



Design and Development of an IoT-Based Automatic Water Gallon Filling System Using a YF-S201 Flow Sensor, ESP32, and Telegram Bot

Singgih Astaguna Tiroatmaja¹, Akhyar Muchtar^{2*}, Andi Ahmad Taufiq³, Muhammad Fauzan Yusuf⁴

¹Department of Electrical Engineering, Universitas Negeri Makassar, Indonesia

¹²³⁴Universitas Negeri Makassar

Correspondence E-mail: akhyarmuchtar03@unm.ac.id.

ABSTRACT

Manual refilling of water gallons at refill drinking-water depots relies on an operator switching the pump on and off, which can cause inaccurate volumes, water wastage, and dependence on the operator. This study designed and built an Internet of Things (IoT)-based automatic gallon-filling system using a YF-S201 Hall-effect flow sensor, an ESP32 microcontroller, a 5 V relay module that switches the AC 220 V depot pump, and a Telegram Bot as a remote user interface. The firmware counts sensor pulses through a hardware interrupt and updates the volume every second in a non-blocking `millis()`-based loop. Two protection layers stop the pump when no flow is detected for 10 s or when the volume reaches 19.0 L, and only a registered Telegram chat ID can issue the `/isi_galon`, `/stop`, and `/status` commands. Volume accuracy was tested in 40 trials with a 19,000 mL target. The pump stopped automatically in every trial, but all trials overfilled: the filled volume ranged from 19,320 to 20,080 mL, with a mean overfill of 612.75 mL and a mean relative error of 3.225% (1.684–5.684%). The overfill is attributed to relay switching delay, flow inertia, and flow-sensor tolerance, so sensor calibration and overshoot compensation are needed.

ARTICLE INFO

Article History:

Submitted/Received 13 Oct 2026

First Revised 13 Oct 2026

Accepted 21 Oct 2026

First Available online 22 Oct 2026

Publication Date 22 Oct 2026

Keyword:

Automation,
Internet of Things,
Flow sensor,
YF-S201,
ESP32,
Telegram Bot,
Refill drinking water.

1. INTRODUCTION

Many electronic devices now apply automatic control to improve operational efficiency and ease of use. Advances in information and communication technology and the Internet of Things (IoT) allow real-time monitoring and control of water infrastructure and give users data and tools to manage water use more effectively [1], and IoT-based real-time monitoring has also been applied to three-phase electrical power systems [2]. Control-system technology is also increasingly introduced in vocational education [3]. One application of automatic control is the filling of refill drinking water. In practice, however, many refill drinking-water depots still fill gallons manually, relying on a switch to turn the water pump on and off.

In a manual filling system, control and monitoring depend on the operator. This can cause problems such as overfilling, inaccurate volumes, and wasted water. To address these problems, several automatic filling systems for refill depots have been proposed, including an Arduino Uno-based automation prototype for refill drinking-water depots [4], a data logger for automatic gallon filling based on an ATmega32 microcontroller [5], and an IoT-based control and monitoring system for automatic refill filling using an ESP32 [6]. Automatic gallon-filling systems have also been introduced to refill-depot operators through community-service mentoring [7].

One way to control the filled volume is to measure the water entering the gallon with a flow sensor. A Hall-effect sensor produces an output voltage in response to a magnetic field [8], and in a turbine-type flow meter the rotation of the rotor is converted into pulses whose frequency depends on the flow rate [9]. The pulse count of a low-cost Hall-effect flow-rate sensor has been shown to correlate with the dispensed volume, so the volume can be estimated from the pulse count after calibration [10]. The YF-S201 sensor works on this principle and can therefore be used to calculate the volume of water in real time.

For the controller, the ESP32 is a low-cost microcontroller module with integrated Wi-Fi and Bluetooth [11], [12], and its general-purpose input/output pins make it suitable for IoT devices that connect sensors and actuators [12]. A remote user interface can be provided through a Telegram bot: the Telegram Bot API is an HTTP-based interface for building bots, and every incoming message identifies the chat it belongs to [13], so commands can be restricted to registered users.

