Received: 23 March 2024 | Revised: 23 May 2024 | Accepted: 12 June 2024 | Published online: 24 June 2024

RESEARCH ARTICLE

Archives of Advanced Engineering Science
2025, Vol. 3(4) 241–251
DOI: 10.47852/bonviewAAES42022901

BON VIEW PUBLISHING

Controlled Fluid Flow Without Controlling Pump Through Arduino

Shamsher Ali[1,*], Syed Muhammad Daniyal Gillani[1], Syed Muhammad Zain[1] and Sudais Ahmad Khan[1]

[1]*National University of Science and Technology, Pakistan*

Abstract: Various industries like chemical, energy, nuclear, process, manufacturing, healthcare, and agriculture require the flow rate to be controlled. It may be constant, steady state, or between the specific upper and lower range. This article explained the development of a cost-effective fluid flow control system using an Arduino microcontroller. The direct use of the pump is not good because there are a lot of challenges (corrosion, power fluctuations, pressure leak) with the pump, and with time as it gets old, the chances of getting good, desired flow from it are not possible. We achieved the controlled flow rate (volume rate and mass rate) in a very precise and cost-effective manner because the flow corresponding to the height is steadier and more constant than getting steady flow from the pump. Hence, our solution is that we don't get direct flow from the pump but from tank B. Tank B must be very carefully designed (especially its height and cross section) to get the desired value of the flow rate from it. As per Bernoulli's theorem, the flow velocity has a direct relation with the height. We just used the pump to refill tank B up to the desired level. In this case, the flow rate only varies when there is a larger change in the head of the tank from which it is coming (it is tank B in our case). The pump automatically becomes ON when the flow rate drops below 2 and fills tank B again to the height which corresponds to the flow rate 3, and when the flow rate becomes 3, the pump becomes OFF automatically. The process is repeated until the desired process is completed. After completion, the valve from tank B can be closed. The hardware integrations, software programming logic, and physical experimental setup are discussed.

Keywords: fluid flow rate, steady-state flow, hall effect sensor, automation, controlled flow, Arduino UNO

1. Introduction

In the process and petrochemical industry, maintaining a steady controlled flow is a must for ensuring stoichiometry, which is mandatory for the accurate mixing of chemicals [1]. In fluid dynamics research, precise control of fluid flow enables scientists to develop steady-state conditions and study phenomena like turbulence, pressure, friction, viscosity, and other important properties. In the oil and gas sector, controlled flow is essential for efficient extraction, transportation, and refining processes, optimizing production, crude oil feeding to fractional distillation, and minimizing waste [2]. Similarly, in the nuclear sector, maintaining precise fluid flow is vital for reactor cooling systems and safety measures, ensuring stable and reliable operation [3]. Thus, the importance of steady controlled flow extends across multiple industries, driving innovation and progress.

The pump is usually used directly for the controlled flow desire. However, there are issues with using the pump; it is not easy to get the pump of your choice, capacity, and power because the manufacturer produces them according to some specific standard. Let's assume according to your calculations you need a pump to maintain the continuous flow rate of 2.5 m^3/s, but if that specification is not available in the market, then you have to compromise on the process control which is not good. Its maintenance costs are bulky also. Moreover, as a machine, it has many operating problems and risks. With electricity fluctuations, it may become short, and due to pressure leakage, the controlled flow is disturbed. Corrosion problems in the impeller reduce the efficiency. Due to all these circumstances, we can say that using the pump for very sensitive processes (where very high-cost chemicals are used as fluid) is highly risky.

This article dives into the development of sustainable and affordable systems for managing fluid flow, powered by an Arduino microcontroller. It focuses on maintaining the controlled flow of the fluid by maintaining/controlling the height level in tank B (shown in Figures 1 and 2). Controlled flow means that the speed, mass flow rate, and volume flow rate are constant or steady state or in the specific upper and lower limit (i.e., in this work, the desired range is 2 m^3/s to 3 m^3/s) [4].

Volume flow rate is the volume of the fluid passed through a point in a second, measured in cubic meters per second (m^3/s). Mass flow rate is the mass of the fluid passed through a point in a second, measured in kilograms per second (kg/s). The speed of the fluid means the distance covered by the fluid in a second, measured in meters per second (m/s).

Controlled, continuous, and steady-state flow is achieved whose speed and flow rate are in the desired range in our case the flow rate is precisely between 2 and 3 (m^3/s) and only varies when there is a larger variation in the head of the tank from which it is coming (it is tank B in our case). The pump automatically becomes ON when the flow rate drops below 2 and fills the tank B again to the height which corresponds to the flow rate 3, and when the flow rate becomes 3, the pump becomes OFF

*Corresponding author: Shamsher Ali, National University of Science and Technology, Pakistan. Email: shali.me43ceme@student.nust.edu.pkt

Figure 1
Arduino UNO development board with Atmega328p microcontroller chip

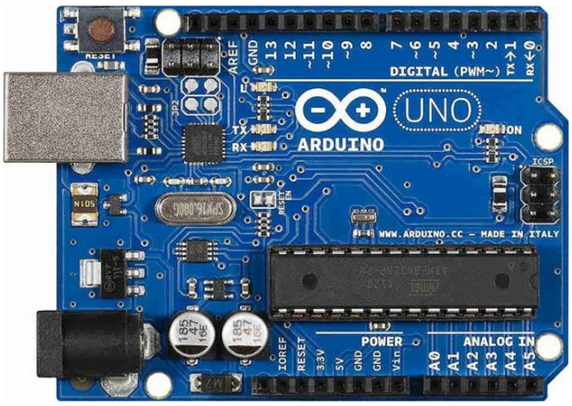


Figure 2
YF-S201 Hall effect sensor for measuring fluid flow rate

automatically. The process is cyclic and repeated again and again until the desired process is completed (process means any chemical, mechanical, or research process where controlled flow is needed). When the process is complete, the valve of the flow from tank B can be closed. The hardware integrations, software programming logic, and physical experimental setup are discussed in the next sections.

