

Internet of Things-Based Home Trash Capacity Tracking System with Instant Notifications

Era Sari Munthe¹, Karno Diantoro², Agus Herwanto³

¹Jayabaya University, Indonesia

²STMIK Mercusuar, Indonesia

³Esa Unggul University, Indonesia

Correspondent: karno@mercusuar.ac.id²

Received : June 23, 2024

Accepted : July 12, 2024

Published : July 27, 2024

Citation: Munthe, E. S., Diantoro, K., & Herwanto, A. (2024). Internet of Things-Based Home Trash Capacity Tracking System with Instant Notifications. *Intellecta : Journal of Artificial Intelligence*, 2(3), 136-154.

<https://doi.org/10.61978/digitus.v2i3.257>

ABSTRACT: Garbage created from routine household activities is collected and stored in household garbage cans. Location This garbage makes rubbish collection easier to live in and helps to maintain a clean household environment. Household garbage cans are often designed to fit specific demands and feature a tight-fitting cover to keep sickness and animals out and to minimize unwanted odors. The layout To stop the spread of bacteria or fungus, something must be easy to clean. Lack of technology to monitor garbage bin fullness and inability to precisely monitor fill capacity, which can lead to trash overflow, offensive odors, and animal nuisances. Thus, volume sensorization techniques and Internet of Things (IoT) technologies are the answers to this challenge. To enable real-time waste capacity volume monitoring and to give users level information about trash charging through the Blynk platform, the system will deliver When the garbage can is full, an alarm sensor-equipped warning will ring. The Arduino IDE and the C programming language are the software used. The findings of the study demonstrate that the garbage can capacity monitoring system The information about waste filling levels that are provided in real-time by this IoT-based system is effective. By using this approach, homeowners can easily keep a clean and healthy home environment by knowing when it's time to remove the trash.

Keywords: Blynk Platform, Sensorization, and Internet of Things (IoT).



This is an open-access article under the CC-BY 4.0 license

INTRODUCTION

Keeping the environment clean and healthy requires effective household trash management (Tiwari & Dixit, 2017; Wijaya et al., 2017). However, the inability to precisely manage garbage bin capacity frequently results in issues like trash overflow, offensive odors, and interruptions from wild animals. The emergence of Internet of Things (IoT) technology presents prospects for resolving this issue via an effective and instantaneous monitoring framework (Parkash & Rajendra, 2016). IoT-Based Waste Management for Smart City (Gherbia & Zehani, 2024; Zavare et al., 2017).

The goal of this research is to develop and deploy an Internet of Things (IoT)--based system for

monitoring household garbage can capacity and sending users immediate notifications (Gerard et al., 2024). It is anticipated that this approach will preserve environmental cleanliness and boost domestic waste management efficiency (Folianto et al., 2015).

This study builds a prototype Internet of Things system using an experimental methodology. The components of the system include a communication module for transmitting data to the cloud platform, a microcontroller for data processing, and a volume sensor positioned on the bin. The Arduino IDE and the C programming language were used for software development (Atzori et al., 2010). Analysis Tools: To ascertain the garbage can's capacity, data gathered from sensors will be examined using a signal processing technique. Notifications to users and data visualization are done via the Blynk platform. Reliability of notifications, response time, and measurement precision will be used to assess system performance (Medvedev et al., 2015).

METHOD

The volume sensorization methodology combines data from communication and processing systems with the use of sensors to measure volume to provide efficient management and monitoring. It is envisaged that this methodology will improve the efficacy of controlling the household waste bin ladder by monitoring the content capacity of IoT-based trash cans based on volume (Catania & Ventura, 2014). The ultrasonic sensor is used to measure the amount or height of rubbish by placing it inside the trash can. Ultrasonic sensors send their acquired data to an internet-connected microcontroller, which subsequently sends it to IoT platforms such as Blynk. Through the Blynk platform, users can monitor the bin's filling level in real-time (Dubey et al., 2017; Giacobbe et al., 2018).

The system also indicates in red when the bin gets close to or fills, and an infrared sensor next to the ultrasonic sensor will sound an alert if the volume capacity of the waste has reached its maximum level. The information gathered is examined to identify trends in the use of the bins and to assist in organizing and making decisions about what can be disposed of in the garbage. To prevent a buildup of waste in the home environment, it is, therefore, possible to monitor the capacity of trash cans and staircases more effectively.

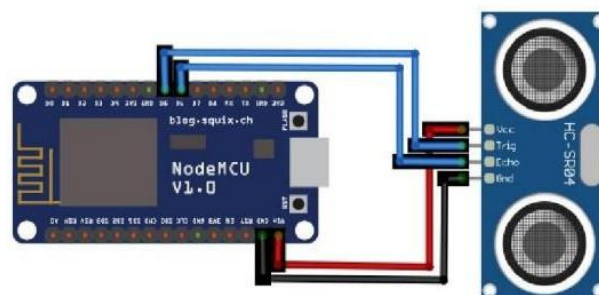


Figure 1. Volume Sensorization Methodology

In the book *Programming Arduino: Getting Started with Sketches*, Simon Monk (2019) describes that the volume sensorization methodology uses electronic sensors like infrared or ultrasonic

sensors to measure and monitor volume in a given environment. The microcontroller subsequently processes this data and sends it to a platform that establishes a connection with users.

This study employs an IoT-based prototype design and an experimental methodology. The process for monitoring the capacity of residential garbage bins includes building a volume sensor system, integrating it with an IoT platform, and analyzing real-time data (Bourougaa-Tria et al., 2022). Because it enables direct evaluation of a system's efficacy in real-world settings and offers quantifiable data for a system's correctness and performance analysis, Encourages quick iterations for the creation and improvement of prototypes.

The goal of this methodology is to guarantee thorough data collecting and in-depth analysis. While IoT connection makes accurate real-time data collecting possible, the experimental approach enables immediate evaluation of system effectiveness in real-world settings (Gibson, 2023; Hong et al., 2014; Idwan et al., 2020). Modern machine learning algorithms and conventional statistical approaches are included in the chosen data analysis methods, which offer deep insights into system performance and future optimization possibilities (Jha et al., 2018; Kale & Lanjewar, 2017).

