International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V71I3P235

Volume 71 Issue 3, 337-344, March 2023

Original Article

Interexchange RFID-RC522 Sensor in Machine-to-Machine Communication Between Two Arduinos using Serial JSON Schema

Yulianto[1], Sidharta[2], Nyoman Wira Prasetya[3], Muhammad Aldiki Febriantono[4]

[1,2,3,4]*Computer Science Department, School of Computer Science, Bina Nusantara University, Indonesia*.

[1]*Corresponding Author : vulianto003@binus.ac.id*

Received: 15 December 2022 Revised: 13 March 2023 Accepted: 22 March 2023 Published: 25 March 2023

Abstract *- The internet of things (IoT) concept has become a general requirement that makes the device work automatically and is controlled and monitored by the user anywhere they are. IoT works by connecting sensors, edge devices, cloud servers, and clients through the machine-to-machine protocol. The sensor has the task of reading the life environment condition to be digital data and transferring the data to the microcontroller or other edge devices. One of the kind sensors popularly used for reading the authentication is the RFID sensor. The RFID-RC522 is a type of sensor device that is open source. Its sensor cannot be directly connected to computers; it needs a microcontroller because the connection is a serial protocol interface. After pre-processing the sensor data from the microcontroller, the output data can be exchanged with other microcontroller devices. The communication between one Arduino to another Arduino for data exchange is known as machine-to-machine communication (m2m). The m2m communication can be realized through a wireless medium or wiring cable. There are huge of research that discusses about m2m in the wireless medium. This study proposes the model of m2m communication with the wiring cable connection to exchange RFID data. The proposed model involves 5 main components. There is RFID-RC522 as a sensor identification card, two Arduino modules connected in m2m schema, and a real-time clock module for recording the identity card when tapping process. The Serial JSON protocol is also involved in wrapping the RFID data to minimize the complexity of parsing data between two Arduino.*

Keywords *- RFID, RC522, RFID-RC522, Arduino, Machine-to-machine, Arduino JSON.*

1. Introduction

IoT, known as the Internet of Things, is a concept that can be used for reading real environments through sensor devices [1][2]. The sensor device is designed to convert the environment condition being digital data. The environment can come from the lighting, sound, humidity level, temperature, etc. [3]. The digital data, which resulted from converting the reading sensor activity, was then transferred to the microcontroller. Wireless or wiring cable connections can transfer data from a sensor to the microcontroller. The communication between microcontrollers, or in other words known as edge devices, can be connected to other edge devices or directly to the cloud for the exchange of data and instruction, known as terms of machine-to-machine (M2M) communication [4].

Furthermore, the microcontroller converts the incoming digital sensor data and arranges how the sensor data will be sent to another device, cloud server, etc. The microcontroller also has the task of controlling how the incoming instruction or any responses will be executed. Because the IoT can be connected to the cloud internet, those IoT technologies make it possible for users in other places can monitor through their gadgets about target object's condition, which is equipped with IoT. One of the sensor types that can be used for authentication is the RFID-RC522 sensor. The RFID-RC522 sensor is commonly used for reading the passive identification card.

Several research has used this sensor for the utilization of smart classrooms [5], attendance systems [6][7], tracking shipment for data inventory in the warehouse [8], and inventory of biomedical equipment [9]. Otherwise, there were also several research that did not utilize the RFID sensor yet for security or identification purposes; for example, the research about adaptive learning models for classrooms [10][11] focussed on increasing the increased enthusiasm for learning. Generally, several IoT researchers use Arduino UNO in their experiments because this microcontroller has 14 pins of digital I/O (Input-Output). However, in reality, only 11 pins can be used for general programming purposes because pins 0 and 1 on Arduino

UNO are used for serial programming. So pin numbers 2 to 13 can be utilized.

Meanwhile, suppose the Arduino UNO is combined with the RFID-RC522 sensor. In that case, the Arduino UNO will only leave 5 pins of digital I/O. to overcome the limitations of Arduino UNO pins. This research aims to combine the two Arduino UNO to separate the pre-processing control of RFID-RC522 so that the second Arduino UNO still have 9 available I/O digital pins.

More specifically, this research proposes the prototype of a combination RFID-RC522 module that involves two Arduino UNO modules. Where on the first Arduino works to process the incoming data from the RFID readings sensor. The first Arduino also acted as a data sender. The RFID data is then transferred to the second Arduino. The second Arduino acted as a receiver. In this study, the second Arduino is represented as other edge devices. The first Arduino UNO connected to the second Arduino UNO using a pair of serial cables. Before the data from the first Arduino send to Second Arduino, the serial JavaScript Object Notation (JSON) or Arduino JSON library to wrap the data implemented. The wrapping of RFID data to JSON is for a reason to minimize the complexity of parsing data programs that exist both on the first and second Arduino. The serial JSON library also can prevent corrupt data that may be occurred during the interexchange of data. The result of several experiments, the RFID data transferred from the first Arduino to the second Arduino through the proposed method, can be available to send the data from the first Arduino to Second Arduino UNO in M2M schema with a score delay of around 264 to 266 milliseconds.