In tank C (shown in Figures 1 and 2), we also fitted a water temperature sensor to closely monitor temperature because tank C is our main target and the place where the most sensitive process/reaction is going on. In engineering, temperature measurement is one of the most important factors. The physical setup and hardware and software components are discussed in detail. The system's versatility across various industries is highlighted, showcasing its potential in sectors like manufacturing, healthcare, and agriculture.

Industry 4.0 refers to the integration of modern technologies such as the Internet of Things (IoT), autonomous control, artificial intelligence (AI), and robotics into various industries including manufacturing, healthcare, and agriculture, replacing old and traditional industrial manufacturing techniques and practices. Additionally, exploring alternative methods like flow control valves and investing in higher-sensitivity equipment, improved actuators, and advanced microprocessors could greatly enhance performance [5]. This article suggests the potential integration of proportional-integral-differential (PID) control to boost accuracy by continuously correcting errors, paving the way for a more responsive and adaptable system. As industries increasingly adopt automation and precision in their processes, the outlook for fluid flow control systems appears promising, offering opportunities for innovation and optimization.

Some important aspects of fluid control in industries include the following [6]:

1) **Precision:** The requirement for precise control over fluid flow rates is crucial in many industrial sectors such as pharmaceuticals, chemicals, and beverages where small variations in flow can impact product quality.
2) **Process optimization:** Fluid flow control is essential for optimizing various industrial processes. For instance, in petrochemical and pharmaceutical industries, maintaining precise control over the flow of fluid is necessary for maintaining the correct composition of chemicals and liquids in a tank.
3) **Temperature regulation:** Alongside fluid flow control, temperature regulation is also crucial in applications such as heat exchangers, chemical reactions, and materials processing.
4) **Safety:** Automating industrial processes such as mixing harmful substances ensures safety and reliability, preventing damage and injuries from harmful substances.

Instead of controlling the pump directly, this research focuses on the setup of a tank (which is tank B in this case) in which the water head is "h" and the speed/flow rate is directly associated with the "h" by the Bernoulli's theorem, that is, $V=(2gh)^{1/2}$. We will see its derivation in the proceeding sections. As said earlier, there are many problems with the pump: first, its maintenance costs, and second, electricity fluctuations: it may also become short, and due to pressure leakage, the controlled flow is disturbed. Corrosion problems in the impeller reduce the efficiency. Thus, to prevent the risks and increase safety and precision the flow from the tank is good. Moreover, the consistency, steadiness, and controlled flow from the tank are better than that from the pump [7].

1.1. Project overview

In addition to the broader context of fluid flow in Industry 4.0, this research paper discusses the development and implementation of a fluid flow process control system using an Arduino microcontroller providing an efficient and cost-effective method of fluid flow control system. The key components used for this implementation include the following:

1) **Arduino UNO R3:** In Figure 1, Arduino UNO is a popular microcontroller board providing an easy way to interface with the ATmega328 microcontroller which acts as the central processing unit or the brain of the entire process control project. It was responsible for creating a processing data captured from input transducers, processing it, and executing control flow instructions and algorithms to the output actuators.

Figure 3
DC centrifugal water pump

Figure 4
Arduino LCD and I2C module

Figure 5
DS18B20 temperature sensor

2) **Hall effect flow sensor (YF-S201):** Flow sensor in Figure 2. An input transducer responsible for measuring the fluid flow rate based on the Hall effect, providing accurate and real-time data.
3) **12V DC centrifugal water pump:** Water pump in Figure 3. Responsible for pumping the fluid from a reservoir to the tank, that is, from lower potential to higher potential. The pump consists of rotors called impellers that produce centrifugal force when rotating; this force allows it to pump water toward the outlet.
4) **Liquid crystal display (LCD) with I2C:** An Arduino LCD module shown in Figure 4, interfaced with the Arduino using the inter-integrated circuit I2C module. The LCD is responsible for outputting real-time data and information to the user.
5) **Water temperature sensor:** The DS18B20 water temperature sensor (Figure 5) was used to monitor the real-time temperature of the fluid and send this raw data to the Arduino microcontroller.
6) **Dual channel relay:** (Figure 6) Used to control the on/off state of the 12V DC pump based on signals received from Arduino. The relay is used to provide power to the DC pump through an external 12V source without damaging the Arduino UNO. It consists of an electromagnet which when powered closes the switch forming a complete circuit and allowing current flow in the circuit.

All these hardware components are cheap and readily available in the market and are perfect for a small-scale fluid process control project.

