PAPER • **OPEN ACCESS**

Development of an Automatic Water Flow Sensor System Using ESP32 for Efficient Water Control

To cite this article: A Majid *et al* 2023 *IOP Conf. Ser.: Earth Environ. Sci.* **1242** 012016

Development of an Automatic Water Flow Sensor System Using ESP32 for Efficient Water Control

A Majid[1], Jamaaluddin[2*], A Wiguna[3], H Setiawan[4], A Farihah[5]

[1,2] Electrical Engineering Study Department, Universitas Muhammadiyah Sidoarjo, Indonesia
[3] Information Technology Education Department, Universitas Muhammadiyah Sidoarjo, Indonesia
[4, 5] Information Technology Department, Universitas Muhammadiyah Sidoarjo, Indonesia

* Corresponding Author's Email: jamaaluddin@umsida.ac.id

Abstract. Access to clean and reliable drinking water is essential for human well-being. This research aims to develop an automatic water flow sensor system based on ESP32 microcontroller for efficient water control. The study employs the Development (Observation) method to observe and record the activities involved in the system development process. The system allows users to adjust the water discharge according to their requirements by utilizing a keypad interface and modifying the Arduino IDE code. The experimental results demonstrate a high accuracy of 92% with an 8% error rate, indicating the system's reliability in controlling the water flow. Moreover, stabilizing the ESP32 microcontroller by providing a voltage of at least 5V ensures optimal performance. The findings suggest the practicality and cost-effectiveness of the developed system, which can contribute to improving water management and conservation efforts worldwide.
Keywords: ESP32, Water Flow Sensor, Water Controller, Plastic Bottle

1. Introduction

Drinking water worthy of consumption is a goal and necessary for daily life, clean water serves as a basic human need [1]. The air we usually use comes from various sources, including lake water, sea water, and well water. The quality of air from various sources varies greatly in both quality and quantity[2]. As for what we usually consume as drinking water every day is mountain water that has been processed through technology to be packaged in such a way that it can be consumed properly by humans [3].

The human body consists of water, and water is basically the main need for humans, so as much as 80% in infants, as much as 60% in adults, and as much as 50% in people over 65 years old or elderly [4][5]. Since water is a very important nutrient for human health and can be used as a solvent, catalyst, lubricant, and body temperature regulator, humans must always move. The body will become healthier by consuming more water because according to medical experts, humans should consume plenty of water[6].

Even so, it is unfortunate that there are not always refillable water stations available in public places, instead, they take a long time to diminish. Some people who have expressed concern about the adverse effects of repeated use of plastic bottles are likely to use tumblers when drinking, but the average bottle distributed is only 600 mL, which only meets about 35% of the total water requirement in the human body[7]. Since it is very difficult to find stations that sell pressurized refillable water in regular locations, when the air inside the tumbler reaches the completion point, it will be possible to start over. This encourages people to buy drinking water that uses plastic bottle caps repeatedly[8].

Every time plastic is used, it takes 500 to 1,000 years to properly maintain it. Plastic can be used in the field to reach air and sunlight that does not contain light, which can exacerbate soil instability and lead to flooding[9]. When the sky is cloudy, the sun's UV rays can cause photodegradation, which can cause plastic to enter food and can even cause people in the vicinity to become ill due to the amplification effect of the food[10]. Then, plastic components made from bottles lying on the ground can harm human health and the environment. For example, if you use plastic that contains hazards, you will be exposed to dioxins. From the last three factors, it is clear that if the use of plastic bottles continues, the environmental ecosystem problems will get worse[11].

The author holds the belief that in order to reduce the frequency of using plastic bottles, some kind of equipment should be developed that can meet the practical, automatic, and economical needs of miners[12]. By having such a device, it will be easier to disassemble the tumbler and encourage people to do it themselves, which will make it safer to use plastic bottles repeatedly.

The way this tool works is that clean water will pass through the flow meter sensor and the flow meter sensor will work to detect water in the form of liters that will come out past the Flow Meter sensor. And the water will be put into a 600ml bottle or according to the bottle we will fill [13]. Flow meter is a water flow detection sensor, where by using a flow meter we will easily find out how much water discharge comes out and makes it easier for us to determine the set points we want when using a flow meter [14].

The development of science and technology encourages humans to create various types of technology that are required by humans to be able to facilitate work. The only technology currently used is in the field of measuring discharge or air flow using a flow meter sensor [15][16].