Previous work has developed IoT-based control and monitoring of automatic refill filling using an ESP32 [6]. This study follows that direction by combining flow-sensor-based volume control with a Telegram bot interface. The firmware counts sensor pulses through a hardware interrupt and runs a non-blocking main loop, provides two protection layers (no-flow detection and a maximum-volume limit), and accepts the `/isi_galon`, `/stop`, and `/status` commands only from a registered chat ID.

Therefore, this study aims to design and build an IoT-based automatic gallon-filling system using a YF-S201 flow sensor, an ESP32 microcontroller, a relay, and a Telegram bot, and to evaluate its filling-volume accuracy to improve efficiency, accuracy, and reliability in filling refill drinking water.

2. METHODS

This section describes the design and testing stages of the automatic gallon-filling system. The stages comprise system design, hardware design, software design, and system testing.

2.1. System Design

The IoT-based automatic gallon-filling system integrates several main components that work together to automatically control the filling process, as shown in **Figure 1**. The YF-S201

flow sensor is the input, the ESP32 is the main controller, the 5 V relay module is the output that switches the AC 220 V depot water pump, and the Telegram Bot API is the IoT interface through which the user sends commands and receives notifications. Each component has a specific function in the system, covering input, processing, and output.

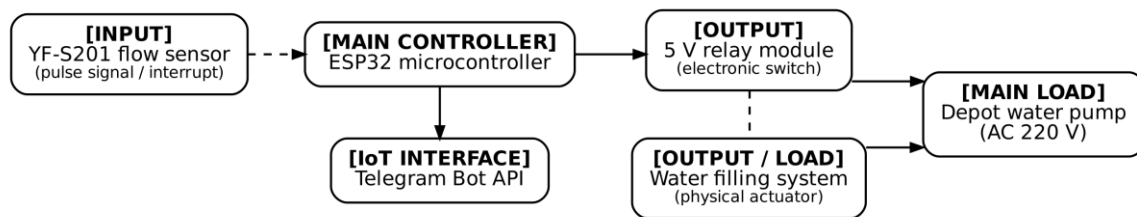


Figure 1. Block diagram of the IoT-based automatic water gallon filling system

2.2. YF-S201 Flow Sensor

The YF-S201 water flow sensor **Figure 2** is the main component for detecting the water flowing into the gallon. The sensor works on the Hall-effect principle [8]: it contains a small turbine that rotates when water flows, and this rotation produces electrical pulses that are sent to the microcontroller [9]. Each pulse is directly related to the amount of water flowing. By counting the pulses per unit time, the microcontroller determines the flow rate and volume of water [10]. The sensor has three pins: VCC, GND, and signal output. In this system, the sensor is the main input that determines when filling must stop based on the volume reached. The YF-S201 has been used together with an ESP32 in other water-monitoring systems and measures flow rates of 1–30 L/min [14].



Figure 2. YF-S201 flow sensor

2.3. Microcontroller

The ESP32 microcontroller **Figure 3** acts as the control center of the system. The ESP32 is a low-cost microcontroller module with integrated Wi-Fi and Bluetooth [11], [12]. It receives the pulse signal from the flow sensor and processes the data to calculate the volume of water in real time. The microcontroller also (i) converts the pulse data into a volume value, (ii) compares the volume with the 19.0 L limit, (iii) sends the control signal to the relay, and (iv) communicates with the Telegram Bot API over the WiFi network. The microcontroller is therefore the brain of the system that governs the entire automation process.



Figure 3. ESP32 microcontroller

2.4. Relay

The 5 V relay module in **Figure 4** replaces the manual switch. It works as an electronic switch controlled by the signal from the microcontroller. When the microcontroller outputs a HIGH signal, the relay is activated and connects the AC 220 V supply to the water pump, so the pump turns on. When a LOW signal is given, the relay cuts the power, and the pump stops. In this system, the relay is the main control output that determines whether the pump is on or off.



Figure 4. Relay module

2.5. Water Pump

The water pump in **Figure 5** is the actuator that moves water from the source into the gallon. The pump is controlled by the relay rather than directly by the user. When the relay is active, the pump runs and starts filling the gallon; when the relay is off, the pump stops automatically. The depot water pump operates on AC 220 V. The pump is therefore the execution component of the system.



Figure 5. Water pump

2.6. Power Supply

The power supply in **Figure 6** provides electrical energy to the system. The ESP32 microcontroller and the 5 V relay module are powered by a 5 V DC power adapter, while the depot water pump is powered from the AC 220 V mains through the relay contacts in **Figure 1**. The power supply must match the requirements of each component so that the system operates stably and safely.