RESULT AND DISCUSSION

Block Diagram Creation

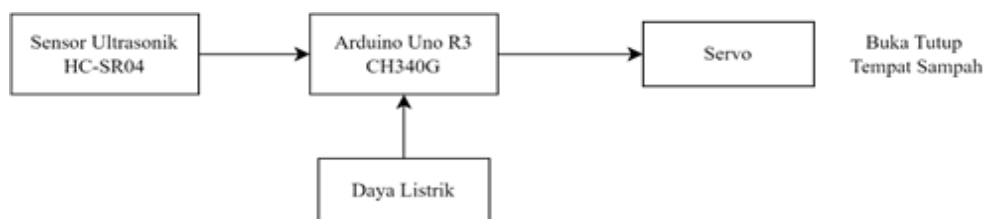


Figure 2. Design of Opener Block Diagrams

Block schematic design for Opener in IoT-Based Trash Can Fill Capacity Monitoring is shown in Figure 2 (Aazam et al., 2016). This block diagram shows how the Arduino Uno R3 CH340G, which has been powered, processes the input from the HC-SR04 Ultrasonic Sensor used in the opener to detect movement on the front of the trash can. This allows the Arduino Uno R3 CH340G to produce an output that powers the servo that opens the container's lid. trash.

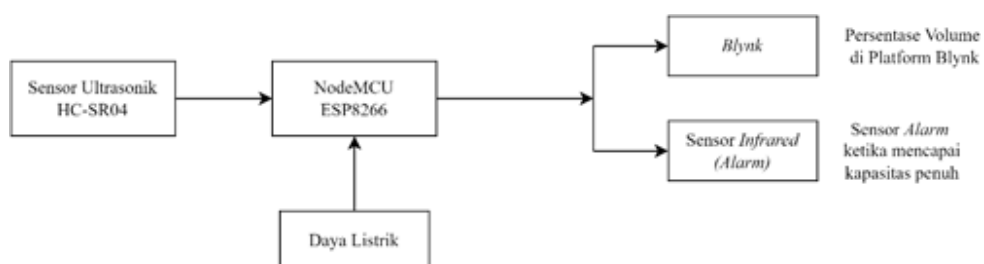


Figure 3: Design of Volume Block Diagram

A block diagram design for Volume in the Application of Internet of Things-Based Trash Can Fill Capacity Monitoring is shown in Figure 3. According to this block diagram, the HC-SR04 Ultrasonic Sensor is utilized as an input to determine the volume of the trash can's contents based on the trash can's volume. Subsequently, the NodeMCU ESP8266, which is powered and connected to the Wi-Fi SSID, will process it. This will enable it to generate two outputs: one that will show the percentage of the trash can's volume capacity on the Blynk platform and another that will trigger an alarm when the trash can's contents are fully utilized.

The circuit's overall operation serves as the foundation for the block diagram design. It is evident from the block diagram above that this system is built with multiple input, process, and output components (Vinithra & Bharathi, 2018).

a. Block for Input from Ultrasonic Sensor

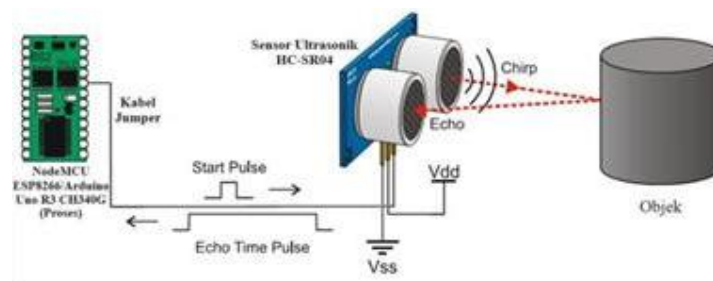


Figure 4. Block of Ultrasonic Sensor Input for Volume and Opener

Figure 4. The operation of the Ultrasonic Sensor in the system is shown in the block diagram. Two HC-SR04 ultrasonic sensors of the same type are used as inputs in this Internet of Things-based trash can fill capacity monitoring system (Sheng et al., 2020; Srinivasan & Latha, 2019). The garbage can lid is opened using the first input, an HC-SR04 ultrasonic sensor that serves as an opener and is coupled to a servo. Secondly, the HC-SR04 ultrasonic sensor is employed as a volume sensor, tracking the volume of the garbage can's contents. The two primary parts of the ultrasonic sensor are the Echo, which serves as a receiver for ultrasonic sound waves that are reflected, and the Trigger, which transmits or sends sound waves. This sensor operates by having the transmitter deliver a sound signal in the direction of the front, and if an object is in the way, the object will reflect the sound to the receiver.

b. Block Procedures

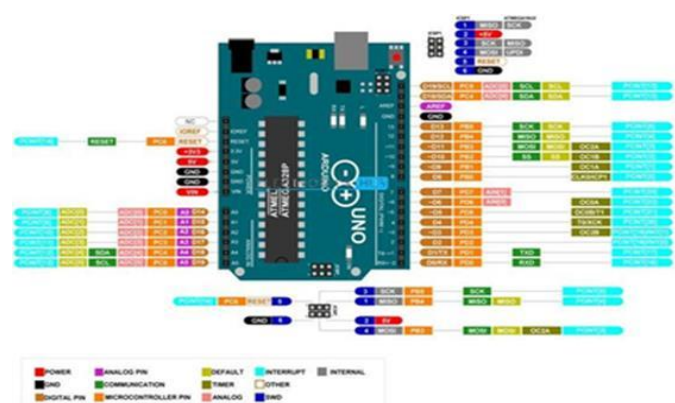


Figure 5. Arduino Uno R3 CH340G Opener Process Block

The process block in this tool, the Arduino Uno R3 CH340G, is described in Figure 5. It serves as a control center for the actions taken, particularly processing data received from the ultrasonic opening sensor. The Arduino Uno's program will process every logical input that enters and identify the right output to open the garbage can lid that is attached to the servo.

