WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

Smart IV Fluid Flow Monitoring, Auto-Control and Patient Safety System

Prof. Rajani V. A [1], Annapurna M Ganachari [2], Mahammadshoyeb Huddar [3], Mohammadayad Sutar [4], Srujala Mallur[5]

Prof. Rajani V, A, Assistant Professor, Department of Electronics and communication Engineering/Jain College of Engineering and Research, Belagavi
Annapurna M Ganachari, Department of Electronics and communication Engineering/Jain College of Engineering and Research, Belagavi
Mahammadshoyeb Huddar, Department of Electronics and communication Engineering/Jain College of Engineering and Research, Belagavi
Mohammadayad Sutar, Department of Electronics and communication and Engineering/Jain College of Engineering and Research, Belagavi
Srujala Mallur, Department of Electronics and communication and Engineering/Jain College of Engineering and Research, Belagavi

ABSTRACT—

In hospitals, IV fluid therapy is one of the most routine ways of giving patients fluids and medicines, yet someone still has to keep an eye on the drip rate and on how much fluid is left in the bag. Doing this by hand means nurses must keep coming back to check each patient. To ease that burden, this paper presents a Smart IV Fluid Flow Monitoring, Auto-Control and Patient Safety System built around an ESP32 microcontroller. An infrared (IR) sensor counts the drops as they fall, and a load cell connected to an HX711 module keeps track of the weight of the fluid container. Once the drop count reaches a target set by the user, a servo motor closes the roller clamp, and a buzzer and LED raise an alert. A web-based interface lets the user set the target drop count and view the drop count, fluid weight, and valve status in real time. The prototype combines monitoring, automatic control, and alerts in a single system, which helps cut down on repeated manual checks.

Index Terms— IV Fluid Monitoring, ESP32, IR Sensor, Load Cell, HX711, Automatic Flow Control, Patient Safety.

I.INTRODUCTION

Intravenous (IV) therapy is a common part of hospital care. It delivers fluids, electrolytes, and certain medications straight into a patient's bloodstream. While it runs, the drip rate and the amount of fluid left in the container need to be watched closely. In a typical setup, the flow is adjusted by hand with a roller clamp, and nurses go back every so often to look at the falling drops and the fluid level. This works well enough, but when one nurse is looking after several patients at once, regular checks are hard to keep up. A system that can observe the infusion continuously would clearly help.

Advances in embedded systems and sensors have made tasks like this much easier to automate. A microcontroller can collect sensor readings, process them, and raise an alert as soon as a predefined condition is met. By adding automatic control and a web interface to the mix, IV fluid monitoring can become more convenient and far less dependent on repeated manual checking.

In this paper, we present a Smart IV Fluid Flow Monitoring, Auto-Control and Patient Safety System based on the ESP32 microcontroller. An IR sensor detects and counts the falling drops, while a load cell paired with an HX711 module weighs the fluid container to estimate how much remains. When the drop count reaches the target set by the user, a servo motor closes the roller clamp, and a buzzer and LEDs give an audible and visual alert. Through the web interface, users can set the target drop count and view the drop count, fluid weight, and valve status.

The main goal of this project is to bring IV fluid monitoring, automatic flow control, and alert generation together in one embedded system. By giving real-time information about how the system is running, the prototype aims to make monitoring easier and reduce repetitive checks. That said, it is a prototype only. It needs further testing and clinical validation before it can be used in patient care, and it is not a substitute for qualified medical supervision or clinically validated infusion equipment.

II. LITERATURE SURVEY

Over the years, researchers have tried to make IV fluid monitoring easier by bringing in sensors, microcontrollers, and wireless technologies. Most of these systems share the same basic aims: tracking the drip rate, measuring how much fluid is left, and letting healthcare staff know when someone needs to step in. Every approach has its own strengths and weak spots.

The authors of *DripOMeter: An Open-source Opto-Electronic System for Intravenous (IV) Infusion Monitoring* built a system that uses optical sensing to spot and track falling IV drops [1]. It offers a simple, low-cost way to watch drip activity in real time. Its focus, though, is on monitoring alone. It has no motorized roller clamp, so it cannot adjust the flow on its own.

