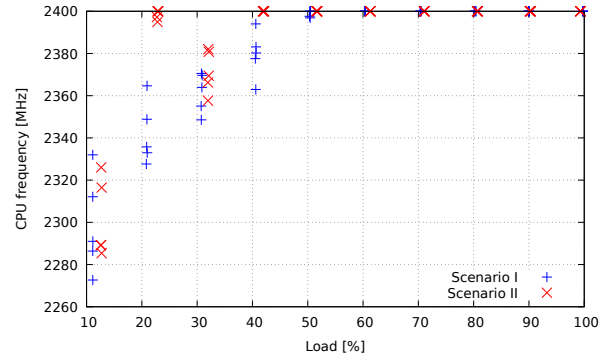
of a single test (one data point in the results) to avoid the influence of the time and power needed to capture a frame or read it from the disk or from the network. The images show an empty classroom with a few detectable objects during both daytime and night. Results of the tests, together with the approximations presented earlier, are shown in Fig. 7 and Fig. 8. It can be seen that the points generally follow the Scenario I approximation for both power and temperature. For Pi 5 and the Large version of the detector, the points however tend towards the line of approximation for Scenario II. This is justified by the fact that the larger model requires more weights and it therefore performs more reads and writes to RAM, which makes its behavior more similar to the Scenario II. The results of the additional tests show that the characteristics obtained for artificial loads from Scenarios I and II form an approximate boundary for the real-life loads.

VII. CONCLUSIONS

We propose models for supply power and CPU temperature dependence on the total load for Raspberry Pi 5 and Raspberry Pi 4B as representative state-of-the-art SBCs. We show that the actual model has to take into account the nature of the load applied to an SBC, since there is a significant difference between loads that use little memory and those that cause frequent data exchanges between CPU (and its cache memory)

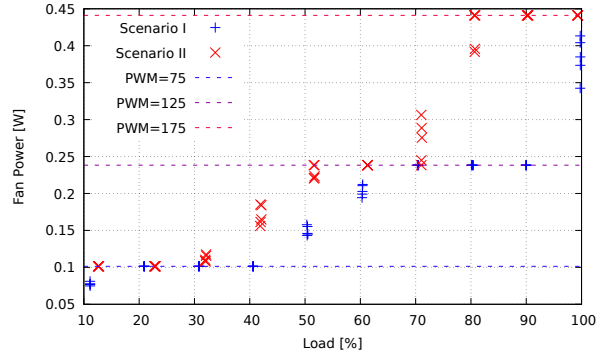


(a) Pi 4B.

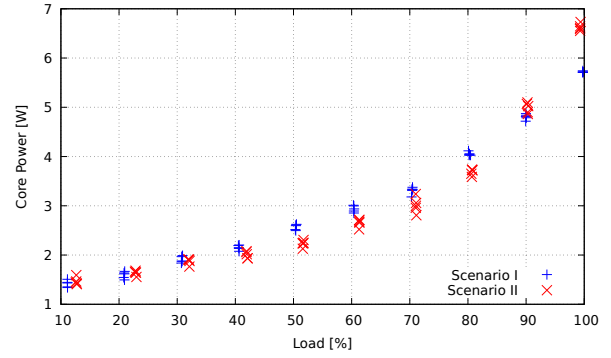


(b) Pi 5.

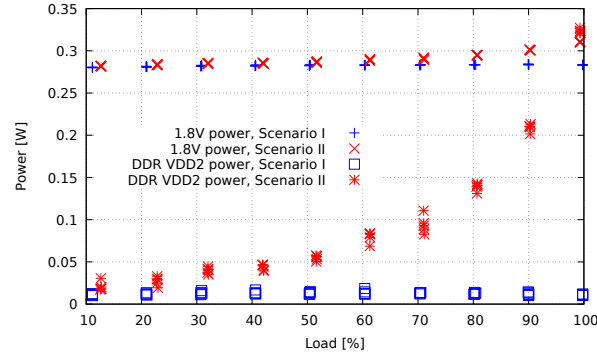
Fig. 5. Average CPU frequency value.



(a) Heatsink fan.



(b) CPU core.