2. Methodology

In Figure 7, the complete process is represented through a block diagram. In Figures 8 and 9, the setup involves the use of three tanks: tank A, tank B, and tank C, with tank A acting as a water source used to pump water to tank B via a DC pump. The impellor inside the DC pump pushes water toward the outlet due to centrifugal force. This output water flows from tank B to tank C, which is placed directly

Figure 6
Dual channel relay module

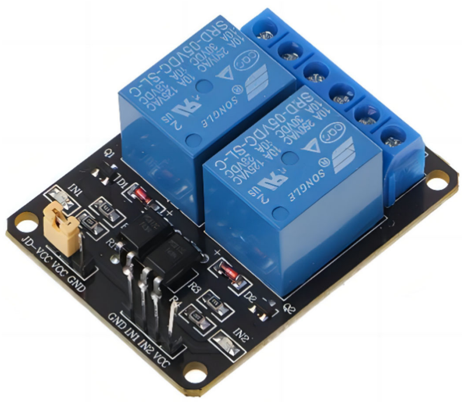


Figure 7
Schematic diagram of tanks and flow

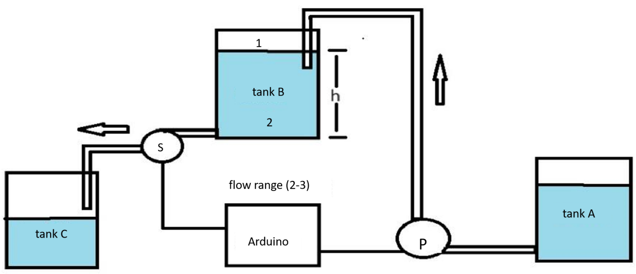
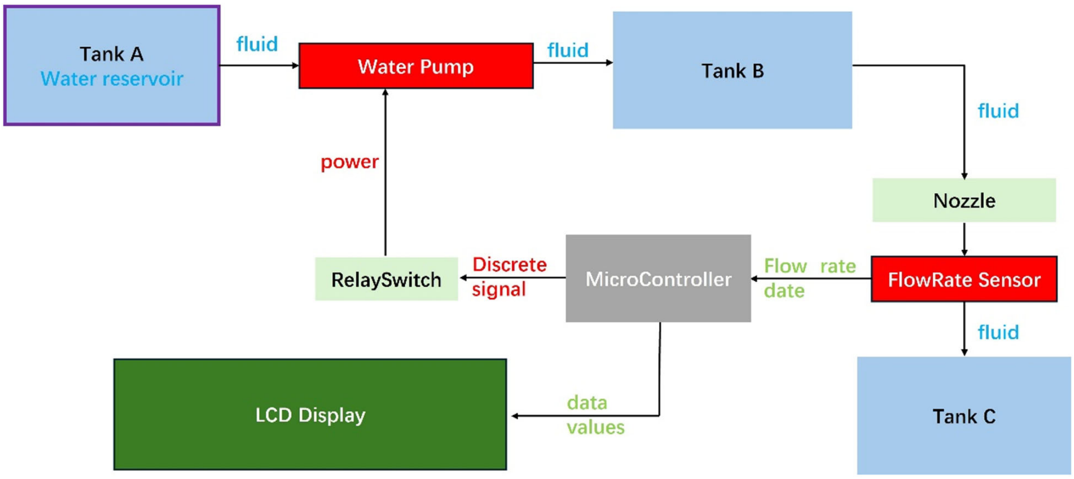


Figure 8
Process flow diagram



underneath it, through a nozzle due to potential energy. The Hall effect flow rate sensor is placed next to the nozzle, the flow rate sensor used is YF201 which works on the principle of the Hall magnetic effect. The assembly consists of a rotor and a Hall effect sensor placed inside a plastic body. As the rotor rotates due to water flow, the sensor generates an electric pulse which is fed into the Arduino microcontroller [8]. The Arduino code converts the sensor value to flow rate in liter/min and sends data to LCD for user monitoring. The Arduino continuously monitors this sensor value and depending on the set point value and threshold value set by the user sends the signal to the relay switch whether to keep the pump running or switch it off. Flow rate depends on two factors, the velocity of the fluid and the area of the nozzle [9]. In this case, the area of the nozzle remains constant, and the change in flow rate depends upon the velocity change due to the height of water in tank B.

The flow rate is related to the velocity and area of the nozzle by Equation (1):

$$Q = AV \tag{1}$$

where Q is the flow rate, A is the area of the nozzle (from tank B), and V is the velocity of the fluid.

The velocity is related to height by Equation (2):

$$V = \sqrt{2gh} \tag{2}$$

As the height of water in the tank changes, the velocity also changes causing a change in fluid flow rate which is constantly being monitored by the user through the microcontroller. A proteus simulation of the project was also done to verify the working of the project.

Let's describe the velocity at "point 2" mathematically.

From Figure 10, apply Bernoulli's equation between point 1 and point 2.

$$P_1 + 1/2\,\rho V_1^2 + \rho g h_1 = P_2 + 1/2\,\rho V_2^2 + \rho g h_2 \tag{3}$$

We have

$P_1 = 0$, tank B is exposed to the atmosphere from the top and bottom (through tank B by pipes).

$v_1 = 0$, because the fluid level drop is too slow.

$h_2 = 0$, we take height from bottom to top.

$P_2 = 0$, tank B is exposed to the atmosphere from the top and bottom (through tank B by pipes).

Hence, Bernoulli's equation becomes

$$\rho g h_1 = \frac{1}{2}\rho V_2^2 \tag{4}$$

$$V_2^2 = 2gh_1$$

$$V_2 = \sqrt{2gh_1} \tag{5}$$

"V_2" is the velocity of the fluid at point 2.

2.1. Circuit diagram (proteus)

A proteus simulation was done to verify the work of the project better. Figures 9, 11, and 12 show the complete circuitry diagram. All the necessary proteus libraries for Arduino UNO, relays, and Hall effect sensor were installed, and the circuit was designed.