Figure 6. Power supply

2.7. Software Algorithm and Flowchart

Overall, the software algorithm for this Internet of Things (IoT)-based automatic water gallon-filling system is divided into three main process blocks: Initialization and Interrupt, Main Logic (Volume Calculation and Protection), and Telegram Message Handling Logic. The system flowchart in **Figure 7** is designed so that the microcontroller can run several tasks in a non-blocking manner (without waiting for one another), allowing sensor readings and responses to user commands to run in real time.

The system begins its operating cycle by executing the `Inisialisasi()` procedure. At this stage, the microcontroller performs the initial setup, which includes enabling serial communication, configuring the relay pin as an output (with an initial OFF/LOW state), and configuring the water flow sensor pin as an input with pull-up. In addition, the system connects to the WiFi network and authenticates with the Telegram Bot API server.

To ensure that no sensor readings are missed while the microcontroller executes other lines of code, the system uses a hardware interrupt on the sensor pin. Whenever a falling-edge signal is detected from the water flow sensor, the microcontroller automatically calls the `Ambil_Pulsa_Sensor()` procedure. This procedure runs independently in the background and increments the `hitungan_pulsa` variable by 1.

Once initialization is complete, the microcontroller enters an infinite loop (main loop). In this block, the system tracks time using the `millis()` function (time in milliseconds) to avoid using the `delay` function, which would halt system operation.

If filling is active and the time interval has reached 1000 milliseconds (1 second), the system briefly disables interrupts to protect the data, then reads and resets the value of `hitungan_pulsa`. The pulse count is converted into a water flow rate using the sensor calibration factor [15]-[17], the volume delivered during that second is calculated, and this volume is accumulated into `Total_Volume`, as expressed in Equations (1)–(3).

$$Q = N / K \quad (1)$$

$$\Delta V = Q / 60 \quad (2)$$

$$V_total = V_total + \Delta V \quad (3)$$

where N is the number of pulses counted in the 1-second interval (*hitungan_pulsa*), K is the sensor calibration factor (*faktor_kalibrasi*), Q is the water flow rate in L/min (*debit_air*), ΔV is the volume delivered during that second in litres (*volume_detik_ini*), and *V_total* is the accumulated volume (*Total_Volume*). (State the value of K used in the firmware here, e.g. K = ... pulses per second per L/min.)

To ensure operational safety, the algorithm is equipped with two layers of protection. The first protection is clogged-flow / empty-water-source detection. The system monitors the time between sensor pulse updates. If no water flow is detected within 10 seconds (*waktu_sekarang* – *waktu_terakhir_aliran* $\geq$ 10000 ms), the system assumes that the water source has run out or that a malfunction has occurred. The system then automatically turns off the pump (relay OFF) and sends an emergency warning message to the user's Telegram.

The second protection is the maximum volume limit. The system continuously compares *Total_Volume* with the standard gallon capacity. When the volume of water delivered reaches or exceeds 19.0 litres, the system stops the pump, sets the filling status to inactive, and sends a Telegram notification that the filling process has finished.

The Telegram Message Logic block serves as a remote user interface (UI) [18], [19]. The system periodically checks whether a new command message has arrived from the Telegram Bot. If there is a message, the system first performs a security check by matching *ID_Pengirim* against the registered *CHAT_ID_Valid*. If the ID is not recognized, the system denies access.

If the ID is validated, the system processes the command according to the content of the message.

- **/isi_galon** resets *Total_Volume* to zero, turns on the pump, sets the status to active, and restarts the protection timer. If the system is already active, it ignores the command and sends a warning message that filling is already in progress.
- **/stop** acts as a remote emergency stop. It forcibly turns off the pump at any time, changes the filling status, and notifies the user.
- **/status** asks the microcontroller to send feedback on the pump status (active/inactive) and the current volume of water that has entered the gallon.

