Evaluation of Embedded System Platforms for Automatic Parking System Control Unit

Wahyu Dewanto[1], Agung Fathurrahman[2], Agus Bejo[3]

[1] Department of Electrical and Information Engineering, Faculty of Engineering, Universitas Gadjah Mada, Jl. Grafika No. 2 Kampus UGM, Yogyakarta 55281 INDONESIA (tel.: 0274-555 225; fax: 0274-4321982, email: [1]wahyud@ugm.ac.id, [2]agung.fathurrahman@mail.ugm.ac.id, [3]agusbj@ugm.ac.id)

[Received: 2 January 2023, Revised: 17 July 2023]
Corresponding Author: Agus Bejo

ABSTRACT — Automatic parking system is one of the parking management technologies that is widely used in various institutions today. An automatic parking system works by controlling a parking gate automatically to open and close the gate and record the vehicle's license plate when entering and exiting using access control such as a smart card or radio frequency identification (RFID). One of the challenges in implementing an automatic parking system is traffic congestion during high traffic conditions. This challenge arises because the control unit in the automatic parking system takes a relatively long time to process and store images from the camera. This research examined several embedded system platforms as automatic parking system control units, including Raspberry Pi 3B, Raspberry Pi 4B, and Orange Pi Zero Plus. The evaluation is intended to find the best control unit platform based on several criteria, such as the execution time in capturing images, storing images, and the consumed power. From the evaluation results, it can be concluded that the Raspberry Pi 4B platform results in the fastest execution time for capturing and storing images, with an average time of 1,827.9 ms. Meanwhile, the Orange Pi Zero Plus platform achieves the lowest power consumption at 1.9 W. Based on the evaluation results, the Raspberry Pi 4B is recommended as the control unit if the automatic parking system requires a high-performance device. Otherwise, the Orange Pi Zero Plus is more recommended if the automatic parking system requires a low-power device.

KEYWORDS — Automatic Parking System, Embedded System, Smart Card, Camera.

I. INTRODUCTION

A parking system is one of the facilities that an institution must normally provide. One of the functions of having a parking system is to increase security at the institution. In practice, two common methods are used in the parking system, namely manual parking system and automatic parking system. The manual parking system involves several attendants at various parking spots and giving drivers a ticket as proof of vehicle ownership. The manual parking system may cause queues during high traffic periods due to slow service. To speed up the process, an automatic parking system can be implemented. An automatic parking system uses a parking gate machine that records the vehicle's license plate and automatically issues a ticket to the driver.

In addition to using a ticket as proof of access, automatic parking systems generally can also utilize a smart card as proof of access [1]–[3]. A smart card is a plastic card that contains an integrated circuit (IC) to store and process information. Accessing systems using smart cards are pretty widely implemented in Indonesia. At least 13,000 units of smart card reader machines are circulating in Indonesia and recorded by Badan Pengembangan dan Pengkajian Teknologi (BPPT) [4]. The automatic parking system is widely used because of its ease of use and more economical cost. This system has also been implemented in the Faculty of Engineering at Universitas Gadjah Mada (UGM) by utilizing faculty cards, employee cards, and student cards since all three cards are already in the form of a smart card.

However, in its implementation, automatic parking systems sometimes still have drawbacks, such as requiring a significant amount of time to record vehicle numbers and open as well as close the parking portal. During high traffic, these processes can lead to congestion at the parking portal's entrance and exit. The long process time is caused by the longer time required to take photos and store image files determined by the hardware platform and memory card used.

Therefore, this research evaluates several control units in the automatic parking system used in the Department of Electrical and Information Engineering, UGM environment. The control units used in this research are three single-board computers with different models, including Raspberry Pi 3B, Raspberry Pi 4B, and Orange Pi Zero Plus. Single-board computer-based platforms such as Raspberry Pi are frequently used for designing embedded system-based systems [5], [6]. In each control unit, an evaluation is conducted to compare the performance of both computational speed and power consumption of each control unit used. From the performance testing results, a recommended good control unit device can be used for the automatic parking system.

II. SYSTEM DESIGN

A. *DESIGN OF PARKING PORTAL SYSTEM*

The parking portal system generally consists of several components: parking portals, control units, cameras, and card readers or buttons can be used instead [7]–[9]. The card reader is one type of card acceptance device (CAD), which is a device that can communicate with a smart card. The card reader needs to be serially connected to the host computer for communication. In this study, the control unit is used as the host computer. The designed parking portal system uses several components, that is, the control unit, smart card reader, CCTV camera, USB-RS485 converter, and parking portal. The general scheme for the designed system can be seen in Figure 1.