Another work, *Intravenous Drip Monitoring System*, paired an LED-LDR sensor setup with a load cell to follow the falling drops and measure the fluid level [2]. Using both sensors gives a fuller picture, showing the drip rate as well as the amount of fluid left in the container. Whether the flow can be regulated automatically, however, depends on the control features built into the design. Arduino-based drip monitoring systems have also been reported [3], showing that low-cost microcontrollers are enough for this kind of task.

Cloud-based IV monitoring systems have also been explored so that infusion data can be viewed from a distance [4]. They let users access readings from anywhere and may offer alerts and data logging. The catch is that they depend on a stable internet connection and extra communication infrastructure. Other studies have looked at optical, ultrasonic, and capacitive sensors for detecting fluid movement [5]. These give us several ways to monitor IV fluid, but the choice of sensor can noticeably change the cost and complexity of the whole system.

Looking across these studies, it is clear that sensors and connected technologies can make IV monitoring much more convenient. Still, there is real value in putting drop counting, fluid weight measurement, automatic flow control, audible alerts, and web-based monitoring together in one system. That is the gap our work addresses. The proposed Smart IV Fluid Flow Monitoring, Auto-Control and Patient Safety System uses an ESP32 microcontroller together with an IR sensor, a load cell, an HX711 module, a servo motor, and a web-based interface. By combining these functions in a single prototype, the system aims to make monitoring simpler and IV fluid delivery better controlled.

III. PROBLEM STATEMENT

In most hospitals today, IV drip rates are still set by hand. A nurse has to keep coming back to look at the falling drops and check how much fluid is left in the bag. That is manageable for one patient, but it gets hard when several patients need attention at the same time. A change in the drip, or a bag that is running low, can easily go unnoticed for a while. What is needed is a simple, affordable system that keeps an eye on the drops, measures the fluid that remains, raises an alert when the preset target is reached, and closes the roller clamp on its own. The proposed system does this by bringing together sensors, an ESP32 microcontroller, a servo-operated roller clamp, and a web-based interface.

IV. OBJECTIVES

The main aim of this project is to build a smart IV fluid monitoring and control system that takes some of the effort out of watching an infusion. More specifically, the project sets out to:

1. Detect and count each falling IV drop with an infrared (IR) sensor, so the infusion can be monitored.
2. Weigh the IV fluid container with a load cell and an HX711 module, so the amount of fluid left can be estimated.
3. Use an ESP32 microcontroller to process the sensor readings and manage monitoring, alerts, and control.
4. Close the IV roller clamp with a servo motor once the target drop count set by the user is reached.
5. Give an audible alert through a buzzer, and show the system state through LEDs, when the target is reached.
6. Build a web-based interface where users can set the target drop count and see the drop count, fluid weight, and valve status.
7. Combine all of these monitoring and control functions in one prototype, and test it to see how well it works.

V. PROPOSED SYSTEM

A. System Architecture

The proposed Smart IV Fluid Flow Monitoring, Auto-Control and Patient Safety System has four main parts: sensing, processing, automatic control, and Wi-Fi communication. Each part has a clear job. Together they watch the IV drops, work out how much fluid is left, raise an alert when the target is reached, and show the current status on a web page.

Sensing comes first. An infrared (IR) sensor sits at the drip chamber and registers every drop that falls through it. At the same time, a load cell connected to an HX711 module weighs the fluid container. Both readings go to the ESP32, which acts as the brain of the system. It keeps the drop count, estimates the remaining fluid from the weight, and compares the drop count with the target set by the user to decide what to do next.

When the drop count reaches the target, the ESP32 tells the servo motor to close the roller clamp, sounds the buzzer, and turns on the red LED, so the user knows right away. Throughout this process, the ESP32 keeps the OLED display up to date and sends updates over Wi-Fi to the web interface. There, the user can set the target drop count and check the drop count, fluid weight, and valve status without standing next to the IV stand. Fig. 1 shows the block diagram of the proposed system.