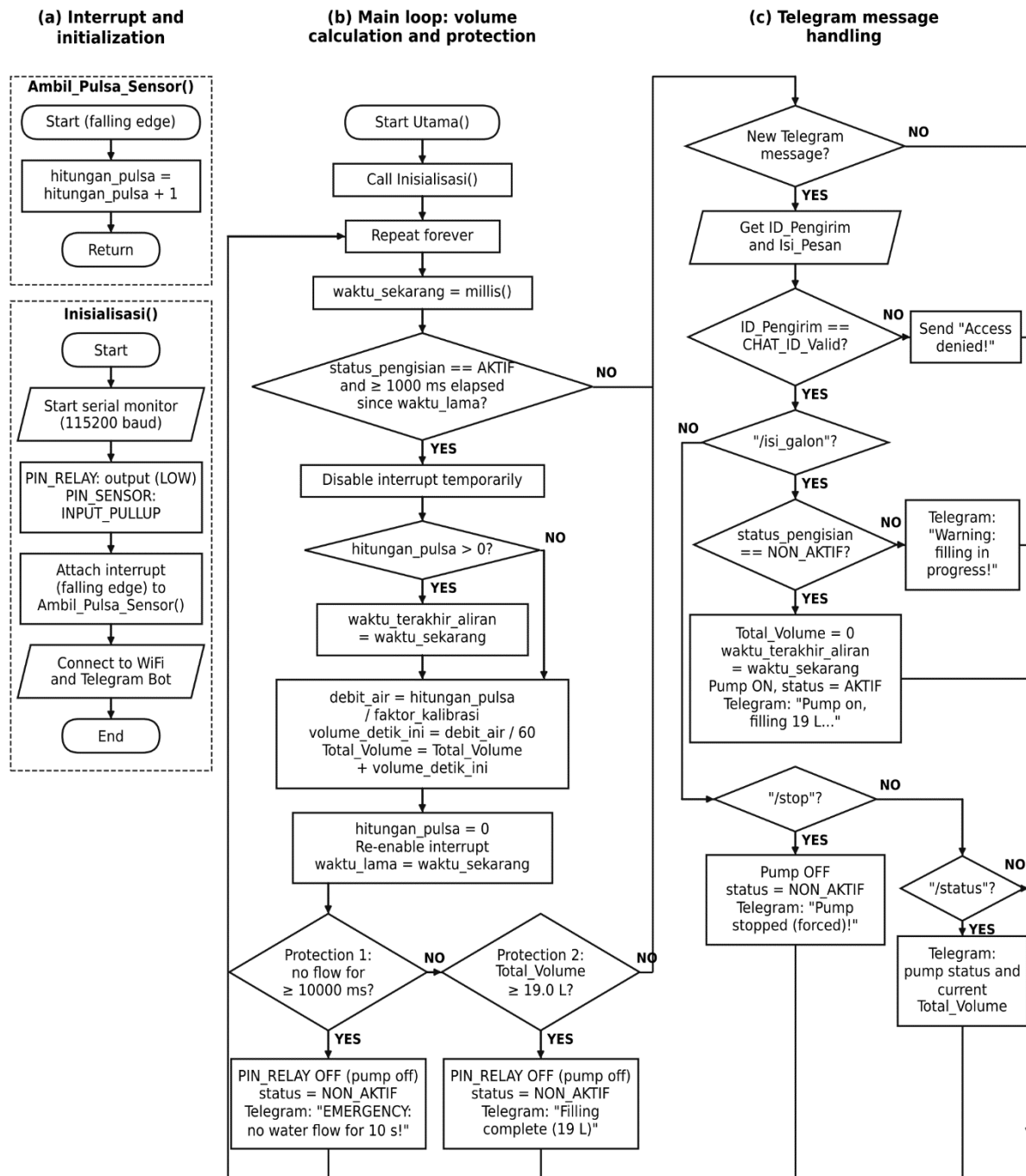


Figure 7. Flowchart of the software algorithm: (a) interrupt and initialization procedures, (b) main loop with volume calculation and protection, and (c) Telegram message handling. Connectors A and B link the panels

2.8. Wiring Design

The wiring design of the automatic water gallon filling system is shown in **Figure 8**.