The problem that currently takes place is that it is difficult for residents to access refillable drinking water in a universal place with realtime output, so the author makes water refill equipment using a flow sensor so that it can recognize the volume of water output in realtime.

2. Methods

A. Software Design

In the ESP32 microcontroller program, this research requires a software tool called Arduino 1.8.19 IDE (Integrated Development Environment) which can be used to assemble, build, and modify Arduino sketches with programs that use C language as the main program language. The following is a display of the Arduino 1.8.19 IDE application to create the coding that will be made:

Figure 1. Arduino IDE 1.8.19 application software

The software creation process using the Arduino 1.8.19 IDE application is described in

Figure 1 above this. It starts with the software downloaded, installed, and set up, a USB cable is added, a Microcontroller Board is used, and an Arduino serial port is selected in the icon tool, and then a sketch is created which will eventually become a program once the sketch is verified and implemented.

B. Hardware Design

The overall appearance of the tool design can be seen in the picture. From the picture, all components are already in an assembled and integrated state. Water flow sensor, LCD, Keyapd, Water pump, Relay are connected to the ESP32 microcontroller.

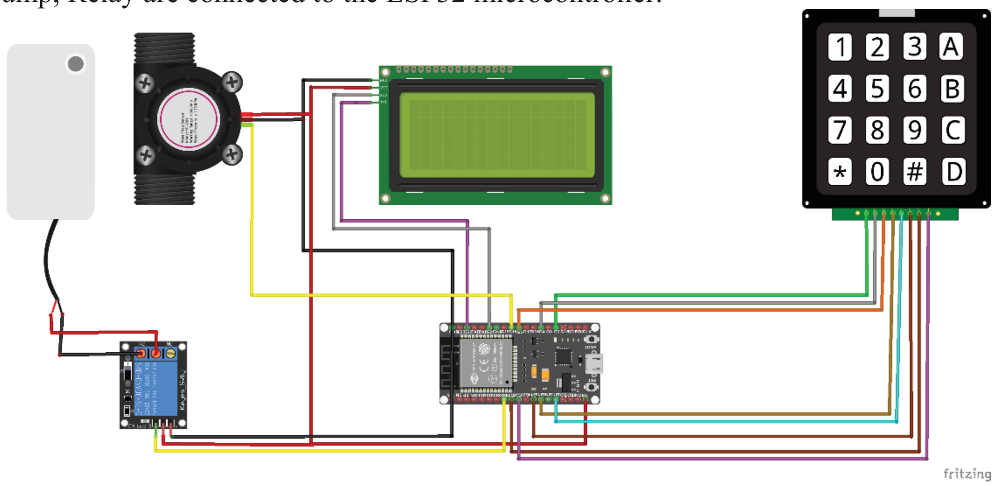

Figure 2. Schematic circuit of the whole water control device

Figure 2 can be explained that the microcontroller used is ESP32. The pins connected to this microcontroller can be seen in table 2 below:

Table 1. Observation of hardware pins on the ESP32 microcontroller

ESP32	DESCRIPTION
GPIO16	col1 (keypad)
GPIO17	col2 (keypad)
GPIO18	col3 (keypad)
GPIO19	col4 (keypad)
GPIO12	row1 (keypad)
GPIO13	row2 (keypad)
GPIO14	row3 (keypad)
GPIO15	row4 (keypad)
GPIO4	Relay
GPIO2	Sensor water flow
VIN 5V	VCC (LCD)
GPIO21	SDA (LCD)
GPIO22	SCL (LCD)
GND	GND (LCD)

C. Flowchart System

The system flowchart in Figure 3 is as follows:

1. Start
   The first thing to do in using this water outlet controller is to connect the device to the electricity.
2. Refill Water
   Users / customers make sure they have brought a drinking bottle to refill.
3. Make Payment
   Users/customers make payments according to the volume of bottles they want to refill.
4. Input Refill Volume (mL)
   After making a payment then the user chooses the volume of water refills according to what is paid..