2. Literature Review

M2M (machine-to-machine) communication is a concept of making the connection between one edge device to another device with the same level, or it can also behave as client and server communication, especially in an autonomous environment [12][13]. The edge device can be the microcontroller device [14][15]. Unlike computers with enormous power computation resources and flexibility to upgrade every part of the component, the microcontroller is basically just a module composed of a single integrated circuit (IC) where the inside IC includes an arithmetic logic unit (ALU), random access memory (RAM), and also non-volatile memory in the form of internal EEPROM where all components designed into one solid packed [16]. One of the microcontroller devices that many researchers have used to support their experiments is Arduino UNO [17]–[19].

Machine-to-machine (M2M) communication has been successfully implemented in many fields to support intermediary communication systems, i.e., intelligent home automation, intelligent medical instruments [9], industrial [8], etc. The M2M is used to make the local connection autonomous and can also be implemented for telematics and telemetry, where the control communication can reach a long distance [20]. The research that talks about optimizing M2M communication, especially in the wireless area, has been proposed by [21] and [22]. M2M is used to gather several data from the sensor and then send it to another device for further processing [23]. Some research about machine-to-machine communication was mostly discussed in the wireless area. For example, the author [24][25][26] talked about how to optimize the power consumption for edge devices in M2M wireless medium. Estimated the node connected in the M2M schema and the wireless field [18]. Wireless M2M communication schema has also been discussed to support temperature monitoring systems [28][25].

The realization of M2M to connect one thing to another thing for data communication has disadvantages, especially in security, where the attacker can easily monitor the network and discover the device type [29]. To improve the security issue, the research conducted by Lokhande [20] proposed an EDDC device that can detect and classify a device with authority. The improving security of M2M has also been conducted by [30].

The utilization of a flashlight on smartphones to send data through lighting media by wireless has been proposed [1]. The id data of smartphone authentication is converted into 72 bytes with a pulse width of 27 milliseconds. The 72 bytes of data were further transmitted through a flashlight that existed on the smartphone. The lighting emitted by the flashlight on the smartphone is then captured with a photodiode sensor. The data captured by the photodiode is then delivered to the edge devices. On the edge device, the FPGA module was used. The total time to send the whole data from the smartphone to FPGA was 1.944 milliseconds.

The research conducted by [6] makes an attendance system based on RFID. He proposed an RFID reader by utilizing an Arduino module to parse incoming RFID data from the RC522 module. The data is then interfaced with a computer through a USB cable. The utilization of an RFID module was also conducted to track the shipment in the warehouse, where the RFID reader module was hooked up to Raspberry pi [8]. Every packet was given a passive RFID card, so every packet's movement in the warehouse could be tracked easily [8]. The result of reading and parsing RFID card data from the Raspberry pi is then transferred to a cloud server. Before the data is transferred to the cloud server, the data is first encapsulated using the MQTT standard protocol. Several protocols can be used for M2M communication, especially in the wireless medium, i.e., SECS/GEM, DDS, OPC UA, and MQTT [31].

The other standard method for wrapping the amount of data is using a JavaScript Object Notation (JSON) format. JSON has been to be a standard schema that is used to wrap data that will be transferred between client and server or vice versa through the Application Programming Interface (API) [32][33]. The JSON data can be readable not only by machines but also by humans easily [34]. Inside the JSON contains the structure of keys and values [35]. The keys refer to the variable's name, and values refer to the information of data which can be in the form of numbers and strings [36][37].

Inspired by research conducted by Laxmi [8] that proposed a combination of an RFID reader with the raspberry pi module; this research presents the combination of the RFID module with a combination of two Arduino UNO in the wire-able of machine-to-machine schema. The proposed framework is also equipped with an RTC module to synchronize a time stamp.

3. Proposed Method

RC522 is known for identifying wireless sensor readers specially designed and distributed as open-source devices, inspired by the research done by [18] that makes a connection between RFID to Arduino loli, where the Arduino loli implemented by [18] has a role as an edge device that processes incoming RFID data and then transmits the data to the cloud by wireless communication. This study aims to propose an RFID-RC522 reader connected to machine-to-machine communication, which can be seen clearly in Figure 1. Thanks to wokwi of web IoT emulators that provide online emulator services, the researcher can take advantage of drawing the sample description based on the figure be easy, which can be seen in Figure 1. This study the specific research to support the part of security identity by taking advantage of RFID, especially for the RFID-RC522 module as the RFID card reader sensor conducted. The RFID module is then connected to Arduino UNO through a serial protocol interface (SPI) wire cable to deliver the RFID data. The incoming string data from RFID is then wrapped into JSON formatted and then sent to other Arduino UNO through Serial wire Communication.