In terms of wiring, the IR sensor and the load cell connect to the ESP32 through separate input lines. The servo motor, buzzer, LEDs, and OLED display are the output devices, and the web interface communicates with the ESP32 over Wi-Fi.

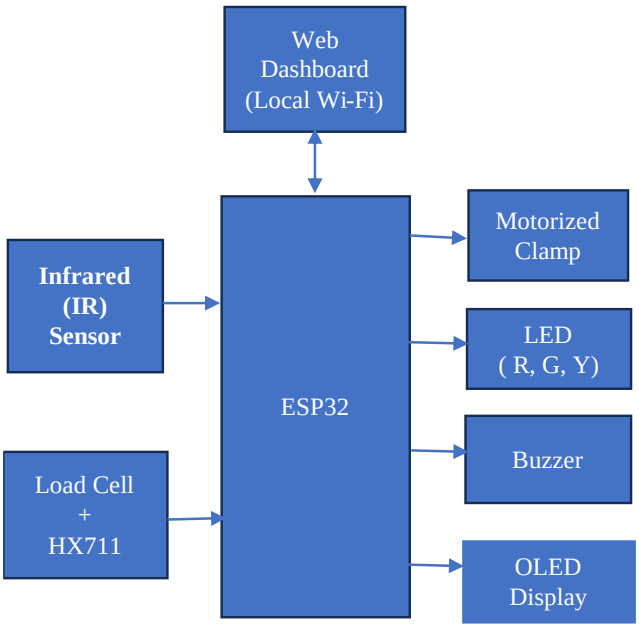


Fig. 1. Block Diagram of the Proposed System

B. Key Components

The system is built from a handful of components, and each one has a specific job in monitoring the IV drops, measuring the remaining fluid, closing the clamp, and raising alerts.

1) ESP32 Microcontroller: The ESP32 is the main controller of the whole setup. It collects data from the sensors, processes the readings, drives the servo motor, buzzer, LEDs, and OLED display, and talks to the web interface over Wi-Fi.

2) Infrared (IR) Sensor: The IR sensor is placed at the drip chamber and picks up each drop as it falls. The ESP32 counts these drops and compares the count with the target.

3) Load Cell and HX711 Module: The load cell measures the weight of the IV fluid container. Its signal is very small, so the HX711 module amplifies it and converts it into digital data the ESP32 can read. Once the system is calibrated, this weight can be used to estimate how much fluid is left.

4) Servo Motor: The servo motor turns the roller clamp to close the IV line when the target drop count is reached. Its mechanical mounting needs to be done carefully so the clamp can move smoothly and reliably between the open and closed positions.

5) Buzzer: The buzzer gives an audible alert when the target drop count is reached, so the user knows when attention is needed.

6) LEDs: Three LEDs show the system state at a glance. Green means the system is running, yellow means it is idle and ready, and red means the target has been reached and the clamp is closed.

7) OLED Display: A 128 × 64pixel OLED display shows the drop count, weight, and system status directly on the device, so the user can check it without opening the web page.

8) Web-Based Interface: Through the web interface, the user can set the target drop count and keep track of key readings such as the drop count, fluid weight, and valve status. It communicates with the ESP32 over Wi-Fi.

7) Power Supply: The power supply gives the ESP32 and the other components the voltage they need to run. Proper voltage regulation and enough current capacity matter here, especially when the servo motor moves, since it draws a noticeably higher current at that moment and can cause the system to reset or misbehave if the supply is weak.