1) Hardware connections: The hardware components utilized in the simulation include the following:

a. Flow sensor (3.3V)

Figure 9
Tank B points 1 and 2

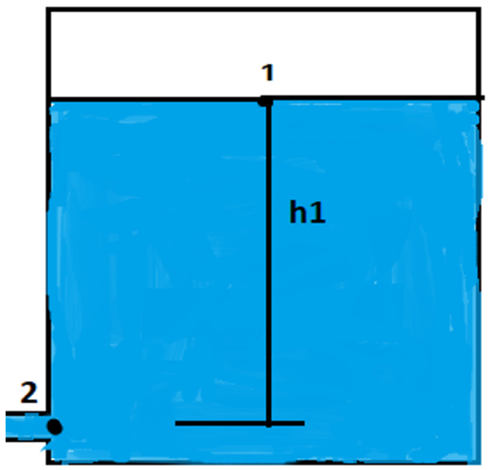
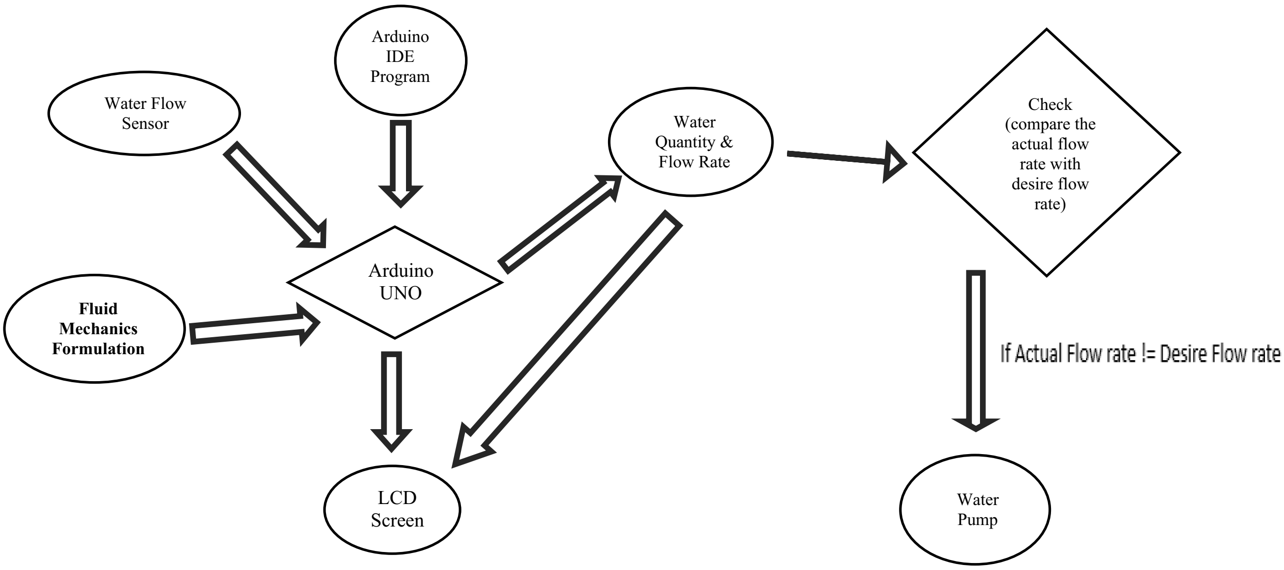


Figure 10
Block diagram of the entire fluid flow control process



b. Temperature sensor (Dallas DS18B20)
c. Relay module
d. LCD with I2C interface
e. Arduino microcontroller

2) LCD connection: SDA (Serial Data Line): Connected to the A4 (analog) pin on the Arduino board.

a. SCL (Serial Clock Line): Connected to the A5 (analog) pin on the Arduino board.
b. VCC: Connected to the 5V pin on the Arduino board.
c. GND: Connected to the GND pin on the Arduino board.

3) Temperature sensor (Dallas DS18B20) connection:
Signal (data): Connected to the same digital pin on the Arduino board as the flow sensor (pin 4).

a. VCC: Connected to the 5V pin on the Arduino board.
b. GND: Connected to the GND pin on the Arduino board.

4) Flow sensor connection:
Signal (data): Connected to a digital pin on the Arduino board (pin 4 in this case).

a. VCC: Connected to the 5V pin on the Arduino board.
b. GND: Connected to the GND pin on the Arduino board.

5) Relay connection: The relay module is connected to a digital pin on the Arduino board, specified as relaying in the code, which is set to pin 3. Typically, a relay module has three pins: VCC, GND, and IN (input). VCC is connected to the 5V pin on the Arduino board. GND is connected to the GND pin on the Arduino board. IN (input) is connected to the digital pin specified by the relay pin (pin 3 in this case) on the Arduino board.

2.2. Arduino program code explanation

The main logic of the system is developed in the program, and if the condition is used, that is, if the flow rate is less than the desired, the pump will become ON automatically and vice versa. Figure 13 represents the program executing process. The libraries for each component are installed first. We used the Arduino integrated development environment (IDE) to write the entire program for our control system. The code for the Hall effect sensor is based on its working mechanism (explained in the previous section); that is, the YF-S201 has some important factors defined by the manufacturer:

1) 6 interrupts/revolution (interrupts means pulses)
2) 8 Hz–8.2 Hz = 1 liter/min,
3) 486 pulses = 1 liter
4) Working flow rate 1–30 liters per minute

Hence, at the start of the program, we defined those factors. After that, we defined some variables following our program logic. Some important data types we used are as follows:

Volatile int: The keyword volatile is used to indicate to the compiler that a variable's value may change at any time outside the control of the program (e.g., by an interrupt service routine). An int is a data type representing a 16-bit signed integer.

Unsigned long: Unsigned long is a data type that represents a 32-bit unsigned integer. It can only hold nonnegative values.

The "millis ()" counts the time since the power on of the control system.

