Development of Controller For Internet of Things Based Anti Pollution Smart Toll Gate System

Wahyul Amien Syafei
Department of Electrical Engineering
Diponegoro University
Semarang, Indoneisia
https://orcid.org/0000-0002-6058-2693

Raihan Afiq
Lawencon International
Wisma Staco 5th Floor. Casablanca
Jakarta, Indoneisia
raihan.afiq @gmail.com

Achmad Hidayatno
Department of Electrical Engineering
Diponegoro University
Semarang, Indoneisia
achmad.hidayatno@gmail.com

Wahyudi
Department of Electrical Engineering
Diponegoro University
Semarang, Indoneisia
wahyudi @gmail.com

***Abstract*—The transaction system applied onto the current toll gate system is less effective in overcoming the queue of vehicles, especially during public holidays or long holidays. Although it is already based on contactless technology, but toll road users still need to juxtapose the e-Toll card manually for four seconds. This is in time causing the queue and density in several toll gates. To alleviate this problem, the toll provider cooperates with bank to issue the e-toll system On Board Unit. This device can strengthen the e-Toll card signal to make reading distance farther. However, it is not popular in the community due to the price. This paper deals with the hardware and software development of controller for IoT based Anti-Pollution Smart Toll Gate system which has lower production costs. Two testing scenarios have been conducted. Ultrasonic sensor reading distance testing shows that the effective distance is approximately 300 cm. Testing the system response to vehicle speed shows that the system is able to work well with vehicle speed up to 20 km/h.**

Keywords— Toll Gate, IoT, RFID, Smart System, Telegram

I. INTRODUCTION

Indonesia has an area of 1.9 million km2. In order to meet the distribution of infrastructure development and distribution of broad needs, the government is intensifying toll road construction projects in several regions. The toll road that is currently being built will operate automatically so it is called Automatic Toll Gate (GTO). GTO which is a type of Electronic Toll Collection (ETC) applies contactless technology to conduct the transactions. Toll road users need to hold RFID cards (tags) for about four seconds so that the RFID reader can process the transactions.

During rush hour, early and weekends, and long holidays, the current GTO system still sometimes causes queues that hamper the speed of the vehicle because the toll gate gates only open after the transaction process is complete. The density that occurs at the toll gate causes health problems directly or indirectly [1].

To alleviate this problem, the toll provider cooperates with bank to issue the e-toll system On Board Unit. This device can strengthen the RFID card signal so thath the reading distance becomes farther. However, it is not popular in the community due to the price. Further, an evidence of toll road transactions in Indonesia until now is not environmentally friendly because it still uses printed paper which is usually left or discarded by toll road users.

Several studies have been conducted to improve the performance of the ETC system. Omsakthi, et al. proposed an automatic toll gate payment system based on vehicle location on the road using the Global Navigation Satellite Systems signal [2]. Bhavke et al. designed an RFID-based toll payment system with a reader using Arduino Mega [3]. Punniamoorthy, et al. designed a toll gate system using the Xilinx Spartan 3AN FPGA as the main device controlling the system [4]. Geetha et al. designed a toll gate system using Bluetooth Low Energy technology as a user RFID tag reader [5]. Devi et al. design a toll payment system using Automatic Number Plate Recognition as the core of the system using ANPR RoadWolf Infrared Video Cameras [6]. The system above though sophisticated but relatively expensive.

On another chance Devi, et al proposed payment of toll road transactions based on license plates [7]. Bhat and Pai designed an automatic control system using Arduino for the ETC system. [8]. Whereas Ramakrishna and Reddy developed the ETC system using Raspberry pi-2. [9].

This paper deals with the development of controller for IoT based anti-pollution Smart Toll Gate (STG) system. It uses an advanced and economical Arduino UNO and Raspberry pi-3 Microcontroller. This STG is a development of a system that previously used SMS notifications. [10, 11]. The developed system can identify the vehicle for both registered (having RFID) and not registered toll road user. This system does not require the driver to stop the vehicle at the toll gate because the transaction process starts when the vehicle triggers an ultrasonic sensor put on the toll gate side. The transaction system also equipped with smart billing system which charges the toll road users according to the type of vehicle detected by the system's camera. It also replaces the payment receipt of toll gate transactions that are currently in the form of printed paper into digital messages through Telegram. This leads to an efficient, effective, anti-pollution, and environmentally friendly transportation sub-system.