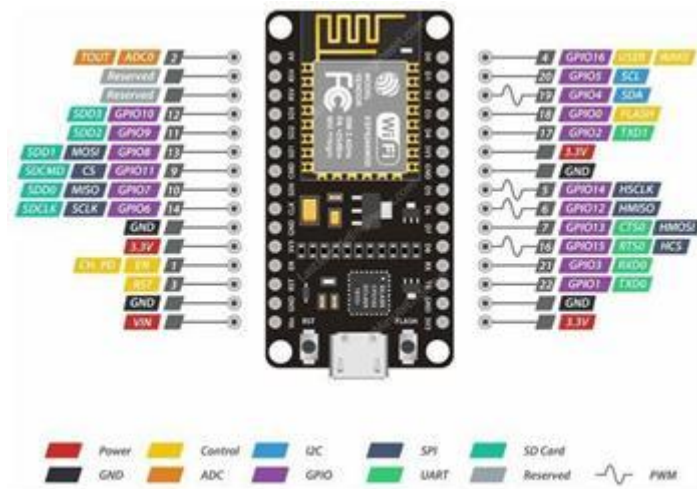


Figure 6: NodeMCU ESP8266 Volume Process Block

The ESP8266 is a process block in this tool that serves as a control center for the actions taken, such as processing data received from the ultrasonic volume sensor, as shown in Figure 6. The program installed in the ESP8266 will process any logic input that enters the device and identify the proper output. To display the volume output capacity of the bin contents on the Blynk platform, this microcontroller needs to be connected to the internet since it needs a wireless network to communicate with the database on the cloud server (Thakker & Narayanamoorthi, 2015).

c. Blok Output

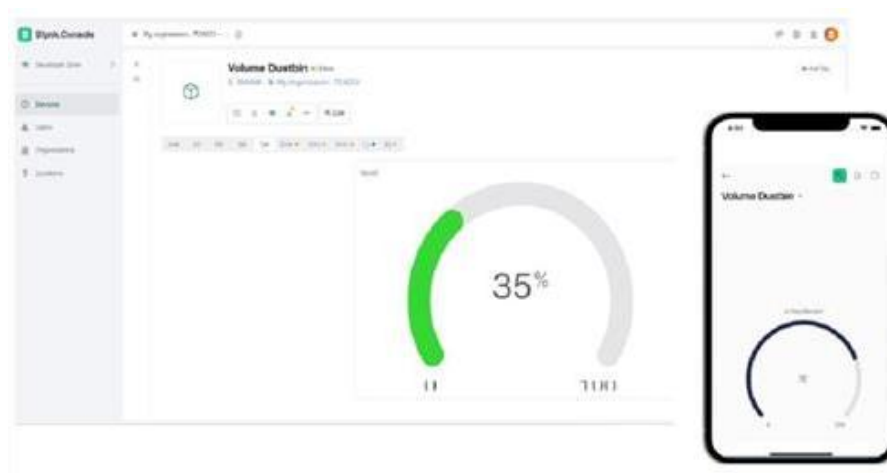


Figure 7: Blynk's Volume Output Block

The output block from the Blynk platform is shown in Figure 7. Blynk is an Internet of Things

platform that lets users utilize web or mobile applications to manage physical devices, like the Arduino IDE(Rath et al., 2018; Saha et al., 2017; Shamin et al., 2019). Using a variety of widgets, users may monitor or control IoT devices in real-time using the platform's user-friendly graphical interface(Kang et al., 2022; Kumar et al., 2016). The volume output block uses the blynk platform to more effectively and in real-time monitor the trash can's volume capacity. This may be done from anywhere at any time using a mobile device. The ESP8266 will process the data supplied from the ultrasonic sensor and output the volume % to the Blynk platform.

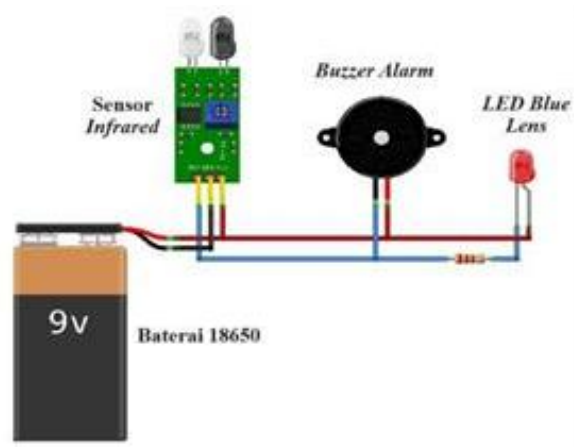


Figure 8: Output Block for Alarm Sensor

The output block employing an infrared sensor as an alert sensor is described in Figure 8. When the trash can's volume reaches its maximum, the infrared sensor which is placed next to the ultrasonic volume sensor sounds an alert. An on/off switch on this sensor allows the alarm to be manually turned on and off. It gets its electrical power from batteries.

Using the Arduino IDE

a. Creating Opener Software with the Arduino IDE

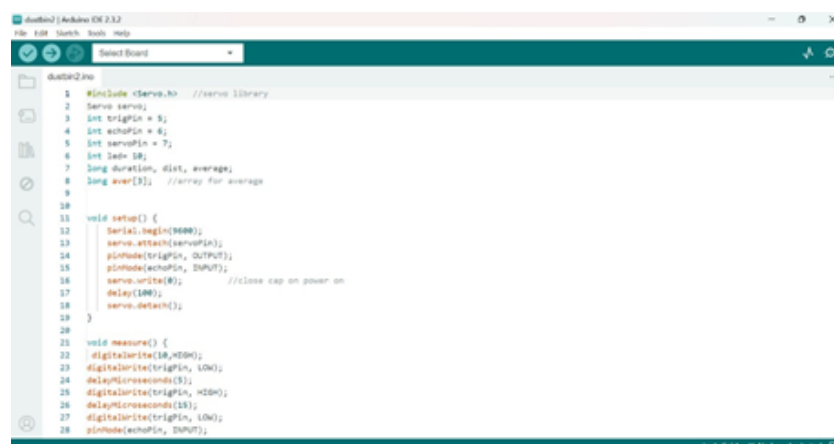


Figure 9: Display for Arduino IDE Opener

A C script code from the Arduino IDE program is shown in Figure 9. IoT-based garbage cans can be opened and closed using this opener software(Lozano et al., 2018; Malik & Sumpena, 2023). It

recognizes motions made by hands or trash that is intended to be thrown in the trash can. The garbage can's lid will open for three seconds once the ultrasonic sensor detects something, then close once more.

b. Arduino IDE Volume Software Design



Figure 10. Arduino IDE Volume Display

AC script from the Arduino IDE program is shown in Figure 10. The purpose of this volume software is to measure or keep track of the garbage can's capacity. The ESP8266 will be used to connect the software's loudness utilizing the wifi SSID. Then, the blynk platform may be used to track garbage bin capacity data(Anagnostopoulos et al., 2017; Bharadwaj et al., 2016).