VI. Implementation

A. Hardware Implementation

The hardware prototype is built around an ESP32 microcontroller. Alongside it are an IR sensor, a load cell with an HX711 module, a servo motor, a buzzer, three LEDs, and an OLED display. The IR sensor watches the falling drops, and the load cell measures the weight of the IV fluid container. The

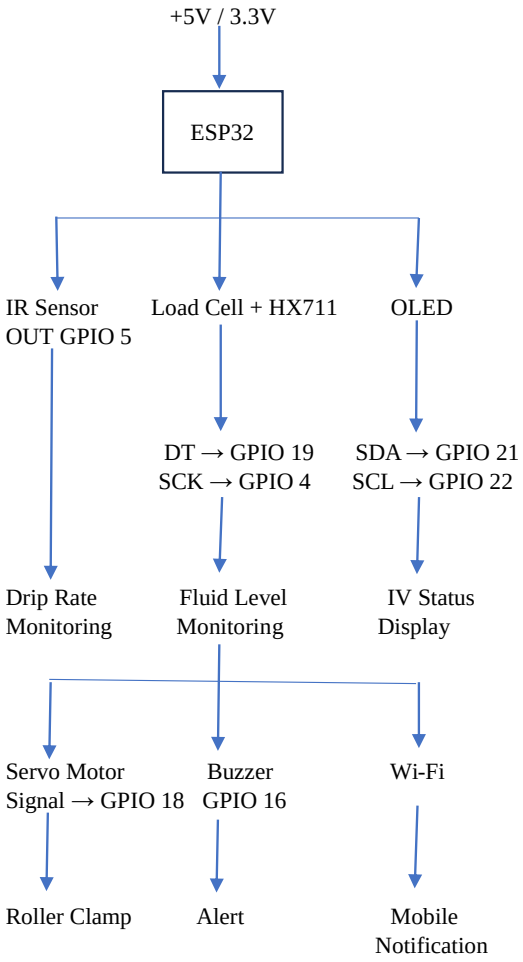
ESP32 reads both, processes the data, and drives the output devices according to the programmed conditions. Fig. 2 shows the circuit connection diagram, and Fig. 3 shows the assembled prototype.

Each component has its own pin on the ESP32. The IR sensor output goes to GPIO 5, and the HX711 data pin (DT) and clock pin (SCK) go to GPIO 19 and GPIO 4. The servo motor is connected to GPIO 18 and closes the roller clamp when the target is reached, following the implemented control logic. The buzzer sits on GPIO 16 and sounds an alert at that moment. For quick visual feedback, three LEDs are connected to GPIO 25, GPIO 26, and GPIO 33 to show the system status.

A 128 × 64pixel OLED display is connected through the I²C interface, with GPIO 21 as SDA and GPIO 22 as SCL. It shows the system's operating information right at the device, so the user does not need to open the web page for a quick check. The ESP32 also uses its built-in Wi-Fi to talk to the web-based interface, where users can set the target drop count and view the drop count, fluid weight, and valve status. All of these parts are put together in a single prototype that handles monitoring, alert generation, and automatic flow control.



Fig. 2. Circuit connection diagram



B. Software Implementation

When the system is powered on, the ESP32 first sets up everything it needs to work with: the IR sensor, the load cell and HX711 module, the servo motor, the buzzer, the LEDs, the OLED display, and the Wi-Fi connection. Once that is done, it starts reading the sensors without a break, counting each IV drop that falls and measuring the weight of the fluid container. The drop count is then compared with the target set by the user.

The program follows a simple chain of steps: Sensor Reading → Data Processing → Flow Monitoring → Threshold Comparison → Control Decision. When the count reaches the target, the ESP32 closes the roller clamp with the servo motor, switches on the buzzer and the red LED, and stops counting. At every step, it refreshes both the OLED display and the web interface. The cycle then waits for the user to set a new target and restart, as shown in Fig. 3.