A detailed description of the proposed model can be seen in Figure 1, involved two Arduino UNO modules connected using three cables acting as serial communication. The three cables are also composed of three different colour cables. The colour black cable is for the ground (GND) connection, and the other cables, pink and green, are for the transmitter cable (Tx) and receiver (Rx) cables. The first Arduino act as a data sender. And the second Arduino act as a receiver. The two Arduino mentioned before can be said to communicate with each other in the m2m schema.

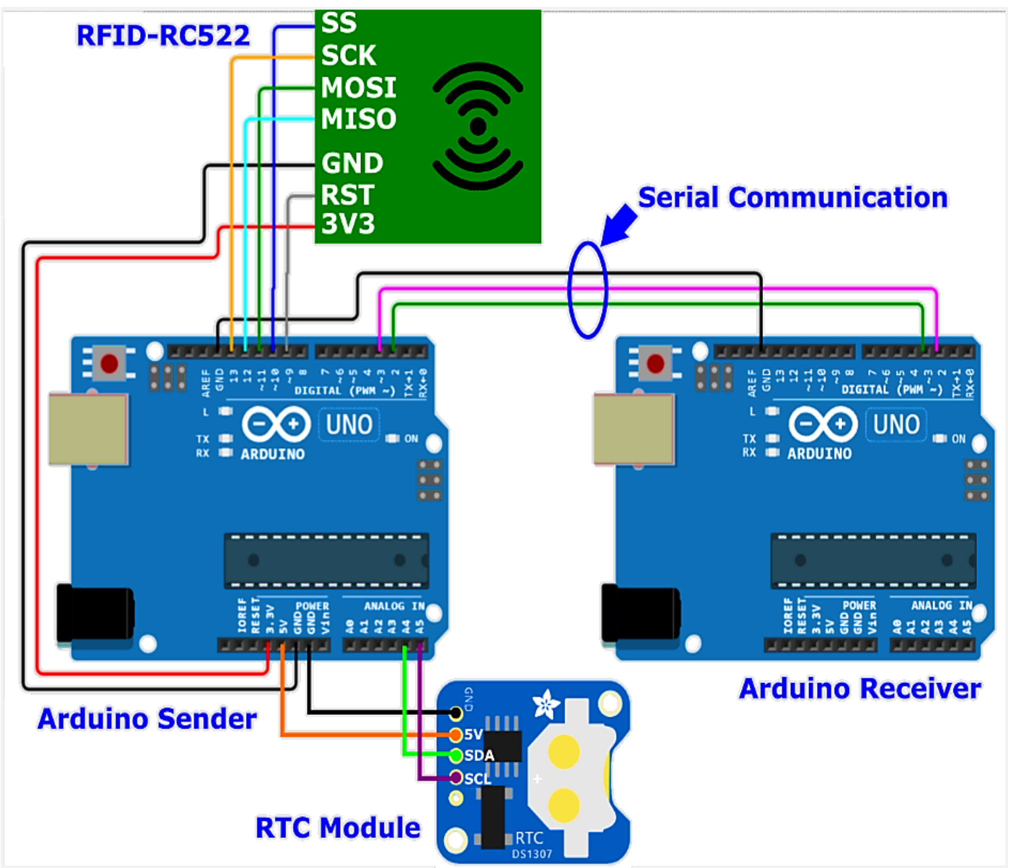

Fig. 1 Proposed model of RFID in machine-to-machine communication interconnected by a serial cable.