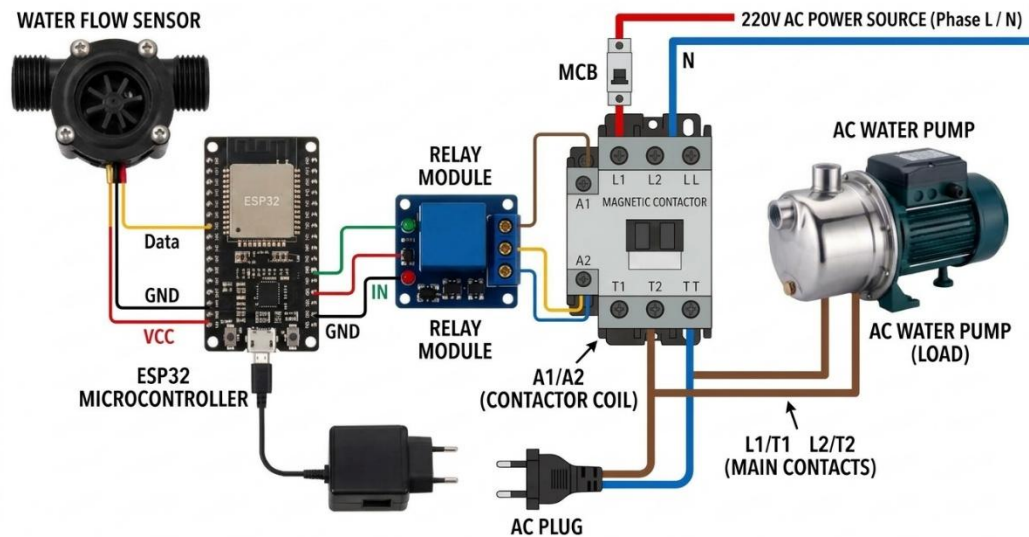


Figure 8. Wiring diagram of the automatic water gallon filling system.

2.9. LabVIEW Design

The automatic water gallon filling system was also designed using the LabVIEW application. **Figure 9** shows the front panel design of the control system, and **Figure 10** shows its block diagram.



Figure 9. Front panel design of the automatic water gallon filling control system

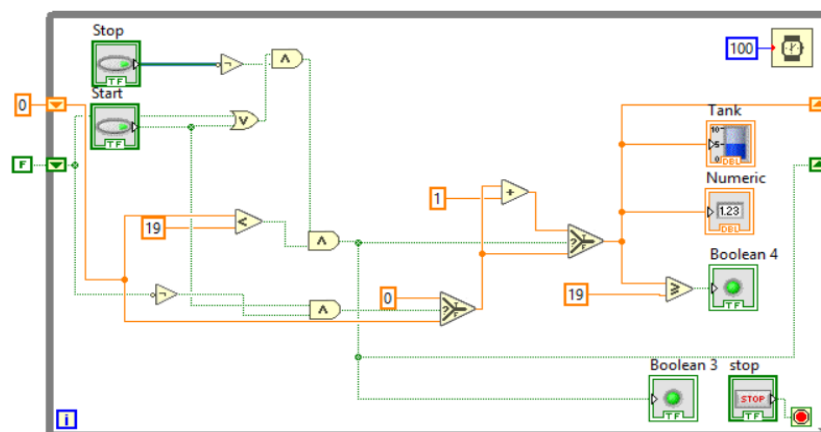


Figure 10. Block diagram of the automatic water gallon filling system

2.10. Testing Procedure

The system was tested in 40 trials with a target volume of 19,000 mL (19 L). For each trial, the filled volume was recorded, the difference from the target was calculated, and the relative error was computed using Equation (4). The test reported in this study evaluates the filling-volume accuracy of the system.

$$\text{Error (\%)} = (V_{\text{filled}} - V_{\text{target}}) / V_{\text{target}} \times 100\% \quad (4)$$

where V_{filled} is the volume of water filled into the gallon (mL), and V_{target} is the target volume (19,000 mL).

3. RESULTS AND DISCUSSION

In the tested system, the flow sensor detects the water flow, and the microcontroller calculates the volume from the number of pulses produced. The relay acts as the actuator switch that turns the water pump on and off automatically. During filling, the flow sensor reads the water flow in real time and sends the data to the microcontroller. When the calculated volume reaches the setpoint, the system commands the relay to switch off so that the pump stops. In practice, however, the filled volume still exceeded the target. The test results are presented in **Table 1** and **Figure 11**.