INPUT_PULLUP: This means that the microcontroller activates an internal resistor connected between the input pin and

Figure 11
Experimental setup of the entire process

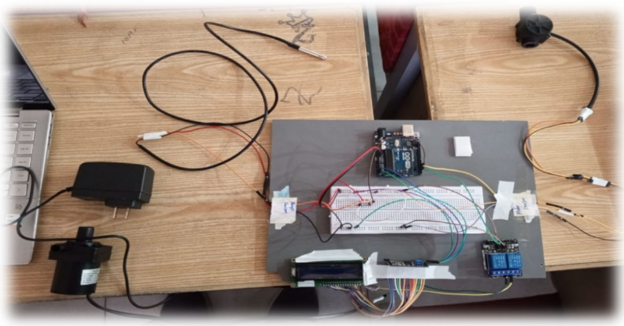
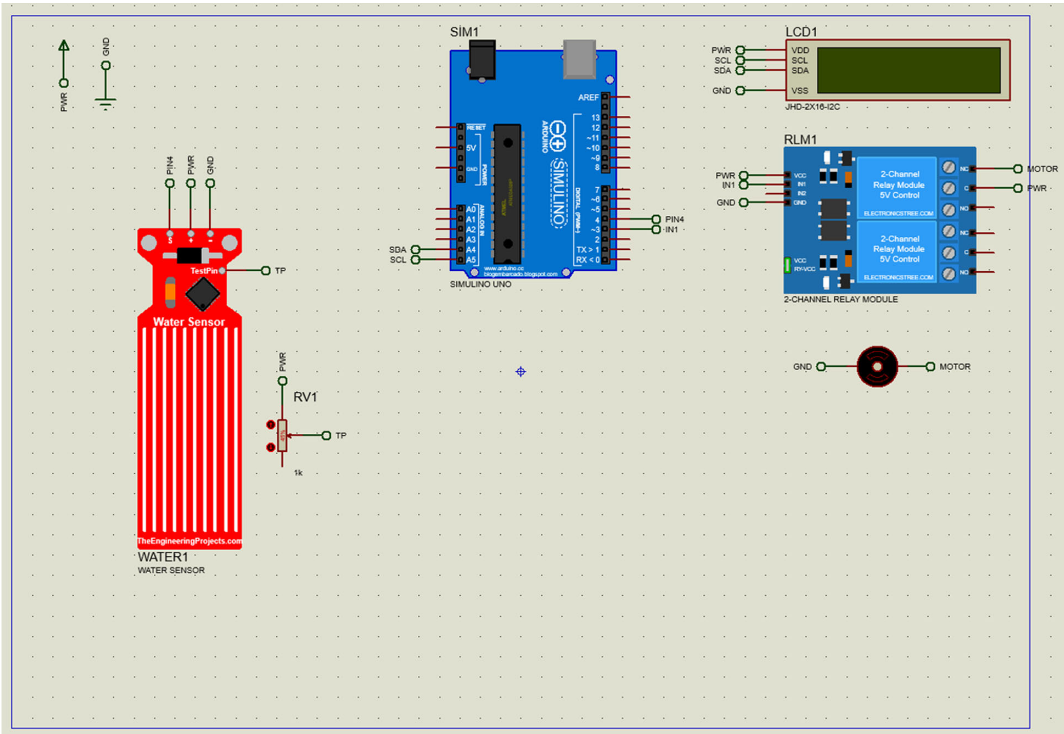


Figure 12
Circuit diagram – proteus



the power supply voltage (VCC). This resistor "pulls up" the voltage on the pin to VCC when the pin is not being actively driven low.

Interrupt mechanism: A mechanism that allows the Arduino to respond immediately to a specific event signaled by a Hall effect sensor without constantly checking its output in the main loop (void loop).

Under void setup:

In the main setup, we set the built-in LED for output and intrpin=2 as an input pin from the sensor (also do pullup).

attachInterrupt (): We set the program to report every 3 s because for calculation of flow rate (L/M) and total quantity gone to the controlled tank, we need a proper fixed value interval of time, and in our experimental, it is suitable but can vary accordance to the sensitivity of control.

Inside the **Char tbh[65]** is for serial print format and spacing.

As Arduino programming works in milliseconds, we convert the milliseconds to seconds.

void Pulse_cnt (): It is a user-defined function that counts the pulses concerning the **attachInterrupt ()** function; the variable rate_count stores the total number of pulses, and the variable **tot_cnt** stores the total number of pulses. At the end, turn the Boolean variable **"run_flag"** into true.

Under void loop (): We set the while loop with a condition that if the difference between the start time and the current time is less than or equal to the interval (3000 ms in this case), the loop is executed.

rate_count is set to zero after every cycle (after every 3 s).

In while loop:

1) If there is no interrupt(pulse) signal from the Hall sensor to the **attachInterrupt ()** function in **setup (),** the built-in LED will be off, and if there is an interrupt, the Pulse_cnt() function will execute, and the **Boolean run_flag** will become true.
2) If the **run_flag** is true, then the **if()** condition will execute, LED will become ON, **and run_flag** will be set to false because this allows the program to run a new iteration from the start **setup ()** to observe another interrupt from the sensor and execute **Pulse_cnt()** again. This entire mechanism will be repeated again and again till the difference becomes larger than the value of interval.

Note: Here, the **Pulse_cnt ()** function is called again and again whenever the interrupt is observed. The program searches for interrupts only during the provided time interval. When the difference reaches larger than the interval, the execution of the **while()** loop will stop, and the program after that will run, and we will get the desired output.

Let's assume our controlled tank demands **2<flow rate <=3:**

If (LperM<2): It is the condition that cares for the flow rate as it drops lower than 1, the pump will immediately ON and will fill tank 1 till the flow becomes greater than or equal to 3.