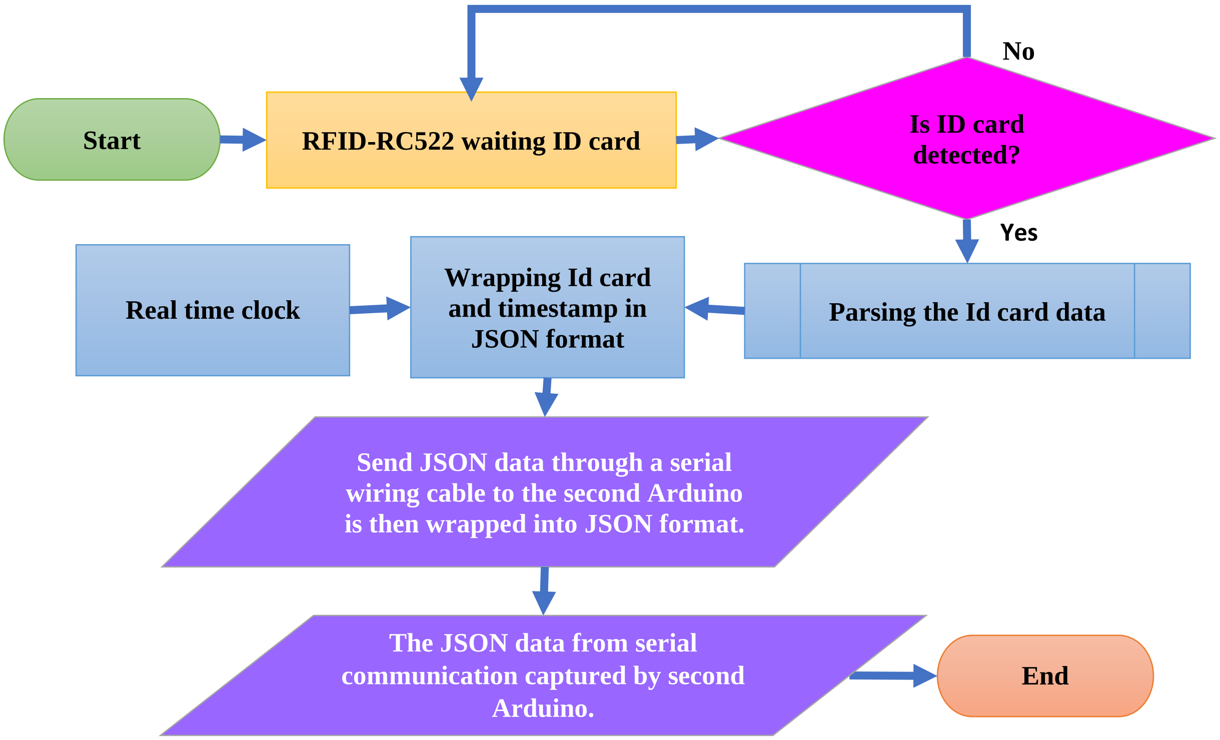

Fig. 2 Flowchart of the proposed method

By default, the Arduino UNO has no internal real-time clock; this study also equipped the real-time clock with the series of Tiny RTC DS1307. This RTC was equipped with a Lithium cell battery with a series CR2032 with a voltage consumption is 3-volt. The battery is to keep the internal clock in the RTC module still working and calibrated. The RTC module will continuously send the data every second in the form of date and time in an hour, second, and minute resolution to the Arduino through the wiring connection mentioned in Table 2. In this study, the Arduino just catches the important time from RTC when the incoming RFID data is detected—the details of how the framework works can be seen in Figure 2.

Figure 2. talks about how the proposed method algorithms flow works overall. Starting from RFID-RC522 by continuously waiting for the incoming Id card. If the Id card is detected, the data is transferred to the Arduino sender to parse the Id card data. The Arduino sender module also catches the timestamp from the real-time clock module (Tiny RTC DS1307). Id Card data and timestamp wrapped in JSON format together using ArduinoJson library [39]. The JSON data is then transferred to the Arduino receiver module or the second Arduino module through wiring serial communications. Furthermore, the details of the wiring connection between the two modules are also displayed in Table 1, Table 2, and Table 3.

Table 1 describes the interconnection between the RFID-RC522 module with the first Arduino. For the remainder, the first Arduino UNO act as the sender of data. The RFID-RC522 module has several pins, i.e., SS, SCK, MOSI, MISO, GND, RST, and 3v3. SS pin acts as a trigger signal when the Arduino sender works to enable the SPI mode interfaces.

Table 1. Pin wiring of RFID-RC522 to sender module Arduino.

	RFID-RC522 Sensor Module	Arduino Sender Module
Pin Connection	SS	Digital pin 10
	SCK (Serial Clock)	Digital pin 13
	MOSI (Master Out Slave In)	Digital pin 11
	MISO (Master In Slave Out)	Digital pin 12
	GND (Ground)	GND
	RST (Reset)	Digital pin 9
	3V3	3V3

The SS pin of RFID-RC522 is connected to digital pin 10 on the Arduino sender. SCK pin acts as an acceptor of the clock pulse that gets incoming data from the Arduino on the digital pin 13. MOSI is the input of SPI that is transferred from Arduino on digital pin 11. MISO is the opposite of the MOSI pin. MISO acts as an output SPI that is transferred from the RFID module to Arduino connected on digital pin 12. The RFID-RC522 also needs an input power for activating. The power supply for the RFID module is 3,3 volts of direct current (DC). To get the 3,3-volt RFID power supply connected to the 3v3 power pin, which is available on the Arduino pins. The GND pin between RFID and Arduino was also connected too. The RST pins on the RFID module act as a reset condition. If the Arduino goes low conditions from digital pin 9, then the RFID system module will be reset.

Table 2. Pin wiring of RTC module to Arduino sender module.

Pin Connection	RTC Module	Arduino Sender Module
	5V	5V
	SDA (Serial Data)	Analog pin A4
	SCL (Serial Clock)	Analog pin A5
	GND	GND

Table 2 describes the pair of wiring connections between the RTC and Arduino sender modules. There are 4 pins on the RTC module that needs to be connected to the Arduino sender module. First SDA pin connected to Analog pin A4 on Arduino. The SCL pin on the RTC module connected to Analog pin A5 on the Arduino sender. The RTC module needs input voltages of amount 5-volt direct current (DC) and GND connected to GND Arduino. The power supply of 5-volt and GND in a circuit can be used together as long as the minimum current is 1 Ampere fulfilled. The minimum current input is to prevent the effect of decreasing voltage when a load of task computing increases.