II. CONTROLLER DESIGN OF IOT BASED STG SYSTEM

Arduino UNO and Raspberry pi-3 microcontroller implementation to control the STG system consists of hardware and software design. Hardware design includes designing On Board Unit (OBU) and OBU reader. Software design is done using Arduino IDE and Node-RED.

A. Hardware Design

The hardware design is divided into two, the design of the OBU which is mounted on the vehicle and the OBU reader which is implemented on the toll gate to detect the OBU.

B. On Board Unit (OBU)

OBU consists of an ATmega328-based Arduino UNO Microcontroller, RFID Reader MIFARE (MF)-RC522 module, and the HC-12 433MHz Serial UART Radio Frequency (RF) module. The General Purpose Input Output (GPIO) wiring and pinout on OBU can be seen in Figure 1.

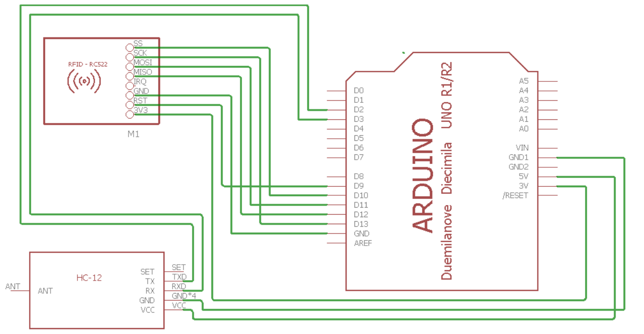

Fig. 1. GPIO of the developed OBU, consists of Arduino UNO Atmega328, RFID reader MF-RC522, and RF-HC12 433MHz.

The implementation of the MF-RC522 and RF-HC12 433MHz RFID module to the Arduino UNO on OBU is briefly explained below.

1) Implementation of RFID reader MF-RC522 module

RFID reader module MF-RC522 uses synchronous communication Serial Peripheral Interface (SPI). MF-RC522 pins are connected to Arduino UNO digital pins following Table I.

TABLE I. MF-RC522 PIN CONNECTIONS TO ARDUINO UNO

No	Pin MF-RC522	Pin Arduino UNO
1	RST	D9
2	SS	D10
3	MISO	D12
4	3V3	3V
5	Ground	Ground
6	IRQ	-

The IRQ pin on the MF-RC522 is not used because this system does not require an interrupt. The implementation of MF-RC522 to Arduino UNO is shown in Figure 2

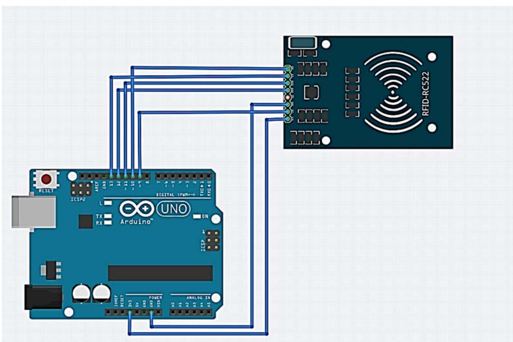

Fig. 2. Implementation of MF-RC522 to Arduino UNO.

The RFID reader module MF-RC522 is communicated to Arduino UNO using the Arduino sketch program list. The success of the implementation can be checked on the Arduino IDE serial monitor as shown in Figure 3.