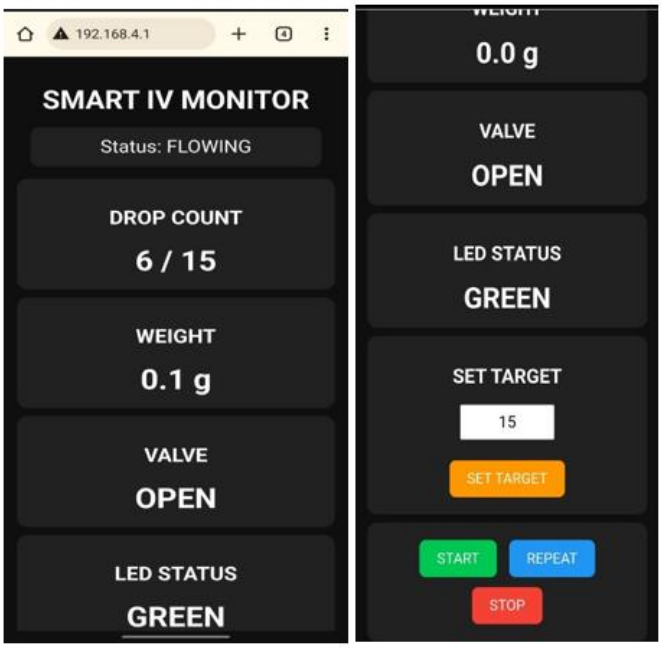
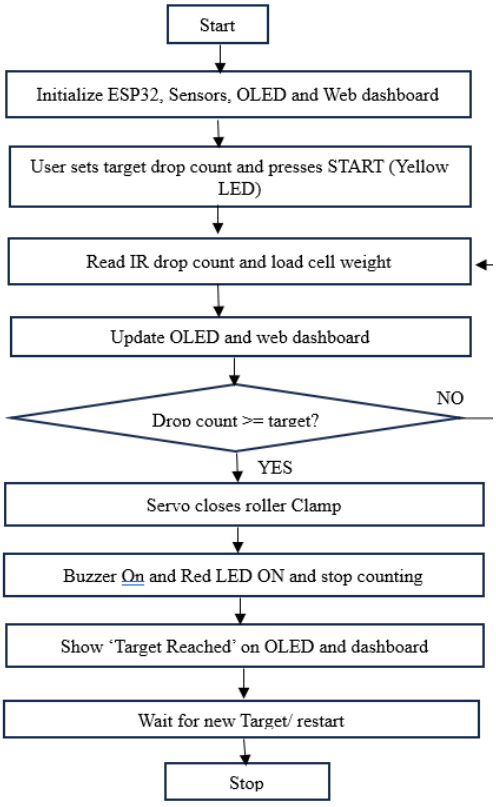


Fig. 3. Flow diagram



C. Sensor Calibration and Threshold Selection

Load Cell Calibration: The load cell is connected to the ESP32 through the HX711 amplifier module, with the DT pin on GPIO 19 and the SCK pin on GPIO 4. To get accurate readings, the load cell is calibrated with a known reference weight, and the calibration factor is adjusted until the measured value matches it. The weight is calculated using the following equation:

$$W = (R - R_o) / C$$

where W is the measured weight, R is the current sensor reading, R_o is the zero-offset reading, and C is the calibration factor. The program starts with a calibration factor of −7050.0, which is then fine-tuned based on the calibration results.

Drop Detection and Flow-Rate Calculation: The IR sensor on GPIO 5 detects each drop as it falls through the IV set. To find the flow rate, the drops are counted over a fixed time interval and the result is converted to drops per minute (DPM) as:

$$Q = (N \times 60) / T$$

where Q is the flow rate in DPM, N is the number of drops detected, and T is the measurement interval in seconds. The flow rate is used for monitoring only. The control decision depends on the target drop count.

Threshold Selection: The target drop count is set by the user through the web-based dashboard. Once the detected drop count reaches this target, the ESP32 uses the servo motor to close the roller clamp, sounds the buzzer, and turns on the red LED. The other two LEDs show the system state: green means it is running, and yellow means it is idle and ready. From the dashboard, the user can also set a new target and restart the monitoring cycle.

VII. Results and Discussion

The prototype brings together an ESP32 microcontroller, an IR drop sensor, a load cell with an HX711 amplifier, a servo motor, a buzzer, three LEDs, an OLED display, and a web-based dashboard. Its job is to watch the IV drops and the container weight, and to step in automatically once the configured target is reached.

During operation, the IR sensor picks up each drop and the ESP32 keeps the drop count updated. Meanwhile, the load cell measures the weight of the IV fluid container. Both the readings and the current system status can be seen on the OLED display and on the web dashboard, which talks to the ESP32 over a local Wi-Fi connection.