Table 3. Pin wiring of Arduino sender to Arduino receiver module.

Pin Connection	Arduino Sender Module	Arduino Receiver Module
	GND	GND
	Digital pin 3 (the pin acts as Tx)	Digital pin 2 (the pin acts as Rx)
	Digital pin 2 (the pin acts as Rx)	Digital pin 3 (the pin acts as Tx)
	5V	5V

As shown in Table 3, it is the final stage of the proposed framework. Both Arduino modules are connected by using serial wiring communication. The digital pin 3 on the Arduino sender acts as serial transmitter data (Tx) connected to digital pin 2 on the Arduino receiver. The digital pin 2 on the Arduino receiver acts as a receiver of serial data (Rx). Furthermore, the digital pin number 2 from the Arduino sender is connected to the digital pin number 3 on the Arduino receiver. The power supply that needs for both Arduino is also connected together for 5-volt and GND.

4. Experiment

This section explains step by step, starting from cable installation, program preparation, and testing. As shown in Figure 3, All modules connected with the wiring that soldered. The 5-volt power supply was obtained by connecting the Arduino receiver module to the laptop using an extended USB cable. For the first experiment, we focus on reading the data from RFID and RTC modules. The RFID module needs to install the MFRC522 library to make the Arduino available to access the RFID sensor module. The Arduino also needs to install the RTClib to access the external real-time clock module.

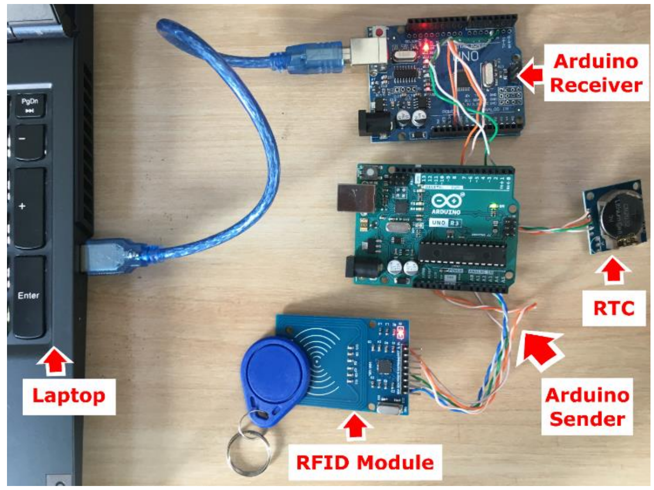

Fig. 3 Implementation of RFID module to Arduino sender connected to Arduino Receiver.

The data is transferred from the Arduino sender to the Arduino receiver through a custom serial configuration. The additional libraries of SoftwareSerial.h to define the new serial Tx and Rx pins are used. The baud rate for custom serial communication between Arduino Sender to Arduino Receiver sets to 9600. Both Arduino modules must be set to the same baud rate to make the serial communication. The default baud rate of serial communication for monitoring Arduino is 115200. The RTClib.h library is also needed to allow the Arduino UNO to access the external RTC module. When the debugging process is targeted to the Arduino sender, the date and time data in the RTC module will automatically be synchronized with the laptop.

5. Result and Discussion

This section discusses experimental results. In this experiment, an amount of 10 id cards for the simulation are used. There are Cards number 1, 2, 3 and up to card 10. The simulation separated to be two different parts. The first one is to measure the average time delay that happens when the Arduino sender receives data from the id card.

When the RFID card is tapped into the RFID sensor, the RFID sensor detects the incoming ID card data. Then it sends the data to the first Arduino or Arduino sender module through a serial interface protocol (SPI) represented by SS, SCK, MOSI, MISO, and RST pins, respectively, can be seen in Table 1. The RFID data is represented as hex code obtained from a Byte array. The class of MFRC522 from MFRC522.h Arduino library will give a 4-byte array starting from an index number of 0 until 3, which contains hex code representing ID card data.

In the experiment, when the RFID card is tapped into the RFID sensor, the processing time (Δt) to get RFID data in one cycle of running the Arduino program will be measured using Equation 1. The time measurement is in the milliseconds' range, where t0 is a starting time located in the first line of the main code in Arduino, and the t1 refers to the last time after the number index 3 of the hex code byte array was accessed.

$$\Delta t = t1 - t0 \quad (1)$$

The result for the first stage can be seen in Table 4. According to Table 4, the process of reading an RFID sensor, parsing the id card data, and wrapping it to JSON, got a total delay variation with a minimum value is 131 milliseconds (mS) and a maximum time delay is 134 mS. The time delay minimum is 131 mS occurred on Card 6 and card number 9. The maximum time delay occurred on Card numbers 4, 5 and 6.