This research used three embedded system platforms as the control unit: Raspberry Pi 3B, Raspberry Pi 4B, and Orange Pi Zero Plus. The specifications of these three embedded system platforms are as follows:

- Raspberry Pi 3B uses Broadcom BCM2837 processor, Quad Core Cortex-A53 (ARM v8) 64-bit SoC @ 1.2GHz with 1 GB LPDDR-900 RAM

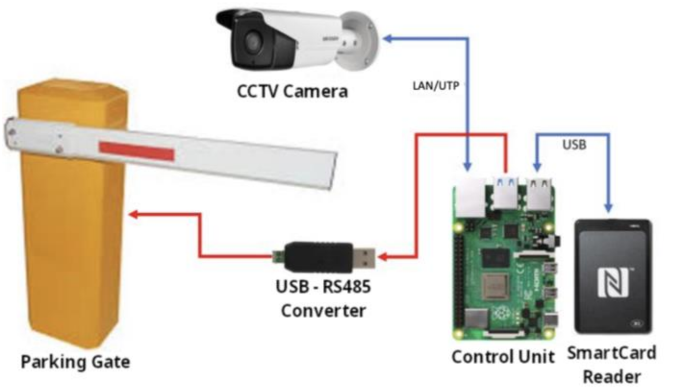

Figure 1. General schematic of the designed parking system.

- Raspberry Pi 4B uses Broadcom BCM2711 processor, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz with 1 GB LPDDR4-3200 RAM.
- Orange Pi Zero Plus uses Allwinner H5 processor, Quad-core Cortex-A53 (ARM v8) 64-bit SoC @ 1.4 GHz with 512 MB DDR3 RAM.

The three-unit controls used three microSD cards from different brands, namely Sandisk, with a maximum read speed of 98 MB/s, VGEN with a maximum read speed of 100MB/s and a maximum write speed of 48 MB/s, and Maestro, which has a maximum read speed of 52 MB/s and maximum write speed of 25 MB/s. Each microSD is a UHS-1 microSD with a speed class of 10 and a capacity of 16 GB. The class value of the microSD represents the minimum read and write speed that can be achieved on the microSD, with class 10 meaning that the microSD has a minimum speed of 10 MB/s [10]. The smart card reader used was the ACR-1252U. This device already supports the USB 2.0 full-speed interface. The smart card reader was connected serially via USB to the unit control [11]. The CCTV camera used was a CCTV camera supporting real-time streaming protocol (RTSP). The unit control can access this type of CCTV camera through a specific IP address [12], [13].

The parking portal was connected to the control unit using a USB-RS485 converter with the Modbus protocol, which is one of the extensively used communication protocols and an industry standard. This protocol is not hardware dependent as it only defines the information structure used for communication between devices [14], [15].

The workflow of the designed parking portal system is as follows:

- An access card, such as a faculty card, employee card, or student ID card, will be tapped on the smart card reader.
- The smart card reader will read the Unique Identifier (UID) number information on the access card and send it to the control unit.
- If the access card is successfully read, the control unit will store the UID number of the access card in the internal memory buffer on the control unit.
- Afterward, the CCTV camera will capture the license plate number of the entering vehicle.
- The control unit will then create a file with the file name of the captured photo in the format of date, time, and UID number used on the card.
- The properly formatted captured photo will be saved to the microSD memory attached to the control unit.

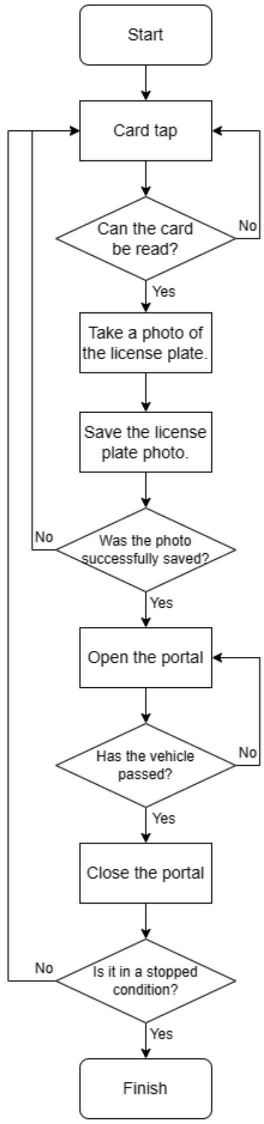

Figure 2. Flowchart of designed parking system.