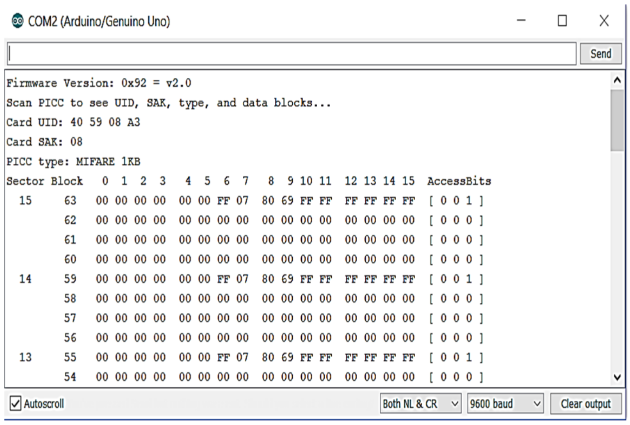

Fig. 3. Testing the implementation of the MF-RC522 module displayed on the Arduino IDE serial monitor.

Figure 3 shows that the Arduino IDE displays the information of Firmware Version, UID Card, SAK Card, PICC type, and Sector Block. However, the system only requires UID Card information as indicated by a red circle. RFID tag that had been read have a unique identity (UID) 40 59 08 A3. This result indicates the RFID reader module is implemented successfully on the OBU.

2) Implementation of RF-HC12 433MHz module

RF-HC-12 module uses asynchronous serial communication. There are four pins that are used to communicate with Arduino UNO, namely TX, RX, VCC, and ground. These pin connections are based on Table II.

TABLE II. RF-HC-12 PIN CONNECTIONS TO ARDUINO UNO

No	Pin RF-HC-12	Pin Arduino UNO
1	TX	D2
2	RX	D3
3	VCC	Output 5V
4	Ground	Ground

Implementation of the RF-HC-12 module connection to Arduino UNO is shown in Figure 4.

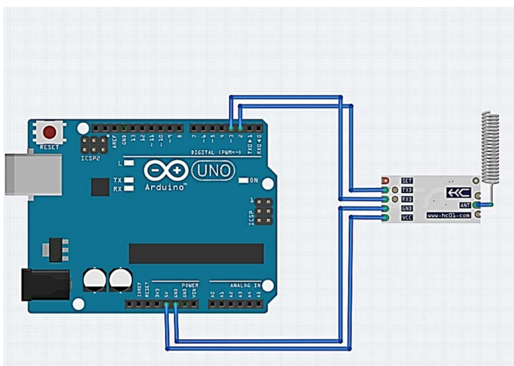

Fig. 4. Implementation of RC-HF-12 to Arduino UNO.

The UID data frame sent by OBU is received by the RF-HC-12 module on the OBU reader. The receipt of UID data frames can be checked using the Raspberry terminal as shown in Figure 5.

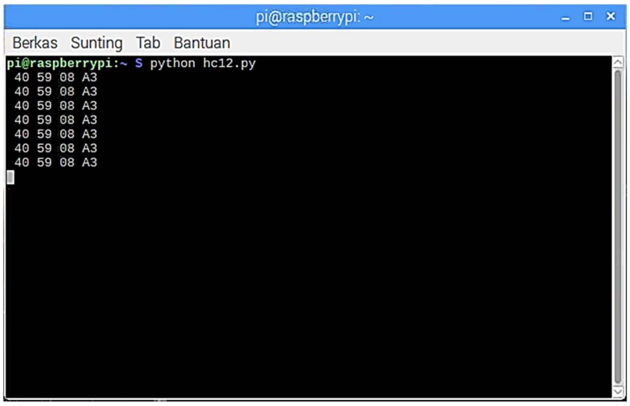

Fig. 5. The receipt of UID data frames by the RF-HC-12 module displayed on the Raspberry terminal.

By entering command `python hc12.py` the Raspberry terminal displays the message received. Figure 5 shows the reading of UID data frames by RF-HC-12 is 40 59 08 A3 which is the unique identity of the RFID tag. The sequence of numbers appears seven times because the test is carried out for seven seconds with a delay of one second each reading. This result indicates that the RF-HC-12 module has been implemented correctly in Arduino UNO. The ready to operate OBU unit is shown in Figure 6.