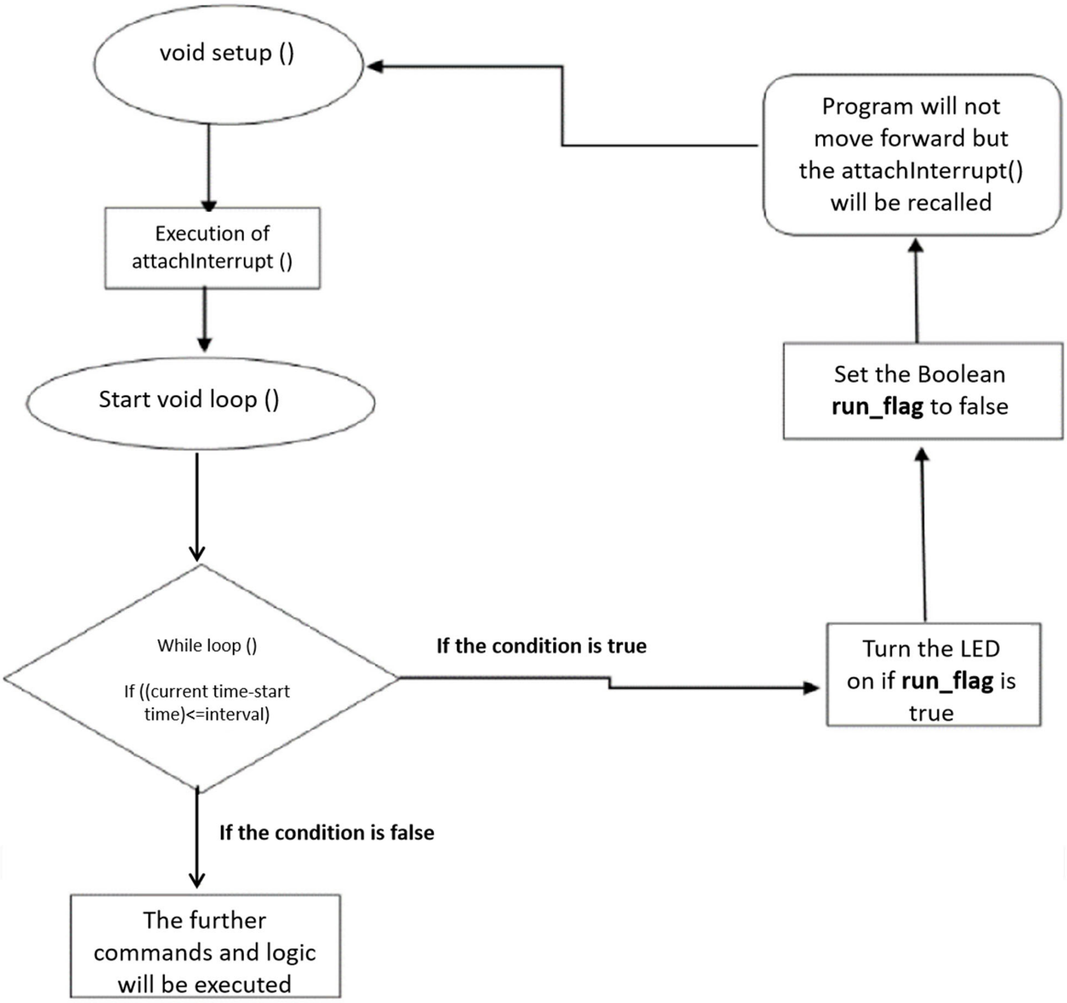
If (LperM>=3): It is the condition that cares when the flow rate becomes greater than or equal to 3.

In the end, the current time is set as the start time.