When the drop count reaches the preset target, the ESP32 stops the counting cycle and moves the servo motor to close the roller clamp. At the same time, it sounds the buzzer and turns on the red LED. The other two LEDs show the system state: green means it is running, and yellow means it is idle and ready. Together, these actions show that drop monitoring, weight

measurement, alerts, and automatic flow control can work side by side in a single prototype.

Experimental Readings:

Table I shows the system response at each stage of a test run with a target of 15 drops. The weight values are calculated from the drop volume, assuming a 20 drops/mL IV set (about 0.05 g per drop) and an initial container weight of 250 g. The drop count, servo state, buzzer state, and LED indication follow the programmed control logic. When the count reaches the target, the servo closes the roller clamp, the buzzer turns on, and the red LED lights up. Measured load cell readings will be compared against these expected values in future testing.

TABLE I: System Response for a Test Run

(Target = 15 Drops)

SI. NO	Target Drop Count	Detected Drop Count	Load Cell Reading (g)	Servo Status	Buzzer Status	LED Status
1	15	0	250	Open	OFF	Yellow
2	15	5	249.75	Open	OFF	Green
3	15	10	249.50	Open	OFF	Green
4	15	14	249.30	Open	OFF	Green
5	15	15	249.25	Closed	ON	Red

VIII. Advantages and Limitations

A. Advantages

The system lets users follow IV fluid drops and container weight in real time, using an IR sensor and a load cell. When the preset drop-count target is reached, the ESP32 closes the servo-operated roller clamp, and the buzzer and LEDs raise an alert. The web dashboard runs over a local Wi-Fi connection, so users can set the target and check the system status without needing internet access. Because the prototype is built from low-cost components, it is a good fit for research and educational work.

B. Limitations

How well the system performs depends on the accuracy of the sensors, a proper load cell calibration, and a servo that works reliably. The IR sensor can miss drops if it is not aligned correctly, and calibration errors can make the weight readings drift. The system also needs a stable power supply and has to stay within range of the local Wi-Fi network. More testing is needed to measure its accuracy and reliability. Before anyone considers using it with patients, it must go through clinical validation and safety testing.

IX. Conclusion and Future Scope

A. Conclusion

This paper presented a Wi-Fi-enabled Smart IV Fluid Monitoring and Automatic Control System. It combines an ESP32, an IR sensor, a load cell, a servo motor, and alert indicators to keep track of IV fluid drops and container weight. A web-based dashboard lets the user set the target and view the system status. Once the preset drop-count target is reached, the servo-operated clamp closes, and the buzzer and red LED turn on. The prototype shows that monitoring, alerts, and automatic control can be brought together using low-cost components.

B. Future Scope

There is room to improve this work in several ways. Sensor accuracy can be improved, the flow-control mechanism can be refined, and remote monitoring with mobile notifications can be added. Data logging and cloud connectivity would allow long-term monitoring and analysis. Further testing under different operating conditions would help confirm how reliable the system is. Finally, extra safety features and proper clinical validation would be essential before the system could be considered for real medical use.

References

[1] K. Venkatesh et al., “DripOMeter: An Open-Source Opto-Electronic System for Intravenous (IV) Infusion Monitoring,” *HardwareX*, vol. 12, 2022, doi: 10.1016/j.ohx.2022.e00345.

[2] R. S. Gaikwad and P. S. Desai, “Intravenous Drip Monitoring System,” *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 7, 2019.

[3] D. Patel and H. Shah, “Patient Intravenous Drip Monitoring System Using Arduino,” *International Journal of Scientific Research in Engineering and Management (IJSREM)*, 2020.

[4] S. K. Sharma and R. Gupta, “Cloud-Based Intravenous Drip Monitoring System,” *International Journal of Innovative Science and Research Technology (IJISRT)*, 2021.

[5] A. R. Patil and M. S. Kulkarni, “Sensor-Based Intravenous Fluid Monitoring System: A Review,” *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 2018.