Table 1. Results of the automatic water gallon filling test (target volume 19,000 mL)

No.	Target (mL)	Filled (mL)	Difference (mL)	Error (%)	No.	Target (mL)	Filled (mL)	Difference (mL)	Error (%)
1	19,000	19,420	420	2.211	21	19,000	19,400	400	2.105
2	19,000	19,980	980	5.158	22	19,000	19,370	370	1.947
3	19,000	19,350	350	1.842	23	19,000	19,450	450	2.368
4	19,000	20,020	1,020	5.368	24	19,000	19,500	500	2.632
5	19,000	19,950	950	5.000	25	19,000	19,410	410	2.158
6	19,000	19,480	480	2.526	26	19,000	20,040	1,040	5.474
7	19,000	19,390	390	2.053	27	19,000	19,960	960	5.053
8	19,000	19,510	510	2.684	28	19,000	20,000	1,000	5.263
9	19,000	19,460	460	2.421	29	19,000	19,340	340	1.789
10	19,000	19,380	380	2.000	30	19,000	19,460	460	2.421
11	19,000	20,050	1,050	5.526	31	19,000	19,390	390	2.053
12	19,000	19,430	430	2.263	32	19,000	19,420	420	2.211
13	19,000	19,990	990	5.211	33	19,000	19,480	480	2.526
14	19,000	19,320	320	1.684	34	19,000	19,520	520	2.737
15	19,000	20,010	1,010	5.316	35	19,000	19,430	430	2.263
16	19,000	19,470	470	2.474	36	19,000	19,370	370	1.947
17	19,000	19,360	360	1.895	37	19,000	19,490	490	2.579
18	19,000	20,030	1,030	5.421	38	19,000	19,510	510	2.684
19	19,000	19,970	970	5.105	39	19,000	20,080	1,080	5.684
20	19,000	19,920	920	4.842	40	19,000	19,400	400	2.105
Mean of 40 trials (n = 40)								612.75	3.225

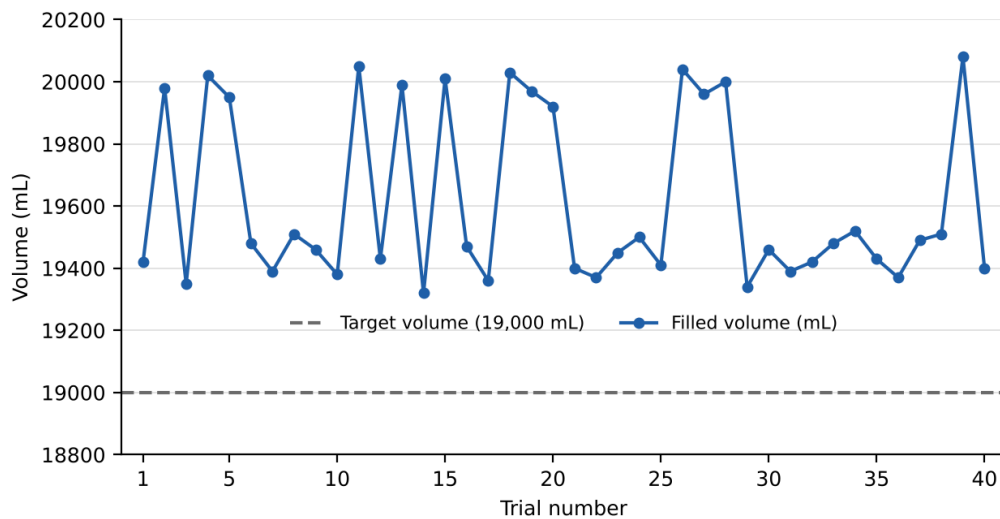


Figure 11. Filled volume in each of the 40 trials compared with the 19,000 mL target

Table 1 shows that every filling produced a volume larger than the target, so the system consistently overfilled in all 40 trials. The filled volume ranged from 19,320 to 20,080 mL, with a mean of 19,612.75 mL (standard deviation 277.19 mL). The overfill ranged from 320 to 1,080 mL, with a mean of 612.75 mL. Based on Equation (4), the minimum error was 1.684%, the maximum error was 5.684%, and the mean error was 3.225%.