2.3. Arduino program code

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 4
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
// Flow Sensor 3.3V
// Set the LCD address
const int I2C_ADDR = 0x27; // Address might vary, check yours
// Set the LCD dimensions (columns x rows)
const int LCD_COLS = 16;
const int LCD_ROWS = 2;
float Celsius = 0;
float celsius=16;
// Create an LCD object. Provide the I2C address, columns, and rows.
LiquidCrystal_I2C lcd(I2C_ADDR, LCD_COLS, LCD_ROWS);
#define HzPerLiter 8.1
#define pulse2liter 486.0
#define interval 3000.0
// Define the pin to which the relay module is connected
const int relayPin = 3;
const byte intrPin = 2;
bool run_flag =false;
unsigned long startTime= millis();
int LperM=0;
volatile int rate_cnt=0;
unsigned long tot_cnt=0;
unsigned long tot_sec=0;
unsigned long tot_min=0;
unsigned long totLit=0;
char tbp[65];
void setup(){
sensors.begin();
Serial.begin(9600);
// Initialize the LCD
lcd.init();
lcd.backlight();
pinMode(LED_BUILTIN, OUTPUT);
pinMode(intrPin, INPUT_PULLUP);
attachInterrupt(digitalPinToInterrupt (intrPin),Pulse_cnt, FALLING);
Serial.println("Water flow = Approximate (No Exit. J-P S.)");
sprintf(tbp, "Report every %d seconds", round(interval/1000)); // Format comment
Serial.println(tbp);
Serial.println(" ");
// Set the relay pin as an OUTPUT
pinMode(relayPin, OUTPUT);
// Initially, turn off the pump
digitalWrite(relayPin, LOW);
}
void Pulse_cnt(){
rate_cnt++;
tot_cnt++;
run_flag= true;
}
void loop(){
while (millis() - startTime <= interval){
digitalWrite(LED_BUILTIN, LOW);
if (run_flag){
digitalWrite(LED_BUILTIN, HIGH);
run_flag= false;
}
}
tot_sec += interval/1000;
tot_min= round(tot_sec/60);
LperM = round(rate_cnt/(HzPerLiter*(interval)/1000)); // (pulses/interval)/(486*interval)
totLit= round(tot_cnt/pulse2liter);
Serial.print(" Instanaeouse Liters/Min : ");
Serial.println(LperM);
lcd.setCursor(0,0);
lcd.print("In L/M:");
lcd.print(LperM);
Serial.print("Total Liters since Startup : ");
Serial.print(totLit);
lcd.print(" T_L:");
lcd.print(totLit);
Serial.print("Total Minutes Since Startup: ");
Serial.println(tot_min);
sensors.requestTemperatures();
Celsius = sensors.getTempCByIndex(0);
celsius= celsius-0.01;
if (LperM <2){
digitalWrite(relayPin, LOW);
Serial.println("Pmp is ON");
Serial.print("T=");
Serial.print(celsius);
Serial.print("C");
lcd.setCursor(0,1);
lcd.print("Pmp ON,");
lcd.print("T=");
lcd.print(celsius);
lcd.print("C");
}
if (LperM>=3){
digitalWrite(relayPin, HIGH);
Serial.println("Pump is OFF");
Serial.print (celsius);
Serial.print("C");
lcd.setCursor(0,1);
lcd.print("Pmp OF,");
lcd.print(celsius);
lcd.print("C");
}
startTime= millis();
rate_cnt =0; }
```

Figure 13
Program/code flow chart



3. Results

Through programming logic, the value of flow is measured in Equation (6). The measured value is taken directly from the sensor reading, while the calculated value is calculated through the formula (Tables 1 and 2):

$$Q = AV \tag{6}$$

and

$$A = \pi r^2 \tag{7}$$

The radius of the pipe is 6.35 mm and π value is 3.14.

$A = \pi r^2 = 3.14(6.35)^2 = 126.612$ mm^2

And V is the velocity of the fluid. V depends upon the height of the fluid in the tank because the value of G is assumed to be constant.

$$V = \sqrt{2gh} \tag{8}$$

Value "h" is variable whose different values correspond to different velocities.

If the viscosity of the fluid is changed, its velocity will be changed because the mass flow rate is changed. Velocity depends upon the height for that case in which viscosity remains the same.

Equation (9) implies that if viscosity is changed, then the mass flow rate will be changed because density is changed due to which velocity will be changed due to which mass flow rate will be changed.

$$\dot{m} = \rho VA \tag{9}$$

ρ is the density of the fluid.

Table 1
Flow rate results for water

Serial number	Liquid	Calculated values (m^3/s)	Measured values (m^3/s)	Error	Percentage
1)	Water	2.34	2.30	0.04	1.709%
		2.96	2.98	0.02	0.675%
		2.11	2.18	0.07	3.3%

Table 2
Flow rate results for kerosene oil

Serial number	Liquid	Calculated values (m^3/s)	Measured values (m^3/s)	Error	Percentage
2)	Kerosene oil	2.47	2.48	0.01	0.4%
		2.58	2.51	0.07	2.7%
		2.76	2.81	0.05	1.8%

When liquid is changed, then its mass flow rate is changed basically which affects the velocity. Kerosene oil density is 720 kgm^{-3}, while the density of water is 1000 kgm^{-3}.

Second, the error can be due to human error, due to the sensor limitation.

The error can be removed if we precisely take the readings at moderate temperatures where evaporation does not occur and use a more detailed system like Raspberry Pi.

We take measured readings directly from the sensor by connecting the flow meter sensor to the analog pin of the Arduino, and the Arduino displays the analog value of the sensor on the LCD connected via the I2C module.

4. Discussion

This paper discusses the fabrication of a process control system involving the control of fluid flow rate by controlling the water level in a tank. The water level in the tank determines the flow rate of water through the nozzle showcasing an easy and effective way of maintaining a certain flow rate along with the desired water level in the tank. The components utilized to achieve this were Arduino UNO, YF201 Hall effect flow rate sensor, and a 12V DC centrifugal water pump. The purpose of this experiment was to showcase how it can be utilized in real-life industrial applications, especially in petrochemical industries, pharmaceutical industries, and power plants where there is a need to have control over the flow rate of fluid and water level in a tank or reservoir [1]. The system can be used in such cases to maintain water levels in the reservoir within the required boundary limits while also maintaining the fluid flow rate. However, there are some limitations to the model presented in this research; the system consists of three tanks (Figure 10). The flow is directly coming from tank B, and with time as the height of the liquid in tank B decreases, the flow rate decreases, and it doesn't remain the same as it was some time before. It is the main drawback of our model that the flow rate is not limited to a constant number like 2, 3, etc. The flow rate condition we applied is a range between two values like we set the flow rate condition we assumed is in between 2 and 3 (m^3/s), and if it goes below 2, the pump becomes ON automatically and fills the tank B till the height when the flow rate reach 3.

This process is repeated as per demand in tank C. Now this limit can be solved if we design tank B wisely as per fluid mechanics principles, its height and cross section must be very large, and the nozzle must be very small as compared to them. A constant value flow rate can be achieved for a long time, and as the flow rate decreases, the pump will fill tank B again. The second limitation of this research model is that it is generalized, as it is designed in a lab and we made assumptions about a lot of conditions; therefore, this model is not directly applicable to the special challenge or issue in the industry, but it can be modified and redesigned according to the specific problem. The specific problem could be attaining constant flow to maintain stoichiometry in the petrochemical industry or maintaining constant flow for fractional distillation in oil and gas industry or anything else.