- Once the photo is successfully saved, the control unit will send a command signal to the parking portal to open.
- The parking portal will then detect whether the vehicle has passed through the portal or not using loop sensors. If the vehicle has not passed through the portal, the portal will remain open. If the vehicle has passed through the portal, the parking portal will close and send a signal to the control unit to repeat the process, as mentioned in the first point.

To clarify the working process of the parking portal system, Figure 2 shows a flowchart of the workflow of how the parking portal system operates.

B. PERFORMANCE EVALUATION SYSTEM DESIGN

The focus of this research is to test and compare the performance of each control unit used. Therefore, not all parts of the parking portal system are implemented in the testing process of this research. In this study, testing was performed on specific parts of the parking portal system, from attaching the smart card to saving the photo results from the CCTV camera to the microSD memory installed on the control unit.

According to previous research, the process requiring relatively high computation on an embedded system unit control is usually the process of capturing and storing images from a CCTV camera [16]–[20]. It happens because processing

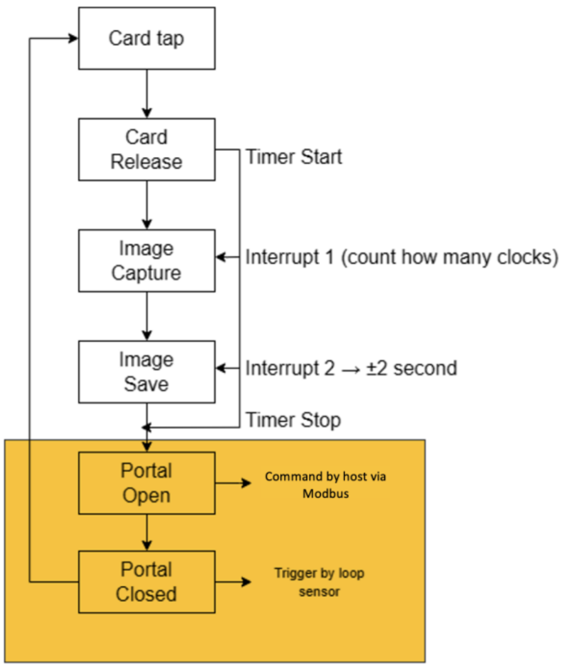

Figure 3. Performance evaluation process that only takes a certain moment (the part that is not shaded).

high-resolution image data requires significant computation for a unit control with limited computational resources.

However, the implementation of image capture and storage using embedded system devices is highly possible. Prior research has proven that the ARM microprocessor architecture is highly feasible for capturing and processing camera images [21]. This result is also supported by another research, which demonstrates that several single-board computers based on the ARM microprocessor architecture, such as Raspberry Pi and Orange Pi, can be used for image data processing and camera image capture [22].

Another reason for only testing the process up to storing the captured images in the internal memory of the control unit is due to the existence of external variables in the process of opening and closing the parking portal. One of the external variables is the difference in time required by the driver to leave the parking portal after the portal is opened. This variable causes the duration between opening and closing the portal to have different values in its implementation. Of course, if the entire system is tested, then the results of this study will be unbalanced. Figure 3 depicts the flowchart of the performance evaluation process used to clarify the workings of the performance evaluation system used.

To obtain a fair test result between the used control units, a performance evaluation process was conducted with the help of an externally mounted microcontroller as a computation time recorder [23]. The microcontroller must be connected to the control unit by connecting the general-purpose input/output (GPIO) pins on the control unit with the GPIO pins on the microcontroller. Subsequently, the microcontroller performed calculations based on the timer/counter. When the control unit started executing the task, the GPIO on the control unit triggered an interrupt on the microcontroller. When the control unit finished executing the task, the GPIO on the control unit triggered an interrupt on the microcontroller again. From this process, the microcontroller stored the calculated result value that had been running according to the existing interrupt triggers. The microcontroller was ESP32 because every available GPIO pin can be used as an interrupt pin [24]. Table I shows the pin mapping between the control unit and the microcontroller used for performance evaluation.