D. Tool Design

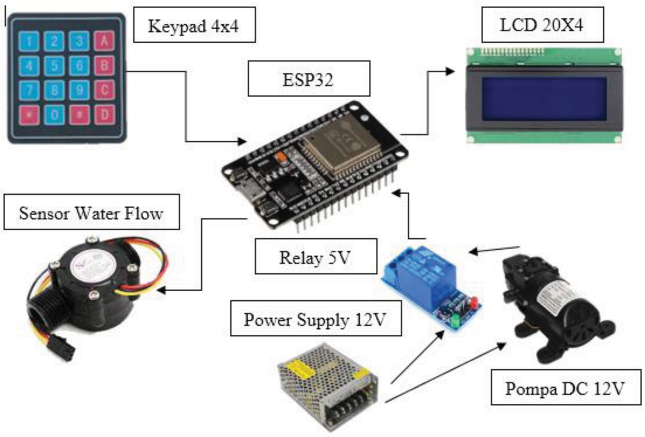

Figure 3. Design of water outlet control device

From fig. 3 it can be explained that the keypad as input for set points, the water flow sensor will send data, then the results will be processed by the ESP32 microcontroller, after processing it will be displayed on the 20x4 I2C LCD in the form of water output data.

E. Block Diagram

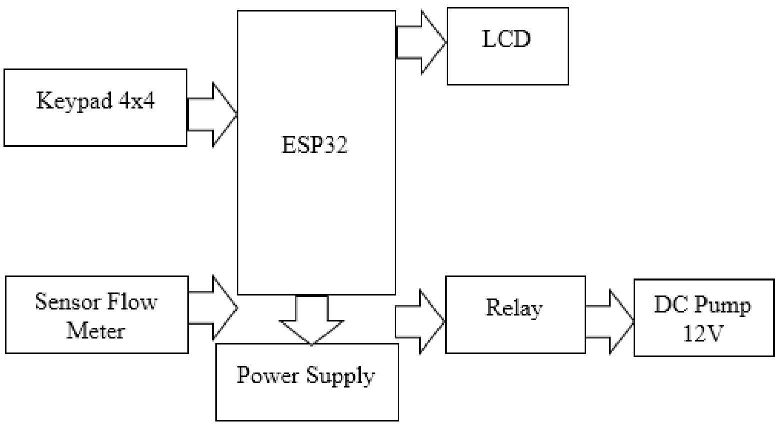

Figure 4. Block diagram of water outlet controller

3. Results and Discussion

The following is a discussion of the tests that will be carried out from the planning that has been assembled. This test is carried out in order to find out how the work and the results of suitability and error in the design with the planning that has been made, therefore after testing it is necessary to make observations and discussions to determine the level of success or deficiencies in the tool, so that conclusions can be drawn from the research conducted.

A. Software Testing

Software testing is done through the Arduino IDE (Integrated Development Environment) to support the C++ programming language. This tool uses an ESP32 microcontroller which will control all other supporting tools such as the water flow sensor as a water exit controller, push button as a set point button, relay as a water pump controller, and 20x4 I2C LCD as a display of the set point menu and water exit. After the program is completed, a compile is done to ensure the program is arranged correctly.

```
void mulaiisi() {
  if ((millis() - oldTime) > 1000)   // Only process counters once per second
  {

    detachInterrupt(sensorInterrupt);
    flowRate = ((1000.0 / (millis() - oldTime)) * pulseCount) / calibrationFactor;
    oldTime = millis();
    flowMilliLitres = (flowRate / 60) * 115;
    totalMilliLitres += flowMilliLitres;

    frac = (flowRate - int(flowRate)) * 10;

    lcd.setCursor(0, 2);
    lcd.print("Spd: ");             // Output separator
    lcd.print(flowMilliLitres);
    lcd.print(" mL/s   ");

    // Print the cumulative total of litres flowed since starting
    lcd.setCursor(0, 3);
    lcd.print("ISI: ");             // Output separator
    lcd.print(totalMilliLitres);
    lcd.print(" mL   ");

    pulseCount = 0;

    attachInterrupt(sensorInterrupt, pulseCounter, FALLING);

    if (totalMilliLitres >= beli) {
      lcd.clear();
      digitalWrite(relay, LOW);
      lcd.setCursor(0, 0);
```

Figure 5. ESP32 program display on Arduino IDE with water flow sensor experiments

B. Hardware Testing

Testing is done to ensure the circuit on a hardware according to the planningTesting is done to ensure the circuit on a hardware according to the planning

Figure 6. Water control hardware circuit

1. Partial System Testing

To ensure the workings and working principles of each tool have been running properly, it is necessary to test the system per part. In taking a data on system testing, testing of each part is carried out until the whole part. The following are the tests carried out:

1. 12V DC Water Pump Testing

   In this test, a water pump is used to send water from a gallon to a water bottle by passing a water flow sensor, where this water pump requires a 12volt DC voltage generated by the water pump power supply so that it must use a power supply so that the required voltage is met.