The data displayed in Table 5. are the results of time delay measurement from reading the RFID sensor to the Arduino receiver module—those data obtained from the experimental reading of the id card. The sensor will capture the id card, which is tapped to RFID-RC522, and the digital data will be delivered to the Arduino transmitter. The Arduino transmitter then forwarded output data in JSON form to the Arduino receiver module. The Arduino receiver module was then monitored using a laptop and got a time delay between 140 to 141 Ms.

During the experiments, the interesting fact explored found that RTC DS1307 has disadvantages. Where the RTC cannot generate specific time resolution in millisecond units. So to measure how much time spends by Arduino for 1 cycle instruction, we involve millis() syntax programs. To get information that is displayed in Table 4 and Table 5, related to how long time in milliseconds spent.

Table 6 shows the difference between the proposed framework from this study with other research that has been done. The research conducted by Laxmi [8] uses a combination of RFID as sensor identification connected with Raspberry Pi as an edge device. Then the edge device sends the whole of the data to the server by wireless through Wi-Fi. The whole of the data that is transferred first is wrapped together using the MQTT protocol. However, the research conducted by Laxmi [8] does not clearly describe how long time was spent in every node until the data reached the main server. Meanwhile, compared with this proposed method, the research that utilises a smartphone's flashlight [1] gives a satisfactory result where the whole time spent is 141 milliseconds.

6. Conclusion

In this study, the proposed prototype RFID sensor that is integrated into machine-to-machine communication schema has been presented in detail in the paragraph. The type of RFID sensor which discussed is RFID-RC522. The sensor is then connected to the Arduino sender module. This Arduino has the task of analyzing the incoming data from the RFID sensor when the identity card is detected. When the RFID sensor detects the id card, the Arduino sender module will catch the identification card data and record the time stamp that the RTC module transfers. The identification card data and time stamps are then wrapped in JSON schema.

Table 4. Response time of reading id card to Arduino sender module.

Card name	Id card	RFID to Arduino Sender
Card 1	ba-a6-70-15	132 – 133 ms
Card 2	be-a3-dc-22	132 – 133 ms
Card 3	dc-72-5d-39	132 – 133 ms
Card 4	dc-6c-e6-39	132 – 134 ms
Card 5	dc-f5-26-39	132 – 134 ms
Card 6	29-37-71-59	131 – 134 ms
Card 7	dc-97-d0-39	132 – 133 ms
Card 8	ec-3b-39-39	132 – 133 ms
Card 9	ec-34-46-39	131 – 132 ms
Card 10	dc-f4-9d-39	132 – 133 ms

Table 5. Response time of reading id card data to Arduino receiver module.

Card name	Id card	RFID to Arduino Receiver
Card 1	ba-a6-70-15	140 - 141 ms
Card 2	be-a3-dc-22	140 - 141 ms
Card 3	dc-72-5d-39	140 - 141 ms
Card 4	dc-6c-e6-39	140 - 141 ms
Card 5	dc-f5-26-39	140 - 141 ms
Card 6	29-37-71-59	140 - 141 ms
Card 7	dc-97-d0-39	140 - 141 ms
Card 8	ec-3b-39-39	140 - 141 ms
Card 9	ec-34-46-39	140 - 141 ms
Card 10	dc-f4-9d-39	140 - 141 ms

Table 6. comparison with another research.

Study Literature	Sensor	Edge device type	Machine-to-Machine Communication	Publish protocol method	Time's total
[8]	RFID	Raspberry Pi	IEEE 802.11 Wi-Fi	MQTT	Not considered
[1]	Photodiode	FPGA board	Infrared	Binary code that operates on a pulse width of 27 ms.	1.944 ms
Proposed framework	RFID	Two Arduino UNO	Costume serial	Arduino JSON	141 ms

The JSON's data is then transferred to the Arduino receiver module through the wiring of the serial cable. The connection between Arduino to Arduino using a wiring serial cable can be said as the exchange of data in machine-to-machine communication schema. The total time spent to deliver the identification card data to the Arduino receiver module is between 140 to 141 milliseconds (mS). The spent time consists of time reading the id card in the Arduino sender module around 132 to 133 mS. Moreover, the time delay for transferring data from the Arduino receiver module to the laptop is around 8 mS.