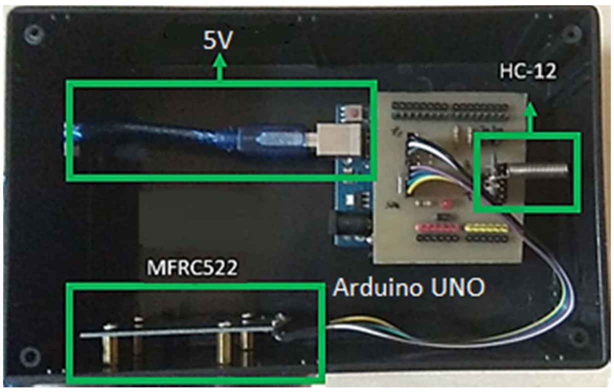

Fig. 6. The developed OBU is ready to operate

C. *On Board Unit (OBU)Reader*

OBU reader is a STG subsystem implemented on the toll gate side. There are four main modules, namely Raspberry pi-3, ultrasonic sensor HC-SR04, RF-HC-12 433MHz, and webcam. The design of the GPIO on the OBU reader is shown in Figure 7.

The implementation of the ultrasonic sensor HC-SR04 and RF-HC-12 to the Raspberry pi-3 on the OBU reader requires two additional circuits namely a voltage divider (VD) to reduce the voltage from 5V to 3.3V and the voltage level shifter (VLS) to increase the voltage from 3.3V to 5V.

The HC-SR04 and RF-HC-12 modules output a voltage of 5V while each Raspberry pi-3 GPIO is only able to work at a voltage of 3.3V so VD is needed. On the other hand, the Raspberry GPIO is only able to provide a voltage of 3.3V while the HC-SR04 and HC-12 modules require an input voltage of 5V so a VLS is needed. This is to prevent Raspberry from damage. [12, 13].

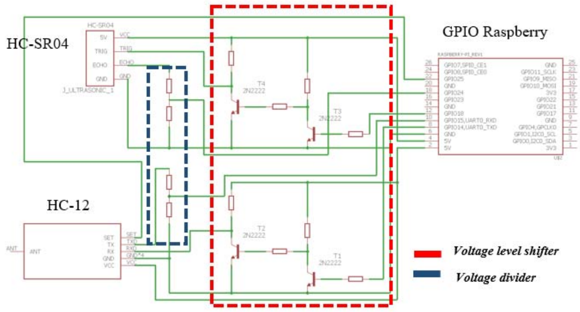

Fig. 7. GPIO of the developed OBU reader consists of Raspberry pi-3, ultrasonic sensor HC-SR04, and RF-HC12 433MHz.

The implementation of the ultrasonic sensor HC-SR04 and RF-HC12 433MHz modules to the Raspberry pi-3 on the OBU reader, is briefly explained below.

1) Implementation of ultrasonic sensor HC-SR04

Pins connection of the ultrasonic sensor HC-SR04 module to the Raspberry pi-3 on the OBU reader follows Table III.

TABLE III. HC-SR04 PIN CONNECTION TO RASPBERRY PI-3

No	Pin RF-HC-12	Interface	Pin Arduino UNO
1	ECHO	VD	18 (GPIO 24)
2	TRIG	VLS	12 (GPIO 18)
3	VCC		Output 5V
4	Ground		Ground

The ultrasonic sensor HC-SR04 module is programmed and tested for its ability using the Python script program list.

2) Implementation of RF-HC-12 module

Pins connection of RF-HC-12 module to Raspberry pi-3 on OBU reader according to Table IV.

TABLE IV. RF-HC-12 PIN CONNECTION TO RASPBERRY PI-3

No	Pin RF-HC-12	Interface	Pin Arduino UNO
1	TX	VD	10 (GPIO 5)
2	RX	VLS	8 (GPIO 14)
3	SET		22 (GPIO 25)
4	VCC		Output 5V
5	Ground		Ground

The implementation of ultrasonic sensor HC-SR04 and RF-HC-12 modules connection to the Raspberry pi-3 on the OBU reader is shown in Figure 8.