The results also form two distinct groups. In 27 trials, the filled volume was between 19,320 and 19,520 mL (overfill 320–520 mL), while in 13 trials it was between 19,920 and 20,080 mL (overfill 920–1,080 mL). This indicates that the overfill is not a single constant offset, and that the cause of the larger overfill in some trials needs further investigation.

The overfill in every trial is attributed to several factors: a delay when the relay switches off, the inertia of the water flow in the hose, and the tolerance of the flow sensor in reading pulses. The conversion from pulse count to volume in low-cost Hall-effect flow sensors depends on calibration [10], [15]–[17], so an uncalibrated conversion factor can contribute to a systematic volume error. In addition, the system does not stop instantly, so some water continues to flow after the pump is switched off. These factors were not measured separately in this study and should be quantified in future work.

Overall, the automatic gallon-filling system based on a flow sensor and a relay was able to stop filling automatically in all 40 trials. However, with a mean error of 3.225% and an overfill in every trial, its filling precision still needs to be improved. Further development is required, such as a faster flow-stopping mechanism, more precise sensor calibration, or compensation of the setpoint for the measured overfill. In addition, the response time of the Telegram commands and the behaviour of the no-flow protection were not evaluated quantitatively in this study and should be tested in future work.

4. CONCLUSION

Based on the design, implementation, and testing of the automatic gallon-filling system, the IoT-based system built using a YF-S201 flow sensor, an ESP32 microcontroller, a relay as the pump controller, and a Telegram bot as the remote interface was able to fill water automatically. The firmware counts the sensor pulses through a hardware interrupt, calculates the volume in a non-blocking main loop, provides two protection layers (no-flow

detection after 10 s and a 19.0 L volume limit), and accepts the /isi_galon, /stop, and /status commands only from a registered Telegram chat ID.

Testing in 40 trials showed that the system stopped the pump automatically in every trial, with a minimum error of 1.684%, a maximum error of 5.684%, and a mean error of 3.225%. All trials overfilled the 19 L target, with a mean overfill of 612.75 mL, and the results formed two groups with overfills of 320–520 mL and 920–1,080 mL.

The overfill is attributed to a delay when the relay switches off, the inertia of the water flow in the hose [20], and the reading tolerance of the flow sensor; in addition, the water flow does not stop instantly, so extra volume is added after the system switches off the pump.

Overall, the designed system meets the research objective of automating the water-filling process with remote control and notification through Telegram. It can serve as a practical solution to improve efficiency and reduce dependence on manual operation, although further development is still needed: calibrating the flow sensor, compensating for the overshoot to improve filling precision, and quantitatively evaluating the Telegram command response and the no-flow protection.

5. AUTHORS' NOTE

The authors declare that there is no conflict of interest regarding the publication of this article. The authors confirmed that the paper was free of plagiarism.

6. REFERENCES

- [1] Palermo, S. A., Maiolo, M., Brusco, A. C., Turco, M., Pirouz, B., Greco, E., Spezzano, G., and Piro, P. (2022). Smart technologies for water resource management: An overview. *Sensors*, 22(16), 6225. <https://doi.org/10.3390/s22166225>
- [2] Putra, N. A. I. E., Fitriati, A., Muchtar, A., and Fajar, N. (2025). Development of a real time three phase power parameter monitoring system using programmable logic controller (PLC) and Internet of Think (IoT). *Jurnal Media Elektrik*, 23(1), 81-88.
- [3] Muchtar, A., Burhan, M. I., Sari, D. A. L., Makmur, E., and Ashari, H. (2024). Pelatihan pemanfaatan media pembelajaran sistem kontrol berbasis elektro pneumatik di SMKN 9 Makassar. *TEKIBA: Jurnal Teknologi dan Pengabdian Masyarakat*, 4(2), 242-248.
- [4] Arfandi, A., and Supit, Y. (2019). Prototipe sistem otomasi pada pengisian depot air minum isi ulang berbasis Arduino Uno. *Simtek: Jurnal Sistem Informasi dan Teknik Komputer*, 4(1), 91-99. <https://doi.org/10.51876/simtek.v4i1.53>
- [5] Supandi, S., Hilda, H., and Hadary, F. (2017). Perancangan sistem data logger pengisian air galon otomatis berbasis mikrokontroler ATmega32. *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, 3(1), 1-6. <https://doi.org/10.26418/jp.v3i1.19239>
- [6] Pamungkas, D. D., and Setyadjit, K. (2023). Rancang bangun sistem kontrol dan monitoring pengisian air minum isi ulang otomatis menggunakan ESP32 berbasis IoT. *Prosiding Senakama*, 2(1), 652-662.
- [7] Muchtar, A., Makmur, E., Ashari, H., Burhan, M. I., and Riska, M. (2026). Pendampingan pemanfaatan sistem pengisian galon otomatis bagi depot air minum isi ulang. *Jurnal Pengabdian Masyarakat*, 4(1), 150-155.
- [8] Paun, M.-A., Sallese, J.-M., and Kayal, M. (2013). Hall effect sensors design, integration and behavior analysis. *Journal of Sensor and Actuator Networks*, 2(1), 85-97. <https://doi.org/10.3390/jsan2010085>