Application

a. Put Hardware Opener Into Practice

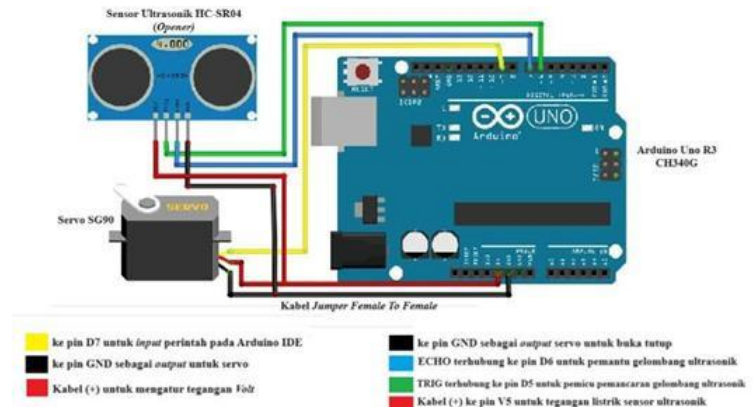


Figure 11: Circuit Board Hardware Opener

The garbage bin content capacity monitoring system's Arduino Uno Servo control circuit with an ultrasonic sensor is described in Figure 11. The Arduino Uno board serves as an automatic distance control with an angle point between 0° and 180° utilizing an ultrasonic sensor. The Arduino Uno will be connected to the SG90 Servo in this system by attaching the servo's cable to the pin on the Arduino Uno board. Through the use of a jumper cable, the Arduino Uno pin and the ultrasonic

sensor will be connected to the Arduino Uno board. Since human ears cannot detect noises at such a high frequency, ultrasonic sensors operate with sound waves that are 20 kHz. This system's ultrasonic sensor doubles as a movement sensor capable of opening an Internet of Things trash can's lid (Ramson & Moni, 2017; Rao et al., 2017).

b. Volume Hardware Implementation

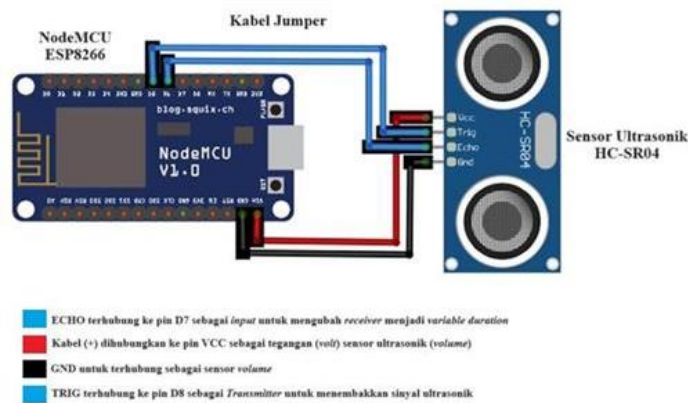


Figure 12: Ultrasonic Sensor with ESP8266 NodeMCU Circuit

The NodeMCU ESP8266 circuit and Ultrasonic Sensor in an Internet of Things-based garbage can capacity monitoring system are described in Figure 12. The ultrasonic sensor in this system will be connected to the NodeMCU ESP8266 utilizing a jumper cable connecting the NodeMCU ESP8266 pin to the ultrasonic sensor. This will be used to determine the garbage can's capacity volume of contents. The volume capacity of the trash can's contents will show the volume % via the blynk platform in real-time if the smartphone's Wi-Fi is enabled.

c. Circuit for an Infrared Sensor

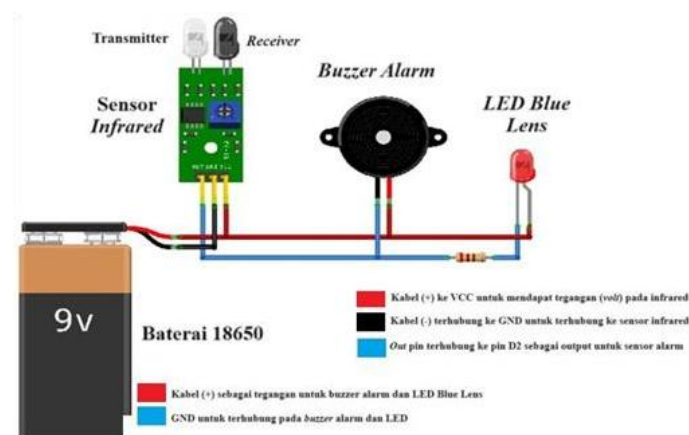


Figure 13: Circuit for an Infrared Sensor

The Infrared Sensor circuit used in an Internet of Things-based garbage can capacity monitoring system is described in Figure 13. The ultrasonic sensor's (volume sensor) process is connected to the infrared alarm sensor in this system. This works to supply voltage to each of the IR sensors, Buzzer, and LED by connecting all of their VCC pins to the (+) battery pole. Link the GND pin of the IR sensor to the (-) battery pole, and the OUT pin of the IR sensor to the (-) buzzer pole.

Attach one leg of the resistor to the (-) LED leg and the other leg to the IR Sensor's OUT pin. You must test whether everything is connected by placing your hand or an object next to the IR Sensor. In the event of success, the LED and buzzer will switch on. Therefore, under this method, the infrared sensor will beep to alert you to the need to empty the garbage can right away if its volume capacity reaches its maximum.

Programming using the Arduino IDE

The most crucial piece of software is the Arduino IDE, which connects complicated machine language to create a logical and understandable language. For the ESP8266 to be uploaded in compliance with the requests and orders given, it must first be initialized in the board's manager of the Arduino IDE before writing a program on it.