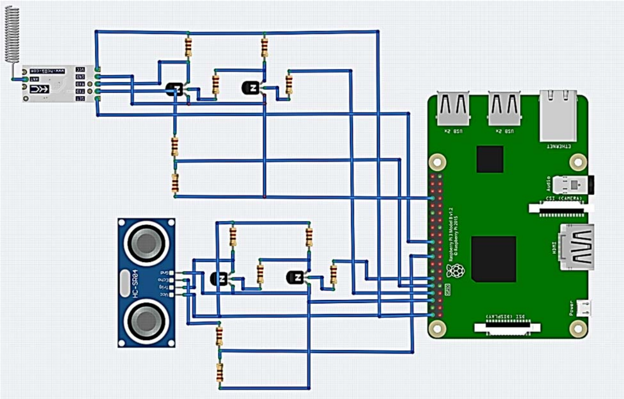

Fig. 8. Implementation of ultrasonic sensors HC-SR04 and RF-HC-12 to Raspberry pi-3 on OBU reader.

The ready to operate developed OBU reader unit is shown in Figure 9.

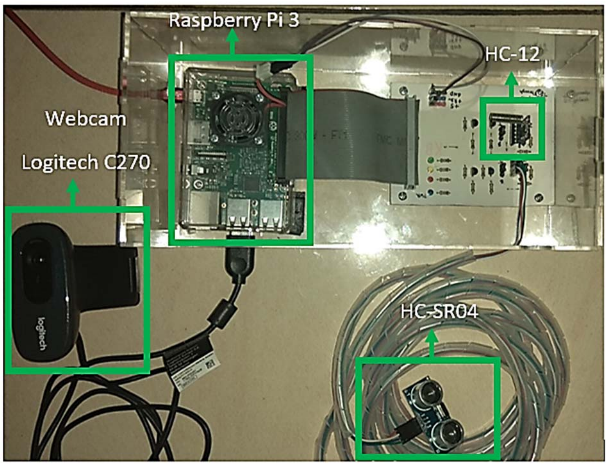

Fig. 9. The developed OBU reader is ready to operate.

D. Software development

Software development is done on OBU and OBU readers. OBU has the duty to read RFID tags and send the UID to the OBU reader using the communication module. The OBU is programmed using the Arduino IDE. OBU reader has the duty to process the received UID, activate the webcam, and set the delay system. The OBU reader is programmed using the Node-RED which is default Raspberry pi-3 application. [14].

1) Coding for OBU

The Arduino IDE application is used to program the RFID reader MF-RC522 module to read RFID tag and send `the UID to the OBU reader using the RF-HC-12 communication module.

Communication between RFID tags and RFID Reader MF-RC522 uses USART serial communication. The process of reading and writing UID data to memory uses the Serial Peripheral Interface (SPI) which is dedicated to handle synchronous serial communication in Arduino.

2) Coding for OBU reader

The control block is implemented in the OBU Reader. Control block programming is done using Node-RED software inherit in Raspberry. This block controls the webcam to capture vehicle images, UART serial communication, and adjusts system delay. The control block in Node-RED is shown in Figure 10.

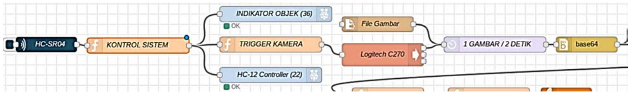

Fig. 10. Implementation of Node-RED for control block..

Python node HC-SR04 is the initial node to measure the distance of the object. Python HC-SR04 node is forwarded to the CONTROL SYSTEM node function. If the ultrasonic sensor output is between 290 and 300 cm then it is in condition 1. If not then it is at condition 0. The output function of node CONTROL SYSTEM becomes input for OBJECT INDICATOR Node (36), HC-12 Controller Node (22), and the CAMERA TRIGGER Function node.

OBJECT INDICATOR Node (36) indicates control at condition 0 or 1. The configuration of OBJECT INDICATOR Node (36) is shown in Figure 11.

OBJECT INDICATOR node (36) is connected to pin 36 (GPIO16) which triggers the green LED on the OBU Reader. Condition 0 (acitve low) will turn on the green LED, which indicates the STG system is ready to process the vehicle.