TABLE I
PIN MAPPING TO CONNECT THE CONTROL UNIT WITH MICROCONTROLLER

Pins on Control Unit		Pins on Microcontroller
Raspberry Pi 3B/4B	***Orange Pi Zero Plus***	***ESP32***
Pin 23	Pin 3	Pin 16
Pin 24	Pin 5	Pin 17
Pin 17	Pin 7	Pin 19

TABLE II
TESTING WITH SANDISK, VGEN, AND MAESTRO MICROSD

Test	Total Time (ms)			Capture Image Time (ms)			Save Image Time (ms)		
	S	***V***	***M***	***S***	***V***	***M***	***S***	***V***	***M***
1	1,919	1,835	2,036	1,871	1,786	1,912	48	49	124
2	1,942	1,828	2,066	1,828	1,778	1,953	114	50	113
3	1,934	1,945	1,983	1,869	1,895	1,871	65	50	112
4	1,934	2,016	2,050	1,882	1,904	1,937	52	112	113
5	1,925	1,938	1,980	1,877	1,889	1,867	48	49	113
6	1,942	2,022	2,095	1,849	1,972	1,981	93	50	114
7	1,849	2,039	1,930	1,800	1,925	1,817	49	114	113
8	1,946	1,988	2,101	1,841	1,939	1,987	105	49	114
9	1,813	1,870	1,953	1,764	1,821	1,842	49	49	111
10	1,964	1,966	2,034	1,847	1,882	1,919	117	84	115
Avg.	1,917	1,945	2,023	1,843	1,879	1,909	74	65.6	114.2

Note: S: Sandisk, V: VGEN, M: Maestro

III. TEST RESULTS

The testing in this study was conducted using several scenarios. The test used several embedded system devices: Raspberry Pi 3B, Raspberry Pi 4B, and Orange Pi Zero Plus. Furthermore, the test was also done using several different brands of microSDs, namely Sandisk, VGEN, and Maestro. The microSD used was a class 10 microSD since it has high speed and does not cause bottlenecks during testing. Class 10 microSD also supports video formats with resolutions up to 4K. Each microSD was tested on each embedded system device used.

In this test, the resolution of the image captured from the CCTV camera was 720p, commonly referred to as HD quality. Each test was conducted using the same CCTV camera. Ten data samples were collected in each test, and the average of the obtained values was calculated.

The operating system used on Raspberry Pi 3B and Raspberry Pi 4B was Raspbian Buster with Linux kernel version 5.10. Meanwhile, Orange Pi Zero Plus used the Armbian Buster operating system with Linux kernel version 5.15. Both operating systems were based on the Debian version 10 (code name Buster) that had been adapted to run smoothly on embedded systems with ARM-based processors. The programming language used was Python version 3. The following are the test results that have been conducted in several scenarios as previously described.

A. TESTING ON RASPBERRY PI 3B

This test was conducted using Raspberry Pi 3B hardware. Table II shows that the total time required to capture a photo and save it to the memory card was 1,917 ms, 1,945 ms, and 2,023 ms.

TABLE III
TESTING WITH SANDISK, VGEN, AND MAESTRO MICROSD

Test	Total Time (ms)			Capture Image Time (ms)			Save Image Time (ms)		
	S	***V***	***M***	***S***	***V***	***M***	***S***	***V***	***M***
1	2,045	1,882	2,068	1,961	1,801	1,986	84	81	82
2	2,064	1,847	1,954	1,982	1,765	1,872	82	82	82
3	2,062	1,812	1,985	1,976	1,730	1,902	86	82	83
4	2,006	1,847	2,068	1,922	1,760	1,986	84	87	82
5	2,060	1,778	2,048	1,976	1,745	1,959	84	33	89
6	2,020	1,744	2,099	1,934	1,663	2,016	86	81	83
7	2,046	1,869	2,076	1,963	1,788	1,993	83	81	83
8	2,050	1,836	2,036	1,966	1,754	1,930	84	82	106
9	2,025	1,829	2,065	1,939	1,747	1,981	86	82	84
10	1,941	1,835	2,047	1,858	1,754	1,964	83	81	83
Avg.	2,032	1,828	2,045	1,948	1,751	1,959	84.2	77.2	85.7

Note: S: Sandisk, V: VGEN, M: Maestro

TABLE IV
TESTING WITH SANDISK, VGEN, AND MAESTRO MICROSD

Test	Total Time (ms)			Capture Image Time (ms)			Save Image Time (ms)		
	S	***V***	***M***	***S***	***V***	***M***	***S***	***V***	***M***
1	2,165	2,203	2,119	2,124	2,162	2,079	41	41	40
2	2,564	2,632	1,971	2,523	2,589	1,931	41	43	40
3	2,312	2,405	1,997	2,272	2,365	1,957	40	40	40
4	2,610	2,418	2,068	2,567	2,377	2,028	43	41	40
5	2,235	2,489	1,927	2,195	2,446	1,887	40	43	40
6	2,652	2,429	2,113	2,609	2,386	2,073	43	43	40
7	2,455	2,150	1,913	2,415	2,110	1,873	40	40	40
8	2,348	2,254	1,946	2,308	2,213	1,906	40	41	40
9	2,531	2,079	2,071	2,491	2,038	2,031	40	41	40
10	2,498	2,520	2,047	2,456	2,477	2,007	42	43	40
Avg.	2,437	2,358	2,017	2,396	2,316	1,977	41	41.6	40