- [9] Baker, R. C. (2016). *Flow measurement handbook: Industrial designs, operating principles, performance, and applications* (2nd ed.). Cambridge University Press. <https://doi.org/10.1017/CBO9781107054141>
- [10] Eronu, E. M., and Odiagbe, M. (2020). Evaluation and modelling of a low budget Hall effect based flow-rate sensor using adaptive calibration paradigm. *European Journal of Engineering and Technology Research*, 5(9). <https://doi.org/10.24018/ejeng.2020.5.9.2154>
- [11] Maier, A., Sharp, A., and Vagapov, Y. (2017). Comparative analysis and practical implementation of the ESP32 microcontroller module for the Internet of Things. In *2017 Internet Technologies and Applications (ITA)*. IEEE. <https://doi.org/10.1109/ITECHA.2017.8101926>
- [12] Hercog, D., Lerher, T., Truntič, M., and Težak, O. (2023). Design and implementation of ESP32-based IoT devices. *Sensors*, 23(15), 6739. <https://doi.org/10.3390/s23156739>
- [13] Telegram. (n.d.). *Telegram Bot API*. Retrieved October 13, 2026, from <https://core.telegram.org/bots/api>
- [14] Öztürk, M. A., Ünsal, E., and Yelkuvan, A. F. (2025). Development of an Internet of Things-based ultra-pure water quality monitoring system. *Sensors*, 25(4), 1186. <https://doi.org/10.3390/s25041186>
- [15] Lalnunthari, J., and Thanga, H. (2017). Dependence of Hall effect flow sensor frequency on the attached inlet and outlet pipe size. In *2017 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, 56–60. IEEE. <https://doi.org/10.1109/ICCE-ASIA.2017.8307842>
- [16] Riupassa, R. D., Rafliis, H., and Hendro. (2018). Optimasi nilai konstanta kalibrasi pada water flow sensor YF-S201. *Sistemik: Jurnal Ilmiah Nasional Bidang Ilmu Teknik*, 6, 1–5. <https://ejournal.utb-univ.ac.id/Ejournal/JurnalDetail/523>
- [17] Rashad, A. A., Ulikaryani, Girawan, B. A., Ariawan, R., Sodikin, J., and Hastuti, H. D. (2023). Uji fungsi dan kalibrasi sensor water flow YF-S201 berbasis Arduino Uno pada mesin penjernih air sungai. *Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, 3(2), 1–9. <https://doi.org/10.55606/teknik.v3i2.1850>
- [18] Tisna, D. R., Maharani, T., and Nugroho, K. T. (2024). Pemanfaatan chatbot Telegram untuk monitoring dan kontrol kualitas air menggunakan ESP32. *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 9(3), 1292–1306. <https://doi.org/10.29100/jipi.v9i3.5329>
- [19] Setyawan, N. R., Sumarahinsih, A., and Setiawan, A. B. (2026). Penerapan IoT pada sistem monitoring dan otomatisasi pengoperasian pompa air kabel duct gardu induk Tanggul. *METROTECH (Journal of Mechanical and Electrical Technology)*, 5(2), 253–261. <https://ejournal.uniramalang.ac.id/metrotech/article/view/8711>
- [20] Ghidaoui, M. S., Zhao, M., McInnis, D. A., and Axworthy, D. H. (2005). A review of water hammer theory and practice. *Applied Mechanics Reviews*, 58(1), 49–76. <https://doi.org/10.1115/1.1828050>