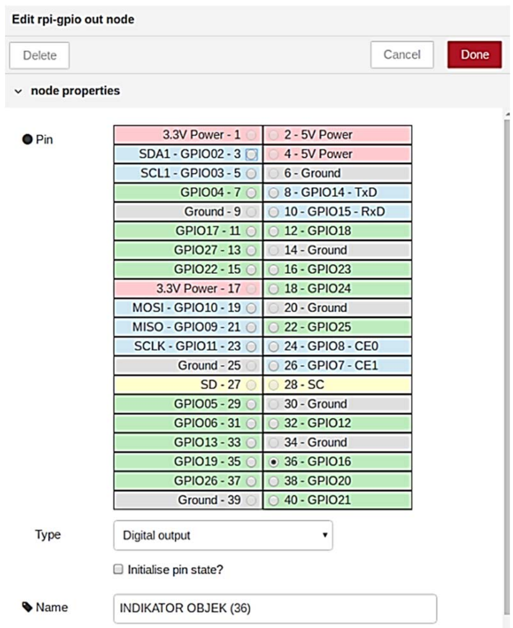

Fig. 11. Node RED configuration for the Object Indicator.

III. Testing Scenario of the STG Control System

The first testing is ultrasonic effective reading distance. The ultrasonic sensor HC-SR04 is put in line of sight to the receiver. Starting from several cm and focus on 300 cm to measure the maximum reading distance. Results of the HC-SR04 sensor implementation which can be observed by Raspberry serial terminal is shown in Figure 12.

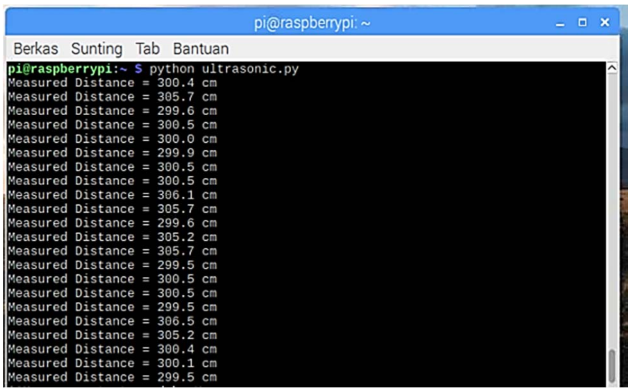

Fig. 12. Testing the reading distance of ultrasonic sensor HC-SR04 implemented to the Raspberry pi-3 on the OBU reader. Displayed using the Raspberry terminal.

The results of ultrasonic sensor HC-SR04 at the same reading distance of 300 cm show different values. Standard deviation (s) of the reading distance by the sensor is calculated using equation 1.

$$s = \sqrt{\frac{\sum_{i}^{n} (x_i - \bar{x})^2}{n}} \qquad (1)$$

where $\bar{x}$ is 300 cm, yields s = 3,259 cm

while ultrasonic sensor HC-SR04 reading error standard is obtained as:

$$e = \frac{s}{\sqrt{n}} = \frac{3{,}259}{\sqrt{22}} = 0{,}694 \text{ cm}$$

These results indicate that the ultrasonic sensor HC-SR04 has been implemented to Raspberry pi-3 correctly and ready to operate.

The second test is the response of the STG control system to the vehicle's speed. This scenario begins by giving the OBU power supply using the USB port found in the vehicle. The next step is to observe the output of the Node-RED debug node when the vehicle passes an ultrasonic sensor and OBU reader on the toll gate. The Node-RED display when testing is conducted is shown in Figure 13.

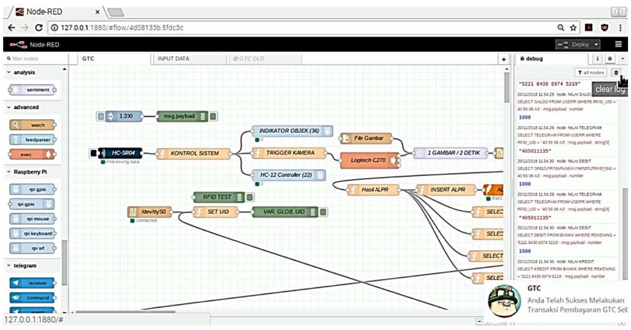

Fig. 13. Display of Node-RED for testing the response of the STG Control System to the vehicle's speed.