Note: S: Sandisk, V: VGEN, M: Maestro

B. TESTING ON RASPBERRY PI 4B

In this test, the Raspberry Pi 4B hardware was used. Table III shows that the total time required to capture and save the photo to the memory card was 2,032 ms, 1,828 ms, and 2,045 ms.

C. TESTING ON ORANGE PI ZERO PLUS

In this test, Orange Pi Zero Plus hardware was used. Table IV shows that the total time required to take a photo and save it to the memory card was 2,437 ms, 2,358 ms, and 2,017 ms.

From the experiment results, a comparison graph can be created between the tested devices, namely using different embedded system devices with the same microSD and using the same embedded system device with different microSD. Figure 4 shows the results of the first test, while Figure 5 shows the results of the second test.

The graph shown in Figure 4 shows that the Raspberry Pi 3B device had a relatively faster average time to capture an image compared to the Raspberry Pi 4B and Orange Pi Zero Plus devices. However, in the case of storing images on a microSD, the Orange Pi Zero Plus device had a relatively faster average time compared to the Raspberry Pi 4B and Raspberry Pi 3B. The graph also shows that the Raspberry Pi 4B device had an average speed of capturing and storing images that were between the Raspberry Pi 3B and Orange Pi Zero Plus devices.

The condition shown in the graph in Figure 4 may be caused by the differences in specifications of the embedded systems as

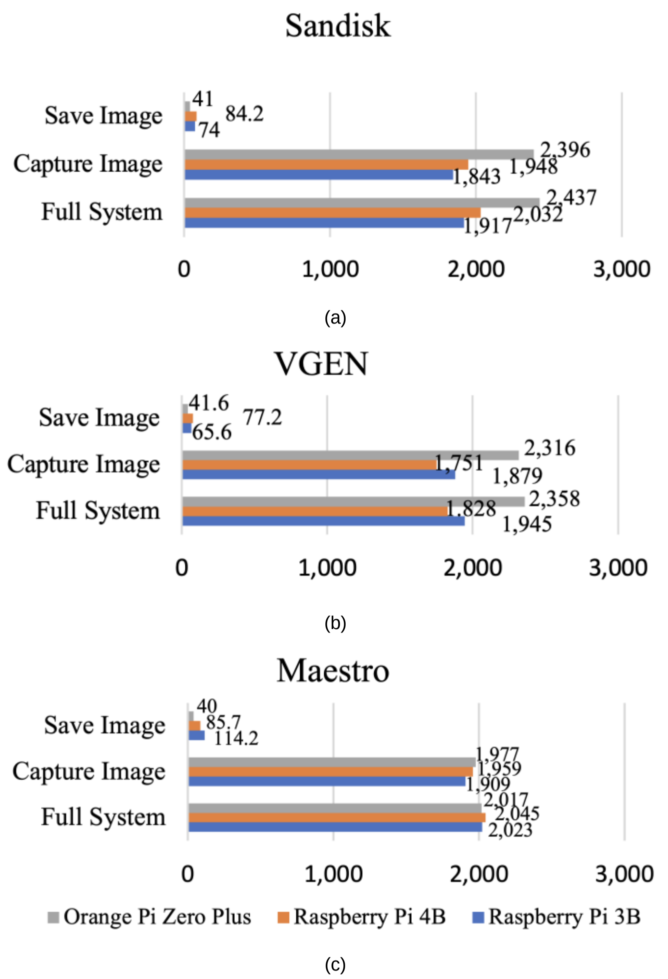

Figure 4. Comparison between each Raspberry Pi 3B, 4B, and Orange Pi Zero Plus with (a) Sandisk, (b) VGEN, and (c) Maestro microSD.

well as the operating systems used. Orange Pi Zero Plus has a computation time that is almost always the longest when seen from the combined computation time of the image capture and image storage processes. It may be due to the processor that has the lowest base frequency compared to the other two embedded systems, which is 400 MHz, although it has a turbo frequency of up to 1.4 GHz. In addition, this device also has relatively small RAM, which is 512MB. Orange Pi Zero Plus also uses the Armbian operating system, which Orange Pi developers do not officially support. Consequently, some processes do not work optimally.