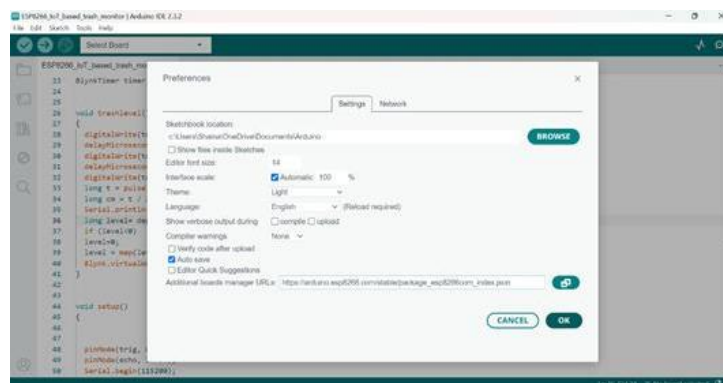


Figure 14. ESP8266 Configuration Display

The configuration process for the ESP8266 is illustrated in Figure 15. To get started, select File - Preferences, then click OK after entering the URL https://arduino.esp8266.com/stable/package_esp8266com_index.json in the additional board's manager.

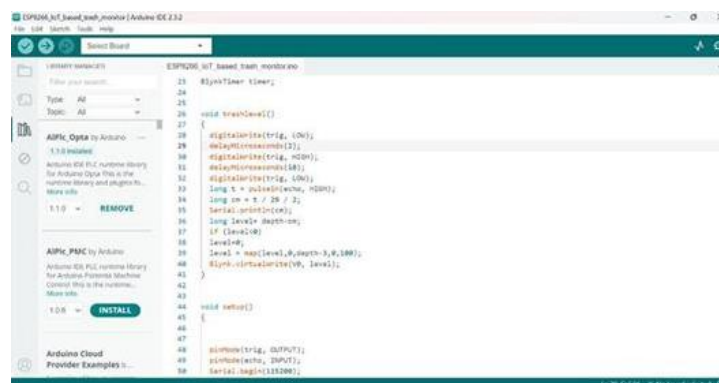


Figure 15. Installation View of the Blynk

Figure 15 shows how to install Blynk in the Library Manager after successfully installing ESP8266 on Arduino. This will enable it to be connected later on when developing applications that will be uploaded to NodeMCU.

settings after entering the token.

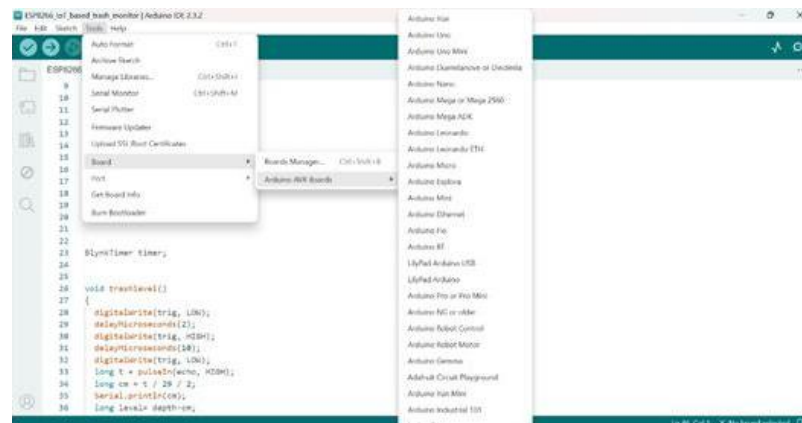


Figure 19. NodeMCU Board Configuration Display via Arduino IDE

Figure 19 indicates that, once all the required information has been entered (token, SSID, and password), synchronize the NodeMCU board with the Arduino IDE tools before uploading the configuration. To do this, click Tools in the Arduino IDE menu bar, scroll down, and click on the board labeled "Arduino / Genuino Uno." Next, select the NodeMCU 1.0 (ESP-12E Module) hardware module.

Blynk UI/UX

Blynk is an application that serves as a bridge between different NodeMCU-integrated devices. This enables users to communicate with each other over an internet connection and remotely control themselves (Patil & Patil, 2017; Popa et al., 2017). Blynk is a versatile platform/application.

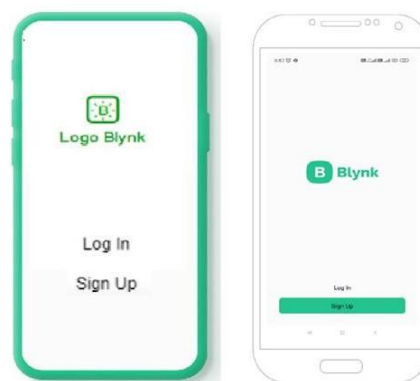


Figure 20: Blynk Application Mockup and UI/UX Installation View

Figure 20 illustrates that upon opening the Blynk program for the first time if you haven't already registered an email on the platform, you'll need to create an account to access the application. If you have, however, already registered an email on the platform, you can log in.

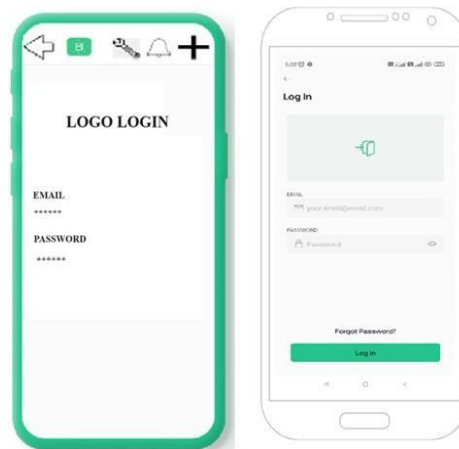


Figure 21: Blynk Application's Login Display Mockup and UI/UX

According to Figure 21, logging in with your registered email address and password is the next step if you have registered.



Figure 22: The Blynk application's mockup and UI/UX project display

The following step is seen in Figure 23, whereby a new Blynk project creation process requires the Auth token in the Blynk console. To access the Blynk console, use the email address that was used for registration. The code in the auth token will be used when we use the Arduino IDE to connect Blynk to NodeMCU.

Testing Modules

- a. The Arduino Uno R3 CH340G Module is being tested.

The purpose of testing the Arduino Uno R3 CH340G is to ascertain whether or not the board's input and output can function properly. Before installing a program on the Arduino Uno Board, testing is done. When the Arduino Uno is connected to a computer using a Unomega USB cable, input testing can be done on the device. The Arduino Uno can be used correctly if it can be read by the computer.