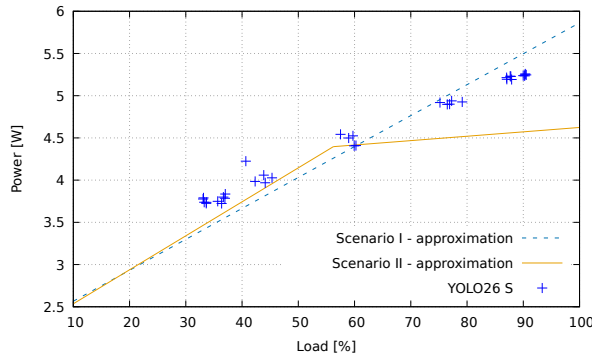


(c) RAM.

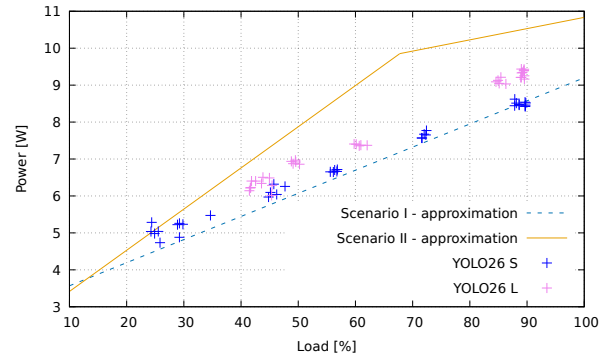
Fig. 6. Average power consumption values for Pi 5 components. The horizontal dashed lines denote power for PWM values used by the OS control mechanism. RAM power stems from 1.8 V rail (RAM VCC1 and other components) and RAM VCC2.

and RAM. The temperature characteristics of the Raspberry Pi 4B and 5 CPUs are found to be similar, although the power delivered to the computer is approximately twice as high for the Raspberry Pi 5. The maximum temperature observed during our experiments is 75°C . We extend our study by looking at the power consumed by separate components of the Raspberry Pi 5, i.e., heatsink fan, CPU core, and RAM. We confirm that CPU is the most power-hungry component and identify the stepwise operation of the heatsink fan. We also present the comparison of energy efficiency for the considered models of Raspberry Pi, showing Pi 5 to be more efficient

than Pi 4B in terms of energy consumption per calculation. Influence of peripherals are out of scope of this work. Their influence on SBC power consumption is limited with wireless connectivity expected to be the biggest contributor according to [11]. We plan to extend this work in the future by studying the dependency of actual computing power (instead of load L) on supply power for different Pis and SBCs from other vendors, varying the loads to verify whether composite loads (Scenario I and II combined) will result in varying power to load characteristics. We also plan to vary ambient temperatures and to verify measurements over multiple SBCs of

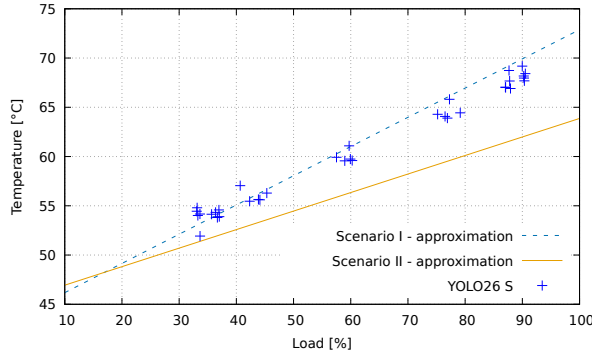


(a) Pi 4B.

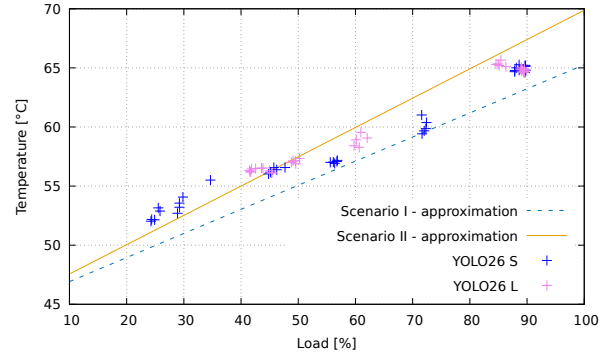


(b) Pi 5.

Fig. 7. Power dependency on load for YOLO26. “L” denotes Large model, “S” denotes Small model.



(a) Pi 4B.