Raspberry Pi 4B is recorded to have the fastest computation time in the combined image capture and storage process. It could be due to its better device specifications. Raspberry Pi 4B used the latest processor released in the second quarter of 2019, compared to the Orange Pi Zero Plus, whose processor was marketed in the fourth quarter of 2017, and Raspberry Pi 3B, which was marketed in the first quarter of 2016. The processor has the best performance as it has the highest base frequency, which is 1.5 GHz. In addition, Raspberry Pi 4B is also supported by the largest RAM size and LPDDR4 type with a clock of 3200MHz, the highest compared to the other two embedded systems. This device also uses the Raspbian operating system, which is officially supported by Raspberry developers, thus having better support and compatibility.

The graph depicted in Figure 5 shows that using different microSD cards had relatively similar speeds, depending on the embedded system device used. However, what is interesting from this experiment is using Raspberry Pi 4B with VGEN

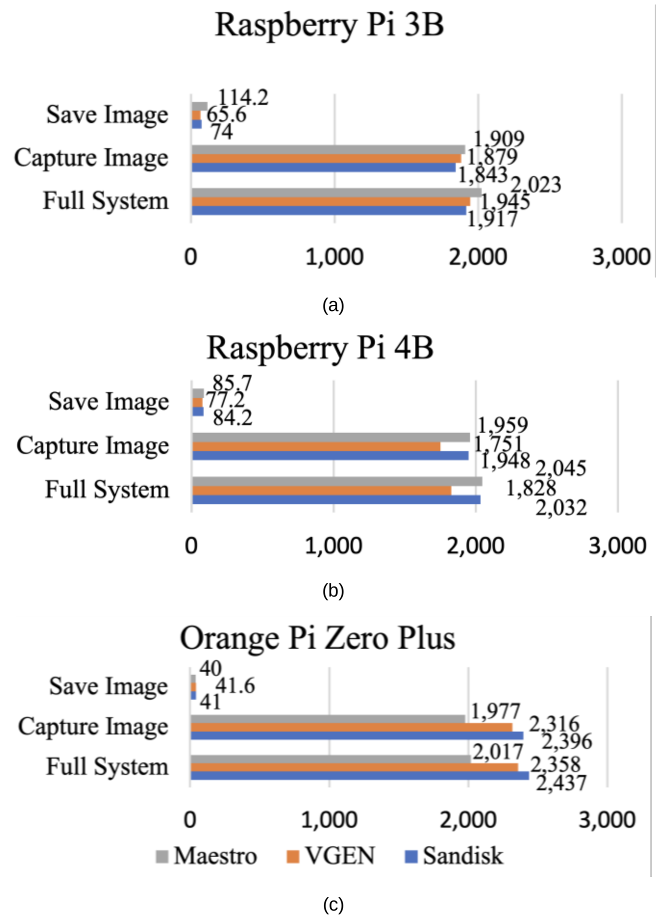

Figure 5. Comparison among Sandisk, VGEN, and Maestro microSD on (a) Raspberry Pi 3B, (b) Raspberry Pi 4B, and (c) Orange Pi Zero Plus.

TABLE V
COMPARISON OF AVERAGE TIME FOR IMAGE CAPTURE, IMAGE STORAGE, AND POWER CONSUMPTION IN SEVERAL HARDWARE PLATFORMS WITH DIFFERENT MEMORY TYPES

Hardware Platform	Memory	Computation (ms)			Power Consumption (W)
		Capture Image	*Save Image*	*Total Average*	
Raspberry Pi 3B	Sandisk	1,842.8	74.0	1,916.8	3.9
	VGEN	1,879.1	65.6	1,944.7	
	Maestro	1,908.6	114.2	2,022.8	
Raspberry Pi 4B	Sandisk	1,947.7	84.2	2,031.9	2.9
	VGEN	1,750.7	77.2	1,827.9	
	Maestro	1,958.9	85.7	2,044.6	
Orange Pi Zero Plus	Sandisk	2,396.0	41.0	2,437.0	1.9
	VGEN	2,316.3	41.6	2,357.9	
	Maestro	1,977.2	40.0	2,017.2	

microSD. In the graph, if the image capture time and image storage time on the microSD are calculated, then the Raspberry Pi 4B device has the fastest average combined time compared to the other two control units. The summary of the results of this experiment can also be seen in Table V.