Using jumper cables to supply power to the board, one can test the Arduino Uno's output. The

Arduino Uno board can be used as intended if all of the outputs work.

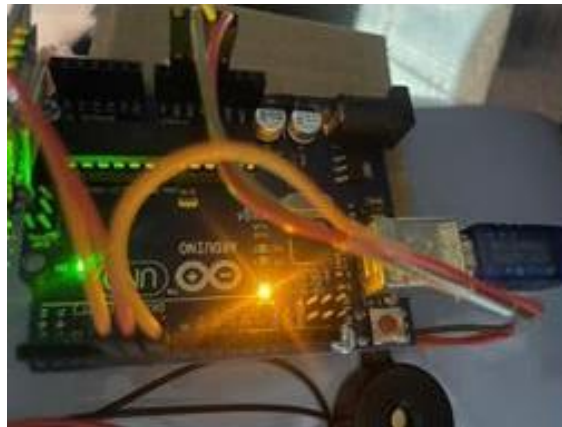


Figure 23. The Arduino Uno R3 CH340G Module is being tested.

Figure 23 describes how the Arduino Uno R3 CH340G module was initially linked to the laptop using a Unomega USB connection for testing. The module has a power indicator that turns on when it is successfully attached, meaning it is ready for usage. The Arduino Uno R3 CH340G module is then set up in the Arduino IDE to be recognized as a legitimate communication port. The microcontroller and laptop can establish a steady and dependable connection with the Arduino Uno R3 CH340G module, according to test results, guaranteeing a smooth and effective programming procedure.

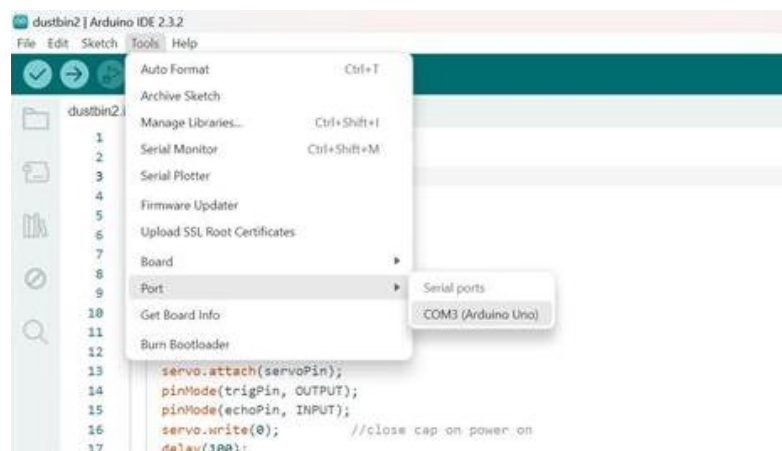


Figure 24. Examining the Connection Display of the Arduino Uno R3 CH340G Module

Figure 24 describes how the Arduino Uno R3 CH340G module was initially linked to the laptop using a USB Unomega cable for testing. The module has a power indicator that turns on when it is successfully attached, meaning it is ready for usage. The Arduino Uno R3 CH340G module is then set up in the Arduino IDE to be recognized as a legitimate communication port. According to test results, the Arduino Uno R3 CH340G module can establish a dependable and steady connection between the laptop and microcontroller, facilitating a seamless and effective programming procedure.

b. Testing of the NodeMCU ESP8266 Module



Figure 25. Testing the ESP8266 NodeMCU Module

A tool test utilizing the NodeMCU ESP8266 module is shown in Figure 26. After turning on the smartphone's Wi-Fi or hotspot, attach the micro USB cable to the NodeMCU ESP8266 to power it up. The Arduino IDE software must be flashed in order to provide commands via script code.

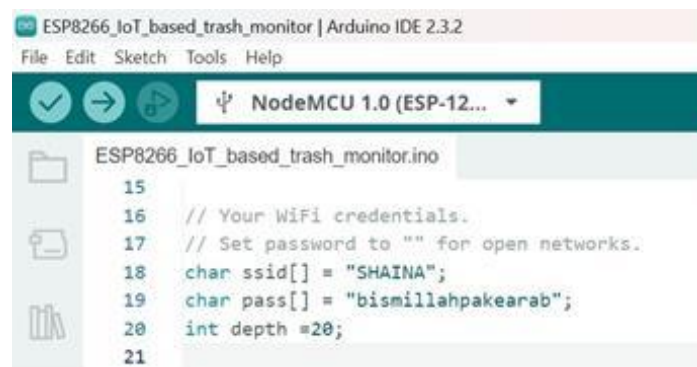


Figure 26. Testing the NodeMCU ESP8266 Wi-Fi Connection

The procedure for testing the NodeMCU ESP8266 module is shown in Figure 26, where the boot process is initiated by clicking the send button. The module sets the necessary pin configuration and passes through several configurations, including SPI_FAST_FLASH_BOOT, during the boot step. The module then attempts to connect to the Wi-Fi network that was previously set up after it has completed booting. The confirmation message "Wi-fi connected" will appear on the serial monitor if the Wi-fi connection process is successful. This shows that the internet connection has stabilized and the NodeMCU ESP8266 module has completed the Wi-Fi/Hotspot connection test.

Implementation of Volume Tool Testing

a. Tool Utilization When the Place Contents Rubbish Volume Is Determined by the Ultrasonic Sensor

Is an experiment to determine the volume of garbage can contents using an ultrasonic sensor instrument. A script code command on the NodeMCUE ESP8266 has been sent to the Arduino IDE to enable it to determine the trash can's volume capacity depending on its volume.