References

[1] Praveen Kumar Reddy Maddikunta et al., "Industry 5.0: A Survey on Enabling Technologies and Potential Applications," *Journal of Industrial Information Integration*, vol. 26, p. 100257, 2022. [CrossRef] [Google Scholar] [Publisher link]

[2] Samuel C. Pereira et al., "Power Line Communication Technology Based on Morphological Filtering for Machine-To-Machine Applications," *Computers and Electrical Engineering*, vol. 93, 2021. [CrossRef] [Google Scholar] [Publisher link]

[3] Josep Petchamé et al., "A Qualitative Approach to Help Adjust the Design of Management Subjects in ICT Engineering Undergraduate Programs through User Experience in a Smart Classroom Context," *Sensors*, vol. 21, no. 14, p. 4762, 2021. [CrossRef] [Google Scholar] [Publisher link]

[4] Xiaohua Zhang, and Lin Chen, "College English Smart Classroom Teaching Model Based on Artificial Intelligence Technology in Mobile Information Systems," *Mobile Information Systems*, vol. 2021, 2021. [CrossRef] [Google Scholar] [Publisher link]

[5] Shuwen Yin et al., "Wireless Sensors Application in Smart English Classroom Design Based on Artificial Intelligent System," *Microprocess Microsyst*, vol. 81, p. 103798, 2021. [CrossRef] [Google Scholar] [Publisher link]

[6] Unnati Koppikar et al., "IoT based Smart Attendance Monitoring System using RFID," *2019 1st International Conference on Advances in Information Technology (ICAIT)*, 2019, pp. 193–197. [CrossRef] [Google Scholar] [Publisher link]

[7] Maredi Aritonang et al., "Implementation of Fingerprint Recognition Using Convolutional Neural Network and RFID Authentication Protocol on Attendance Machine," *Proceedings of the 2020 10th International Conference on Biomedical Engineering and Technology*, pp. 151–156, 2020. [CrossRef] [Google Scholar] [Publisher link]

[8] Aishwarya Raj Laxmi, and Ayaskanta Mishra, "RFID based Logistic Management System Using Internet of Things (IoT)," *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 556–559, 2018. [CrossRef] [Google Scholar] [Publisher link]

[9] F. Salinas Gómez, N. E. Suárez Escobar, and J. Vázquez, "RFID + Wi-Fi System to Control the Location of Biomedical Equipment Within Hospital Areas and Linked to an Intelligent Inventory," *Health Technology (Berlin)*, vol. 10, no. 2, pp. 479–483, 2020. [CrossRef] [Google Scholar] [Publisher link]

[10] Meghana P. Lokhande et al., "Machine-to-Machine Communication for Device Identification and Classification in Secure Telerobotics Surgery," *Security and communication networks*, vol. 2021, 2021. [CrossRef] [Google Scholar] [Publisher link]

[11] Mohd Javaid et al., "Significance of Quality 4.0 towards Comprehensive Enhancement in Manufacturing Sector," *Sensors International*, vol. 2, 2021. [CrossRef] [Google Scholar] [Publisher link]

[12] M. G. Sarwar Murshed et al., "Machine Learning at the Network Edge: A Survey," *ACM Comput Survey*, vol. 54, no. 8, 2022. [CrossRef] [Google Scholar] [Publisher link]

[13] Farshad Firouzi, Bahar Farahani, and Alexander Marinšek, "The Convergence and Interplay of Edge, Fog, and Cloud in the AI-Driven Internet of Things (IoT)," *Information Systems*, vol. 107, p. 101840, 2022. [CrossRef] [Google Scholar] [Publisher link]

[14] Rajkumar Rajavel et al., "IoT-based Smart Healthcare Video Surveillance System Using Edge Computing," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 6, pp. 3195–3207, 2022. [CrossRef] [Google Scholar] [Publisher link]

[15] Syafaruddin et al., "Arduino UNO AVR Atmega328 Microcontroller Based Dual Axis Solar Tracker Design," *ICIC Express Letters, Part B: Applications*, vol. 13, no. 4, pp. 337–345, 2022. [CrossRef] [Google Scholar] [Publisher link]

[16] Ali Abdulameer Aldujaili et al., "Smart Internet of Things Kindergarten Garbage Observation System Using Arduino UNO," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 6, pp. 6820–6828, 2022. [CrossRef] [Google Scholar] [Publisher link]

[17] Chee Ken Nee et al., "Design and Development of Face Mask Reminder Box Technology using Arduino Uno," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 7, pp. 92–97, 2022. [CrossRef] [Google Scholar] [Publisher link]

[18] Congcong Shi et al., "Identity Authentication with Association Behavior Sequence in Machine-to-Machine Mobile Terminals," *Mobile Networks and Applications*, vol. 27, no. 1, pp. 96–108, 2022. [CrossRef] [Google Scholar] [Publisher link]

[19] Yali Wu, Ningbo Zhang, and Kaixuan Rong, "Non-Orthogonal Random Access and Data Transmission Scheme for Machine-to-Machine Communications in Cellular Networks," *IEEE Access*, vol. 8, pp. 27687–27704, 2020. [CrossRef] [Google Scholar] [Publisher link]

[20] Yeduri Sreenivasa Reddy et al., “A Successive Interference Cancellation Based Random Access Channel Mechanism for Machine-to-Machine Communications in Cellular Internet-of-Things,” *IEEE Access*, vol. 9, pp. 8367–8380, 2021. [CrossRef] [Google Scholar] [Publisher link]