The graph in Figure 5 also demonstrates that the same microSD test results can yield different speeds when used on different embedded systems. It can be caused by differences in the level of compatibility of the embedded system with the microSD used. Of the three tested microSD cards, the Maestro had the lowest read and write speeds. Although the microSD cards have different maximum read and write speed specifications, this test showed insignificant differences in speed. This condition is caused by the embedded system not reaching maximum speed when writing to the microSD while saving the image. The microSD inserted into the embedded system was also used to load the operating system, affecting the speed of saving the image because some of the microSD resources were used for the operating system.

The Orange Pi Zero Plus device appears to have the fastest image storage time compared to the other two embedded systems. This condition could be caused by the Armbian operating system used, which is relatively lighter compared to the Raspbian operating system used by the Raspberry Pi 3B and Raspberry Pi 4B devices.

This study also collected data on the energy consumption of each embedded system device used. The energy consumption data were obtained from the average energy used for each image capture and storage computation on the microSD. Energy consumption data were measured using a KWh meter whose values were collected during the system's image capture and storage process on the memory card. Table V also shows a comparison of the energy consumption of each embedded system device used.

IV. CONCLUSION

Based on the results of the tests conducted, it can be concluded that Raspberry Pi 4B combined with VGEN memory card had the fastest computing ability in the process of capturing and storing images compared to Raspberry Pi 3B and Orange Pi Zero Plus. In terms of power consumption, Orange Pi Zero Plus had the lowest power consumption in the process of capturing and storing images compared to Raspberry Pi 3B and Raspberry Pi 4B. Meanwhile, the type of memory card used does not significantly impact the speed of image capture and storage.

CONFLICT OF INTEREST

The authors declare that the article entitled "Evaluation of Embedded System Platforms for Automatic Parking System Control Unit" is written free of any conflict of interest.

AUTHOR CONTRIBUTION

Conceptualization, Wahyu Dewanto; writing—original draft preparation, Agung Fathurrahman; writing—review and editing, Agus Bejo.

REFERENCES

[1] N. Jesani *et al.*, "Smart Card for Various Application in Institution," *2020 IEEE Int. Stud. Conf. Elect. Electron., Comput. Sci. (SCEECS)*, 2020, pp. 1–5, doi: 10.1109/SCEECS48394.2020.26.

[2] R. Bankar *et al.*, "A Review on IoT Based Smart Card System for Students," *2020 4th Int. Conf. Invent. Syst., Control (ICISC)*, 2020, pp. 1–3, doi: 10.1109/ICISC47916.2020.9171219.

[3] H. Taherdoost and M. Masrom, "An Examination of Smart Card Technology Acceptance Using Adoption Model," *Proc. ITI 2009 31st Int. Conf. Inf. Technol. Interfaces*, 2009, pp. 329–334, doi: 10.1109/ITI.2009.5196103.

[4] A. Bejo, M.F. Hamzah, and A. Suwastono, "Perancangan Smart Card Reader Menggunakan STM32F4 Discovery Kit," *J. Nas. Tek. Elekt., Teknol. Inf.*, Vol. 6, No. 3, pp. 342–351, Aug. 2017, doi: 10.22146/jnteti.v6i3.337.

[5] A. Suryanto *et al.*, "Optimalisasi Keluaran Panel Surya Menggunakan Solar Tracker Berbasis Kamera Terintegrasi Raspberry Pi," *J. Nas. Tek. Elekt., Teknol. Inf.*, Vol. 10, No. 3, pp. 282–290, Aug. 2021, doi: 10.22146/jnteti.v10i3.1142.

[6] M.R.A. Cahyono, I. Mariza, and Wirawan, "Sistem Pemantauan dan Pengendalian Sepeda Listrik Berbasis Internet of Things," *J. Nas. Tek. Elekt., Teknol. Inf.*, Vol. 11, No. 1, pp. 53–60, Feb. 2022, doi: 10.22146/jnteti.v11i1.3183.

[7] K. Kasym *et al*., "Parking Gate Control Based on Mobile Application," *2018 Jt. 7th Int. Conf. Inform. Electron., Vis. (ICIEV), 2018 2nd Int. Conf. Imag. Vis., Pattern Recognit. (icIVPR)*, 2018, pp. 399–403, doi: 10.1109/ICIEV.2018.8640954.

[8] R. Yasirandi, Y.A. Setyoko, and P. Sukarno, "Security Document for Smart Parking Gate based on Common Criteria Framework," *2019 7th Int. Conf. Inf., Commun. Technol. (ICoICT)*, 2019, pp. 1–8, doi: 10.1109/ICoICT.2019.8835234.