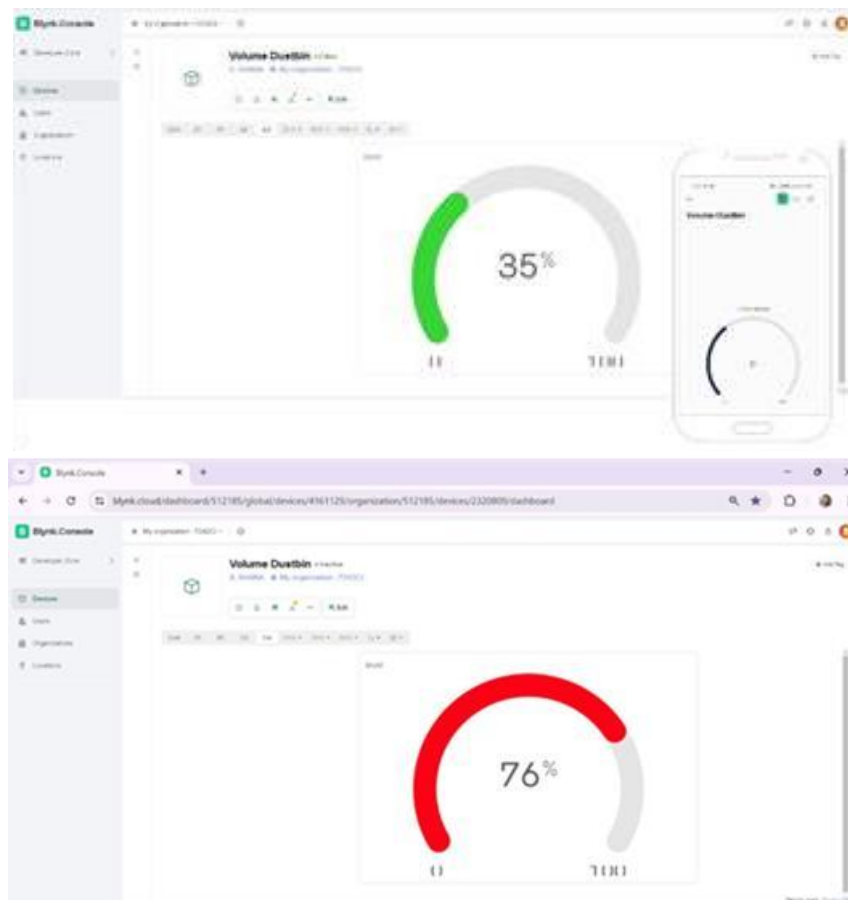


Figure 27. Blynk Platform After Volume Is Detected by Ultrasonic Sensor

The volume % display on the Blynk platform is tested in Figure 27. If the real-time volume percentage display is present. Thus, the Arduino IDE software has successfully flashed the NodeMCU ESP8266, and the ultrasonic sensor can determine the garbage can's volume capacity.

The trash can's volume capacity will then determine the output, which will be sent over the Blynk platform as a volume percentage.

CONCLUSION

Real-time waste volume monitoring has been accomplished by the developed and deployed Internet of Things-based trash bin capacity monitoring system(Murugaanandam et al., 2018). The volume of the bin's contents capacity may be detected thanks to the HC-SR04 ultrasonic sensor and NodeMCU ESP8266. The data is transmitted to the Blynk platform so that users can keep an eye on the bin's contents capacity using a laptop or mobile device. By using this tool, you can keep an eye on the trash can's capacity based on its volume and size.

It has been demonstrated that integrating IoT technology with residential trash cans improves waste management effectiveness(Memon et al., 2019; Monika et al., 2016). To prevent an excessive buildup of trash, a system that has an alarm sensor to alert the user when the trash can's capacity is

reached makes it easier for the user to know when the trash can is full and to remove its contents right away (Narayan et al., 2019). This helps to keep things clean and lowers the possibility of health issues because of collected garbage. With the use of this instrument, health-related issues can be resolved in real time by tracking the capacity of garbage bin contents (Navghane et al., 2016; Pardini et al., 2019).

REFERENCE

- Aazam, M., St-Hilaire, M., Lung, C. H., & Lambadaris, I. (2016). Cloud-based smart waste management for smart cities. In 2016 IEEE 21st International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD) (pp. 188–193). IEEE.
- Anagnostopoulos, T., Zaslavsky, A., & Medvedev, A. (2017). Robust waste collection exploiting cost efficiency of IoT potentiality in smart cities. 2017 IEEE International Conference on Recent Advances in Internet of Things (RIoT), 1–6.
- Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15), 2787–2805.
- Bharadwaj, A. S., Rego, R., & Chowdhury, A. (2016). IoT-based solid waste management system: A conceptual approach with an architectural solution as a smart city application. 2016 IEEE Annual India Conference (INDICON), 1–6.
- Bourougaa-Tria, S., Mokhati, F., Tria, H. E., & Bouziane, O. (2022). SPubBin: Smart Public Bin Based on Deep Learning Waste Classification: An IOT system for Smart Environment in Algeria. *Informatica (Slovenia)*, 46(8). <https://doi.org/10.31449/inf.v46i8.4331>
- Catania, V., & Ventura, D. (2014). An approach for monitoring and smart planning of urban solid waste management using the smart-M3 platform. *Proceedings of 15th Conference of Open Innovations Association FRUCT*, 24–31.
- Dubey, S., Anand, P., Kumar, Y., & Panwar, A. (2017). IoT-based smart garbage monitoring system using ESP8266 with GPS and GPRS. *International Journal of Advanced Research in Electronics and Communication Engineering*, 6(6), 566–570.
- Folianto, F., Low, Y. S., & Yeow, W. L. (2015). Smartbin: Smart waste management system. 2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 1–2.
- Gerard, M., Gupta, M., Sastry, R. K., & Srinivas, K. (2024). Patent Analytics of Internet of Things (IoT) based Technologies for Smart Greenhouses. *Journal of Intellectual Property Rights*, 29(4), 276–284. <https://doi.org/10.56042/jipr.v29i4.1158>
- Gherbia, A., & Zehani, S. (2024). Enhancing Security and Privacy Measures in Internet of Things (IoT) Implementations. *Proceedings - 8th IEEE International Conference on Image and Signal Processing and Their Applications, ISPA* 2024.