(b) Pi 5.

Fig. 8. Temperature dependency on load for YOLO26. “L” denotes Large model, “S” denotes Small model.

the same type. An extended study on AI applications is also envisaged. The presented results offer input data for further studies on e.g., temperature changes that can damage CPU of an SBC.

REFERENCES

- [1] J. Malmodin, N. Lövehagen, P. Bergmark, and D. Lundén, “ICT sector electricity consumption and greenhouse gas emissions – 2020 outcome,” *Telecommunications Policy*, vol. 48, no. 3, p. 102701, 2024.
- [2] Raspberry Pi Holdings, “Computing for everybody, Final Results for the twelve months to 31 December 2024,” <https://investors.raspberrypi.com/reports/5/presentation>, last accessed 11.05.2026, April 2025.
- [3] L. Chiaraviglio, P. Wiatr, P. Monti, J. Chen, L. Wosinska, J. Lorincz, F. Idzikowski, and M. Listanti, “Impact of energy-efficient techniques on a device lifetime,” in *2014 IEEE Online Conference on Green Communications (OnlineGreenComm)*, 2014.
- [4] C. Natalino, L. Chiaraviglio, F. Idzikowski, C. R. L. Francês, L. Wosinska, and P. Monti, “Optimal lifetime-aware operation of green optical backbone networks,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3915–3926, 2016.
- [5] M. Schmid, A. Zippelius, A. Hanß, S. Böckhorst, and G. Elger, “Investigations on high-power LEDs and solder interconnects in automotive application: Part ii—reliability,” *IEEE Transactions on Device and Materials Reliability*, vol. 23, no. 3, pp. 419–429, 2023.
- [6] M. Kasioulis, M. Symeonides, G. Pallis, and M. D. Dikaiakos, “Power estimation models for edge computing devices,” in *European Conference on Parallel Processing*. Springer, 2023, pp. 257–269.
- [7] M. Kasioulis, M. Symeonides, G. Ioannou, G. Pallis, and M. D. Dikaiakos, “Energy modeling of inference workloads with AI accelerators at the edge: A benchmarking study,” in *2024 IEEE International Conference on Cloud Engineering (IC2E)*. IEEE, 2024, pp. 189–196.
- [8] L. Ardito and M. Torchiano, “Creating and evaluating a software power model for linux single board computers,” in *Proceedings of the 6th International Workshop on Green and Sustainable Software*, 2018.
- [9] M. S. Aslanpour, A. N. Toosi, R. Gaire, and M. A. Cheema, “Wattedge: a holistic approach for empirical energy measurements in edge computing,” in *International Conference on Service-Oriented Computing*. Springer, 2021, pp. 531–547.
- [10] G. Bekaroo and A. Santokhee, “Power consumption of the Raspberry Pi: A comparative analysis,” in *2016 IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies (EmergiTech)*, 2016.
- [11] F. Kaup, P. Gottschling, and D. Hausheer, “PowerPi: Measuring and modeling the power consumption of the Raspberry Pi,” in *39th Annual IEEE Conference on Local Computer Networks*, 2014, pp. 236–243.
- [12] F. Kaup, S. Hacker, E. Mentzendorff, C. Meurisch, and D. Hausheer, “The Progress of the Energy-efficiency of Single-board Computers,” Networks and Distributed Systems Lab, Otto von Guericke Universität Magdeburg, Germany, Tech. Rep. NetSys-TR-2018-01, January 2018.
- [13] M. N. Pujiula, “Thermal study of a Raspberry PI for the construction of a small-sized supercomputer,” Bachelor’s thesis, Escola Tècnica Superior d’Enginyeria Industrial de Barcelona, June 2016.
- [14] Raspberry Pi OS - Debian Bookworm from 13.05.2025. (accessed 01.04.2026). [Online]. Available: <https://www.raspberrypi.com/software/operating-systems/>
- [15] Bosch BMP280 environmental sensor. (accessed 11.05.2026). [Online]. Available: <https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmp280-ds001.pdf>
- [16] Gnuplot software. (last accessed 11.05.2026). [Online]. Available: <http://www.gnuplot.info/>
- [17] Ultralytics, “YOLO26 detector description,” <https://docs.ultralytics.com/models/yolo26> (last accessed 11.05.2026).