[9] C.C. How *et al.*, "Smart Parking Reservation Mobile Application*," 2022 IEEE Int. Conf. Distrib. Comput., Elect. Circuits, Electron. (ICDCECE)*, 2022, pp. 1–5. doi: 10.1109/ICDCECE53908.2022.9792684.

[10] N.O. Nwazor, "A Raspberry Pi Based Speaker Recognition System for Access Control," *2019 Int. Res. J. Eng., Technol. (IRJET)*, Vol. 6, No. 3, Mar. 2019, pp. 7412–7419.

[11] A. Bejo, R. Winata, and S.S. Kusumawardani, "Prototyping of Class-Attendance System Using Mifare 1K Smart Card and Raspberry Pi 3," *2018 Int. Symp. Electron., Smart Devices (ISESD)*, 2018, pp. 1–5. doi: 10.1109/ISESD.2018.8605442.

[12] B.N. Rao and R. Sudheer, "Surveillance Camera using IoT and Raspberry Pi," *2020 2nd Int. Conf. Invent. Res. Comput. Appl. (ICIRCA)*, 2020, pp. 1172–1176, doi: 10.1109/ICIRCA48905.2020.9182983.

[13] A.H.H. Basri, S.N. Ibrahim, N.A. Malik, and A.L. Asnawi, "Integrated Surveillance System with Mobile Application," *2018 7th Int. Conf. Comput., Commun. Eng. (ICCCE)*, 2018, pp. 218–222, doi: 10.1109/ICCCE.2018.8539244.

[14] Z. Jiang, Z. Si, and C. Luo, "Design & Implementation to an RFID Based Conference Management System," *2014 10th Int. Conf. Comput. Intell., Secur.*, 2014, pp. 143–147, doi: 10.1109/CIS.2014.29.

[15] W. You and H. Ge, "Design and Implementation of Modbus Protocol for Intelligent Building Security," *2019 IEEE 19th Int. Conf. Commun. Technol. (ICCT)*, 2019, pp. 420–423, doi: 10.1109/ICCT46805.2019.8946996.

[16] H. Wang, X. You, and R. Wang, "Design of Image Capture and Transmission Embedded System for Remote Monitoring," *2012 8th Int. Conf. Inf. Sci., Digit. Content Technol. (ICIDT2012)*, 2012, pp. 661–664.

[17] K. Loukil *et al.*, "Design and Test of Smart IP-Camera Within Reconfigurable Platform," *2017 2nd Int. Conf. Anti-Cyber Crimes (ICACC)*, 2017, pp. 25–29, doi: 10.1109/Anti-Cybercrime.2017.7905257.

[18] L.M. Fawzi, S.Y. Ameen, S.M. Alqaraawi, and S.A. Dawwd, "Embedded Real-Time Video Surveillance System Based on Multi-Sensor and Visual Tracking," *Appl. Math., Inf. Sci*, Vol. 12, No. 2, pp. 345–359, Mar. 2018, doi: 10.18576/amis/120209.

[19] T.A. Mounir *et al.*, "Performance Evaluation of Basic Image Processing Algorithms in CPU, GPU, Raspberry Pi and FPGA," *Int. J. Comput. Sci. Eng. (IJCSE)*, Vol. 9, No. 4, pp. 312–325, Jul.–Aug. 2020.

[20] E. Harwood, *Digital CCTV: A Security Professional's Guide*. Amsterdam, Netherlands: Butterworth-Heinemann, 2007.

[21] G. Yang and K. Shen, "ARM9 Embedded System of the Image Acquisition and Processing," *2010 Int. Conf. Anti-Counterfeiting Secur., Identif.*, 2010, pp. 138–141, doi: 10.1109/ICASID.2010.5551517.

[22] A. Mishra and A. Dixit, "Embedded Image Capturing & Digital Converting Process Using Raspberry Pi System Interfacing and Comparision of Generation 2 Verses Generation 1 Models in Raspberry Pi," *Int. J. Comput. Sci., Inf. Technol. (IJCSIT)*, Vol. 6, No. 2, pp. 1798–1801, Mar.–Apr. 2015.

[23] V.B. Vales *et al.*, "Fine Time Measurement for the Internet of Things: A Practical Approach Using ESP32," *IEEE Internet Things J.*, Vol. 9, No. 19, pp. 18305–18318, Oct. 2022, doi: 10.1109/JIOT.2022.3158701.

[24] Espressif Systems, "ESP32 Series Datasheet," ESP32 Series Datasheet v4.3, 2023.