It is shown that the control system can work well at a vehicle speed of 10 km/h. This is indicated by the appearance of the output notification for each debug node until notification comes out on the Telegram Bot. When the vehicle's speed was increased to 20 km/h, the OpenALPR output did not produce a license plate payload so the process did not continue at the next stage. At speed of 30 km/h, the system did not emit anything because the ultrasonic sensor could not detect the passing objects. Data on the results of the test are shown in Table V.

TABLE V. TESTING THE RESPONSE OF THE STG CONTROL SYSTEM TO THE VEHICLE SPEED.

OBU speed (km/h)	Controller	UART port	Webcam
10	ON (1)	ON (1)	ON (1)
20	ON (1)	ON (1)	ON (1)
30	OFF (0)	OFF (0)	OFF (0)

IV. CONCLUSIONS

The hardware and software design of controller for the Internet of Things Anti-Pollution Smart Toll Gate prototype control system has been completed and examined. Ultrasonic sensor reading distance testing shows that the effective distance is approximately 300 cm. Testing the system response to vehicle speed shows that the system is able to work well with a speed up to 20 km/h. Further development is to increase the speed of the system's response to the vehicle's speed and reduce the processing time from the time the vehicle is detected until the notification is sent.

ACKNOWLEDGEMENT

This research was financially supported by The Faculty of Engineering, Diponegoro University, Indonesia through Strategic Research Grant 2020.

REFERENCES

[1] Sodikin, "Kajian Masalah Antrian Pada Sistem Pengumpulan Tol Konvensional Terhadap Rancangan Sistem Pengumpulan Tol Elektronik," UNDIP, 2006.

[2] A. A. Chapate and D. D. Nawgaje, "Electronic Toll Collection System Based on ARM," vol. 4, no. 1, pp. 46–49, 2015.

[3] R. Omsakthi, "Automatic Toll collection system using GNSS signal," vol. 1, no. 10, pp. 142–145, 2014.

[4] S. Mahajan et al., "Automatic Toll Gate System Using Advanced RFID and GSM Technology," vol. 4, no. 1, pp. 565–568, 2016.

[5] D. Punniamoorthy and P. V. Kumar, "Modified Toll Gate System with Enhanced Security Using FPGA," vol. 4, no. 2, pp. 52–61, 2014.

[6] M. Geetha, B. Sangeetha, J. Vidhya, S. Ramya, and L. N. K. Ragavi, "Automatic Toll Gate Payment System Using Hybrid Mechanism," no. 9, pp. 18–21, 2016.

[7] D. Devi, M. Radha, P. Anuja, and R. Anusha, "Electronic Toll Collection Using Automatic Number Plate Recognition," International Journal Latest Trends Eng. Technol., 2017.

[8] V. Bhat and P. Pai, "Arduino Based RFID Controlled Automatic E-TOLL Collection System," International Journal of Emerging Research in Management &Technology, vol. 9359, no. 8, pp. 274–276, 2017.

[9] S. Ramakrishna and K. K. Reddy, "Smart Toll Collection using Raspberry Pi 2," International Research Journal of Engineering and Technology (IRJET), vol. 03, no. 14, pp. 1390–1394.

[10] M. A. Fatkhurrahman and W. A. Syafei, "Perancangan Prototipe Sistem Gerbang Tol Cerdas Berbasis RFID Dan Notifikasi Pembayaran Via Social Messenger," TRANSIENT, vol. VOL. 6, NO, 2017.

[11] A. F. Listyono and W. A. S, “Perancangan Prototipe Identifikasi Kendaraan Jalan Tol Berbasis RFID Dan Notifikasi Pembayaran Via SMS,” TRANSIENT, vol. VOL.4, NO., 2015.

[12] Elec Freaks, “HC SR04 Ultrasonic Ranging Sensor Module.” [Online]. Available: https://www.pcbway.com/blog/News/New_product_in_gift_shop__HC_SR04_Ultrasonic_Ranging_Sensor_Module.html.

[13] Magpi, “Magpi - Raspberry Pi 3 Model B Specs,” 2016. [Online]. Available: https://www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/.

[14] J. Foundation, “Node-RED.” [Online]. Available: https://nodered.org/.