[21] Upendra Singh et al., “A Survey on LTE/LTE-A Radio Resource Allocation Techniques for Machine-to-Machine Communication for B5G Networks,” *IEEE Access*, vol. 9, pp. 107976–107997, 2021. [CrossRef] [Google Scholar] [Publisher link]

[22] Abdullah M. Almasoud, and Ahmed E. Kamal, “Wireless-Powered Machine-to-Machine Multicasting in Cellular Networks,” *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 2, pp. 515–528, 2020. [CrossRef] [Google Scholar] [Publisher link]

[23] Monireh Allah Gholi Ghasri et al., “Novel Relay Selection Algorithms for Machine-to-Machine Communications with Static RF Interfaces Setting,” *IEEE Access*, vol. 8, pp. 189989–190008, 2020. [CrossRef] [Google Scholar] [Publisher link]

[24] Prasad Challa, and B Eswara Reddy, “An Energy-Balanced Routing Algorithm Based on Forward Aware Factor for Machine-to-Machine Communication Networks,” *Indian Journal of Computer Science and Engineering*, vol. 12, no. 1, pp. 287–296, 2021, [CrossRef] [Google Scholar] [Publisher link]

[25] Sachin Kadam et al., “Fast Node Cardinality Estimation and Cognitive MAC Protocol Design for Heterogeneous Machine-to-Machine Networks,” *Wireless Networks*, vol. 26, no. 6, pp. 3929–3952, 2020. [CrossRef] [Google Scholar] [Publisher link]

[26] Denis Makarov et al., “Comprehensive Real-Time Pavement Operation Support System Using Machine-to-Machine Communication,” *International Journal of Pavement Research and Technology*, vol. 13, no. 1, pp. 93–107, 2020. [CrossRef] [Google Scholar] [Publisher link]

[27] I.Lakshmi, "The Internet of Things (IoT) Needs to Become a Reality in IT world," *SSRG International Journal of Computer Science and Engineering*, vol. 6, no. 2, pp. 23-33, 2019. [CrossRef] [Publisher link]

[28] Mohd Majid Akhtar et al., “Efficient Data Communication Using Distributed Ledger Technology and Iota-Enabled Internet of Things for a Future Machine-to-Machine Economy,” *Sensors*, vol. 21, no. 13, p. 4354, 2021. [CrossRef] [Google Scholar] [Publisher link]

[29] Shams A. Laghari, Selvakumar Manickam, and Shankar Karuppayah, “A Review on SECS/GEM: A Machine-to-Machine (M2M) Communication Protocol for Industry 4.0,” *International Journal of Electrical and Electronic Engineering and Telecommunications*, vol. 10, no. 2, pp. 105–114, 2021. [CrossRef] [Google Scholar] [Publisher link]

[30] Junhee Lee, Edman Anjos, and Srinivasa Rao Satti, “SJSON: A Succinct Representation for JSON Documents,” *Information Systems*, vol. 97, 2021. [CrossRef] [Google Scholar] [Publisher link]

[31] E. Rushdy, W. Khedr, and N. Salah, “Framework to Secure the OAuth 2.0 and JSON Web Token for Rest API,” *Journal of Theoretical and Applied Information Technology*, vol. 99, no. 9, pp. 2144–2161, 2021. [Google Scholar] [Publisher link]

[32] Ihda Chaerony Siffa, Jan Schäfer, and Markus M. Becker, “Adamant: A JSON Schema-Based Metadata Editor for Research Data Management Workflows,” *F1000Res*, vol. 11, 2022. [CrossRef] [Google Scholar] [Publisher link]

[33] Zouhaier Brahmia et al., “JUpdate: A JSON Update Language,” *Electronics (Switzerland)*, vol. 11, no. 4, p. 508, 2022. [CrossRef] [Google Scholar] [Publisher link]

[34] Mohd Kamir Yusof et al., “Native JSON Model for Data Integration in Business Intelligent Applications,” *Journal of Theoretical and Applied Information Technology*, vol. 100, no. 18, 2022. [Google Scholar] [Publisher link]

[35] Jung Kyu Park, and Eun Young Park, “Search Performance Evaluation of JSON Datasets in a Big Data Environment,” *International Journal of Mechanical Engineering*, vol. 7, no. 1, pp. 198–203, 2022. [Google Scholar] [Publisher link]

[36] Ping Tan et al., “Teaching Management System with Applications of RFID and IoT Technology,” *Education Science (Basel)*, vol. 8, no. 1, p. 26, 2018. [CrossRef] [Google Scholar] [Publisher link]

[37] Blanchon, Benoıt, C++ JSON Library for IoT Simple and Efficient, 2018. [Online]. Available: *https://github.com/bblanchon/ArduinoJson*