<https://doi.org/10.1109/ISPA59904.2024.10536814>

- Giacobbe, M., Puliafito, C., & Scarpa, M. (2018). The big bucket: An IoT cloud solution for smart waste management in smart cities. *European Conference on Service-Oriented and Cloud Computing*, 43–58.
- Gibson, P. A. (2023). Shortening the Supply Chain through Smart Manufacturing and Green Technology. *Sustainability*, 15(22). <https://doi.org/10.3390/su152215735>
- Hong, I., Park, S., Lee, B., Lee, J., Jeong, D., & Park, S. (2014). IoT-based smart garbage system for efficient food waste management. *The Scientific World Journal*.
- Idwan, S., Mahmood, I., Zubairi, J. A., & Matar, I. (2020). Optimal management of solid waste in smart cities using the Internet of Things. *Wireless Personal Communications*, 110(1), 485–501.
- Jha, A., Saha, S., & Paul, S. (2018). Smart dustbin management system: An approach towards a smart city. *International Journal of Scientific & Engineering Research*, 9(3), 1–4.
- Kale, N. S., & Lanjewar, D. S. (2017). IoT-based intelligent waste management system for smart city. *International Journal of Innovative Research in Computer and Communication Engineering*, 5(2), 1267–1273.
- Kang, K., Besklubova, S., Dai, Y., & Zhong, R. Y. (2022). Building demolition waste management through smart BIM: A case study in Hong Kong. *Waste Management*, 143. <https://doi.org/10.1016/j.wasman.2022.02.027>
- Kumar, N. S., Vuayalakshmi, B., Prarthana, R. J., & Shankar, A. (2016). IOT-based smart garbage alert system using Arduino UNO. *2016 IEEE Region 10 Conference (TENCON)*, 1028–1034.
- Lozano, Á., Caridad, J., Paz, J. F., Villarrubia González, G., & Bajo, J. (2018). Smart waste collection system with low consumption LoRaWAN nodes and route optimization. *Sensors*, 18(5), 1465.
- Malik, M., & Sumpena, S. (2023). Analisis Sistem Pemantauan Pemisah Sampah Logam dan Non Logam berbasis Internet of Things. *Jurnal Engine: Energi, Manufaktur, Dan Material*, 7(1). <https://doi.org/10.30588/jecemm.v7i1.1462>
- Medvedev, A., Fedchenkov, P., Zaslavsky, A., Anagnostopoulos, T., & Khoruzhnikov, S. (2015). Waste management as an IoT-enabled service in smart cities. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems* (pp. 104–115). Springer.
- Memon, S. K., Shaikh, F. K., Mahoto, N. A., & Memon, A. A. (2019). IoT-based smart garbage monitoring & collection system using WeMos & Ultrasonic sensors. *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (ICoMET)*, 1–6.
- Monika, K. A., Rao, N., Prapulla, S. B., & Shobha, G. (2016). Smart dustbin-an efficient garbage monitoring system. *International Journal of Engineering Science and Computing*, 6(6), 7113–7116.

- Murugaanandam, S., Ganapathy, V., & Balaji, R. (2018). Efficient IOT-based smart bin for a clean environment. 2018 International Conference on Communication and Signal Processing (ICCSP), 715–720.
- Narayan, S., Assaf, M. H., & Prasad, S. K. (2019). Wireless sensor-enabled public trash bins for smart cities. *Sensors*, 19(16), 3467.
- Navghane, S. S., Killedar, M. S., & Rohokale, V. M. (2016). IoT-based smart garbage and waste collection bin. *International Journal of Advanced Research in Electronics and Communication Engineering*, 5(5), 1576–1578.
- Pardini, K., Rodrigues, J. J., Kozlov, S. A., Kumar, N., & Furtado, V. (2019). IoT-based solid waste management solutions: A survey. *Journal of Sensor and Actuator Networks*, 8(1), 5.
- Parkash, P. V., & Rajendra, S. (2016). IoT-based intelligent bin for smart cities. *International Journal of Scientific & Engineering Research*, 7(4), 105–110.
- Patil, S. L., & Patil, A. N. (2017). Monitoring of garbage bins using IoT. *International Journal of Innovative Research in Computer and Communication Engineering*, 5(4), 7843–7846.
- Popa, C. L., Carutasu, G., Cotet, C. E., Carutasu, N. L., & Dobrescu, T. (2017). Smart city platform development for an automated waste collection system. *Sustainability*, 9(11).
- Ramson, S. J., & Moni, D. J. (2017). Wireless sensor networks based smart bin. *Computers & Electrical Engineering*, 64, 337–353.
- Rao, S. K., Reddy, A., & Kumar, D. S. (2017). IoT-based smart garbage monitoring system. *International Journal of Scientific Development and Research*, 2(5), 452–455.
- Rath, M., Samantaray, B., Sethhi, B., & Panda, A. (2018). IoT-based smart garbage monitoring system. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 3(5), 1214–1219.
- Saha, H. N., Auddy, S., Pal, S., Kumar, S., Pandey, S., Singh, R., & Maity, T. (2017). Waste management using the Internet of Things (IoT). 2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON), 359–363.
- Shamin, N., Samal, P., Chakraborty, S., & Sardana, H. K. (2019). Development of IoT-based smart waste management system: An effective way to monitor waste segregation and collection. 2019 International Conference on Communication and Signal Processing (ICCSP), 957–960.
- Sheng, T. J., Islam, M. S., Misran, N., Baharuddin, M. H., Arshad, H., Islam, M. R., & Islam, M. T. (2020). An Internet of Things-based smart waste management system using LoRa and TensorFlow deep learning model. *IEEE Access*, 8, 148793–148811.
- Srinivasan, S., & Latha, R. (2019). RFID and IoT-based smart waste management system. 2019 International Conference on Communication and Signal Processing (ICCSP), 842–846.
- Thakker, S., & Narayanamoorthi, R. (2015). Smart and wireless waste management. 2015 International Conference on Innovations in Information, Embedded and Communication

Systems (ICIIIECS, 1–4.

Tiwari, A., & Dixit, A. (2017). Garbage collection management system. *International Journal of Computer Science and Mobile Computing*, 6(6), 7–11.

Vinithra, S., & Bharathi, A. (2018). Smart bin monitoring system using IoT. 2018 *International Conference on Communication and Signal Processing (ICCSP)*, 1026–1029.

Wijaya, A. S., Zainuddin, Z., & Niswar, M. (2017). Design a smart waste bin for smart waste management. 2017 *5th International Conference on Instrumentation, Control, and Automation (ICA)*, 62–66.

Zavare, S., Parashar, R., Patil, S., Rathod, P., & Babanne, V. (2017). Smart city waste management system using GSM. *International Journal of Computer Science Trends and Technology*, 5(3), 74–78.