Figure 7. Testing the water pump with a wattmeter

2. 12V Power Supply Testing

   The following power supply test aims to determine how the voltage coming out of the power supply circuit, this power supply circuit provides a voltage of 12vdc, the voltage generated by the power supply is used to supply the water pump. This experiment was carried out in the umsida electrical engineering laboratory.

Figure 8. Power supply testing

3. 20x4 I2C LCD Testing

   This test aims to display menu options for refillable water and display water output through the water flow sensor. The following is the arch of the test for the display on the I2C LCD:

Figure 9. Testing the reading results of the water flow sensor with the LCD display

C. Analysis of Test Results

The following test is a test of the entire circuit which will be tested for accuracy and the error rate can be seen in the table as follows:

Table 2. Overall water out testing

Testing	Command	Actual	Difference	Error%
1	600	600	0	0%
2	600	600	0	0%
3	600	600	0	0%
4	600	600	0	0%
5	600	**612**	**12**	**1%**
6	600	600	0	0%
7	600	600	0	0%
8	600	600	0	0%
9	600	600	0	0%
10	600	600	0	0%
11	1.000	1.000	0	0%
12	1.000	1.000	0	0%
13	1.000	**980**	**20**	**2%**
14	1.000	1.000	0	0%
15	1.000	1.000	0	0%
16	1.000	1.000	0	0%
17	1.000	1.000	0	0%
18	1.000	**1.008**	**8**	**7%**
19	1.000	1.000	0	0%
20	1.000	1.000	0	0%
21	1.500	1.500	0	0%
22	1.500	1.500	0	0%

23	1.500	1.500	0	0%
24	1.500	1.500	0	0%
25	1.500	**1.518**	**18**	**1%**
26	1.500	1.500	0	0%
27	1.500	1.500	0	0%
28	1.500	1.500	0	0%
29	1.500	1.500	0	0%
30	1.500	1.500	0	0%

It can be concluded from the experimental results described above that the percentage of water filling has a fairly high accuracy rate of 92%. Meanwhile, the failure rate is 8%. Filling the volume of water carried out with three different tests, including 600 mL, 1,000 mL, and 1,500mL among the three has a good level of accuracy while filling large volumes, 1,500 mL has rather poor accuracy. This is because the pressure on the water pump used changes, and disrupts the measurement of water volume, and the pressure on the water flow sensor which does not operate optimally according to what we want. In addition, the process of changing the refill of gallons of water and the process of turning on and off the device that requires a water pump with maximum pressure to increase the maximum level of filling accuracy is also a major contributing factor to refilling water in this test.

4. Conclusion

Based on the comprehensive testing and data collection, the utilization of an ESP32-based water flow sensor for the water outlet controller has been successfully demonstrated. The system effectively enables users to regulate water discharge by utilizing a keypad for selecting desired settings and adjusting the speed within the Arduino IDE code, thereby ensuring the water is dispensed in accordance with predefined set points. To ensure stability and optimal performance of the ESP32 microcontroller in governing the water flow sensor system, it is crucial to provide a voltage supply of at least 5V, considering the voltage requirement specified by the ESP32 microcontroller used in this study.

This conclusion highlights the viability and functionality of the proposed system for water flow control. Future research endeavors should focus on exploring alternative microcontroller options with varying voltage requirements and investigating the potential integration of additional sensors or actuators to enhance the overall control and monitoring capabilities of the system. Furthermore, it would be valuable to conduct field studies to assess the system's performance and durability under real-world conditions, considering factors such as water quality, environmental variability, and long-term reliability.

References

[1] S. D. Br Pelawi and S. Manan, "Sistem Monitoring Volume Air Menggunakan Sensor Ultrasonik Dan Monitoring Output Volume Air Menggunakan Flow Meter Berbasis Arduino," *Gema Teknol.*, vol. 19, no. 2, p. 6, 2017, doi: 10.14710/gt.v19i2.21863.

[2] R. A. Amin Suharjono, Listya Nurina Rahayu, "Aplikasi Sensor Flow Water Untuk Mengukur Penggunaan Air Pelanggan Secara Digital Serta Pengiriman Data Secara Otomatis Pada PDAM Kota Semarang," *J. TELE*, vol. 13, no. 1, pp. 7–12, 2015, [Online]. Available: https://jurnal.polines.ac.id/index.php/tele/article/view/151

[3] A. Bain and B. A. Rozany, "Sistem Kendali Dan Penghitung Biaya Air PDAM Menggunakan Water Flow Sensor," *J. Ilm. Komput.*, vol. 13, pp. 1525–1534, 2017, [Online]. Available: http://ojs.stmik-banjarbaru.ac.id/index.php/progresif/article/view/206

[4] A. Arfandi and Y. Supit, "Prototipe Sistem Otomasi Pada Pengisian Depot Air Minum Isi Ulang Berbasis Arduino Uno," *Simtek J. Sist. Inf. dan Tek. Komput.*, vol. 4, no. 1, pp. 91–

99, 2019, doi: 10.51876/simtek.v4i1.53.
[5] A. R. Ardiliansyah, M. D. Puspitasari, and ..., "Rancang Bangun Prototipe Pompa Otomatis Dengan Fitur Monitoring Berbasis IoT Menggunakan Sensor Flow Meter dan Ultrasonik," *Explor. IT! J. ...*, vol. 5, no. 36, pp. 59–67, 2021, [Online]. Available: https://www.jurnal.yudharta.ac.id/v2/index.php/EXPLORE-IT/article/view/2601%0Ahttps://www.jurnal.yudharta.ac.id/v2/index.php/EXPLORE-IT/article/download/2601/1979
[6] Kallam. and Siregar., "Viskosimeter Digital Menggunakan Water Flow Sensor G1/2 Berbasis Mikrokontroller," *Saintika Fis.*, vol. 4, no. 1, pp. 1–5, 2013.
[7] R. Hartono, "Optimasi Penggunaan Sensor Water Flow HF-S201 Guna Mengukur Aliran Air Mendukung Mitigasi Banjir," *Indones. J. Appl. Informatics*, vol. 5, no. 2, p. 161, 2022, doi: 10.20961/ijai.v5i2.44603.
[8] A. Bagaskoro and S. T. A. Budiman, "Prototipe Pengisian Air dalam Botol Kemasan Berbasis Arduino," 2021, [Online]. Available: http://eprints.ums.ac.id/id/eprint/93560%0Ahttp://eprints.ums.ac.id/93560/2/Naskah Publikasi.pdf
[9] A. Solih and J. Jamaaluddin, "Rancang Bangun Pengaman Panel Distribusi Tenaga Listrik Di Lippo Plaza Sidoarjo Dari Kebakaran Berbasis Arduino Nano," *JEEE-U (Journal Electr. Electron. Eng.*, vol. 1, no. 2, pp. 61–68, 2017, doi: 10.21070/jeee-u.v1i2.1171.
[10] A. Supriyadi and J. Jamaaluddin, "Analisa Efisiensi Penjejak Sinar Matahari Dengan Menggunakan Kontrol ATMEGA16," *JEEE-U (Journal Electr. Electron. Eng.*, vol. 2, no. 1, pp. 8–15, 2018, doi: 10.21070/jeee-u.v2i1.1172.
[11] I. Anshory, D. Hadidjaja, and I. Sulistiyowati, "Monitoring Perubahan Tegangan dan Pemodelan Matematika Fungsi Transfer Motor BLDC Dengan System Identification Toolbox Program Studi Teknik Elektro , Fakultas Saintek , Universitas Muhammadiyah Sidoarjo," pp. 18–25.
[12] P. B. Megantoro, I. Anshory, I. Sulistiyowati, and ..., "Design and Simulation of PID Speed Controller on BLDC Motor," *Pros. Semin. ...*, pp. 41–49, 2022, doi: 10.31284/j.JREEC.2022.v2i1.2623.
[13] J. W. SIMATUPANG, B. PRASETYO, M. GALINA, and A. SUHARTOMO, "Prototipe Mesin Penjual Air Mineral Otomatis berbasis Arduino Mega 2560 dan RFID-RC522," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 10, no. 2, p. 484, 2022, doi: 10.26760/elkomika.v10i2.484.
[14] A. Ghurri, S. P. G. G. Tisna, and S. Syamsudin, "Pengujian Orifice Flow Meter dengan Kapasitas Aliran Rendah," *Mechanical*, vol. 7, no. 2, pp. 61–66, 2016, doi: 10.23960/mech.v7.i2.201610.
[15] M. I. Tohari, J. Jamaaluddin, I. Sulistiyowati, T. Elektro, and U. M. Sidoarjo, "Sistem pengenalan suara sebagai pengendali peralatan audio berbasis arduino uno 1," pp. 86–90, 2021.
[16] P. K. Olla and W. Azhar, "Rancang Bangun Peak Flow Meter dengan Output Suara Berbasis Android," *Avitec*, vol. 3, no. 1, pp. 43–56, 2021, doi: 10.28989/avitec.v3i1.884.