It cannot be directly utilized in a large-scale industry where it is crucial to keep errors to the minimum and maintain other environmental parameters in control. For industrial integration, some further adjustments need to be made so that the proposed system can be used as an easy, efficient, and reliable method for fluid control and level management in industries. The major constraint with the proposed design is that the control process is slow and error prone as external environmental factors and interfering and modifying constraints have not been catered for when designing the system. This can be fixed by introducing PID control in our control system. PID control continuously calculates the error value between the measured value and the ideal value, applies correction, and provides the corrected value to the controller which is then processed for further control. Integration of PID control improves the accuracy of the system by removing errors accumulated during the process. Such controlled output is crucial in industries where errors occurring due to various environmental factors can result in a dangerous and harmful outcome [10]. For example, in pharmaceutical industries, petrochemical industries, and nuclear power plants, a slight change in the composition of elements in a mixture during the mixing process can ruin the entire process, in some cases resulting in a deadly outcome. Additionally, instead of controlling fluid flow by maintaining the water level in the tank, flow control valves can be utilized that may provide better results compared to the approach used in this paper to control the flow rate (Comparison of Level Control Configuration for Industry, n.d.). Furthermore, to improve the system proposed in this paper, the addition of a temperature and pressure sensor and actuators, for example, a boiler or heater, can also prove to be useful as most fluid control processes require the entire system to be kept under a controlled environment [11]. A controlled environment is where all the external factors that may influence the overall rate or quality of fluid, such as temperature and pressure, are kept constant or stable as per the conditions demand. To achieve this, the integration of various sensors and actuators that control such parameters is necessary to ensure proper management of processes in industries. Further improvements such as the use of higher-sensitivity equipment, better actuators, and a high memory and clock speed microprocessor can have a significant impact on the overall performance of the system [12].

5. Future Scope

In the vast realm of industries, ensuring precise fluid flow is crucial for various applications, ranging from chemical production to nuclear reactor safety. However, advancements in technology and innovative approaches are paving the way for an exciting future in fluid flow control [13].

Imagine utilizing compact yet powerful microcontrollers like Arduino to develop cost-effective control systems tailored to specific industry needs. These systems can offer customization and scalability, providing adaptable solutions for both large-scale factories and smaller operations. Moreover, sophisticated control algorithms such as PID hold promise for enhancing control accuracy and efficiency.

Automation is emerging as a significant driver of progress in fluid flow control. By integrating technologies like IoT, AI, and robotics, industries can achieve unprecedented levels of precision and efficiency. For instance, in manufacturing settings, automated systems can continuously monitor and adjust fluid flow parameters in real time, ensuring consistent product quality. Similarly, in healthcare, automated controls can precisely regulate medication dosages, enhancing patient safety and care [14].

Even agriculture stands to benefit from automation in fluid management. Imagine smart irrigation systems equipped with sensors that analyze soil moisture levels and adjust water flow, accordingly promoting optimal crop growth while conserving water resources [15, 16].

In essence, the future of fluid flow control is characterized by innovation, efficiency, and sustainability.

Through the integration of advanced technologies and novel approaches, industries are poised to unlock new levels of precision and performance in managing fluid processes, ultimately leading to improved productivity and environmental responsibility [17, 18].

6. Conclusion

The study demonstrates the significance of fluid flow control systems in various industries, especially in the context of Industry 4.0, where automation and precision are essential for optimizing processes. This paper presents a detailed overview of the hardware and software components used in the control system, emphasizing its cost-effectiveness and efficiency.

Through the physical setup and simulation, the authors show how the system effectively controls fluid flow rates by monitoring water levels and adjusting pump operation accordingly. However, it is noted that the proposed system may have limitations in large-scale industrial applications due to its slow response time and susceptibility to external environmental factors.

To address these limitations and enhance the system's performance, the authors suggest incorporating PID control, which can improve accuracy by continuously correcting errors. Additionally, they propose the integration of temperature and pressure sensors, as well as actuators like boilers or heaters, to create a more controlled environment for fluid processes [19, 20].

Furthermore, the authors recommend exploring alternative methods such as flow control valves for better flow rate management and investing in higher-sensitivity equipment, improved actuators, and advanced microprocessors to enhance overall system performance [21].

In summary, while the developed control system shows promise for industrial applications, further refinement and enhancements are necessary to meet the stringent requirements of large-scale industries, particularly in terms of accuracy, speed, and environmental robustness.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data are available from the corresponding author upon reasonable request.

Author Contribution Statement

Shamsher Ali: Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Writing – review & editing, Supervision, Project administration. **Syed Muhammad Daniyal Gillani:** Methodology, Investigation, Writing – original draft. **Syed Muhammad Zain:** Formal analysis, Visualization. **Sudais Ahamd Khan:** Resources.

References

[1] Popescu, D., Mircioiu, M., & Moise, A. (2011). Numerical control for petrochemical installations. In *15th International Conference on System Theory, Control and Computing*, 1–6.

[2] Vogt, E. T. C., & Weckhuysen, B. M. (2015). Fluid catalytic cracking: recent developments on the grand old lady of zeolite catalysis. *Chemical Society Reviews*, *44*(20), 7342–7370. https://doi.org/10.1039/c5cs00376h

[3] Dharmaiah, G., Mebarek-Oudina, F., Sreenivasa Kumar, M., & Chandra Kala, K. (2023). Nuclear reactor application on Jeffrey fluid flow with Falkner-skan factor, Brownian and thermophoresis, nonlinear thermal radiation impacts past a wedge. *Journal of the Indian Chemical Society*, *100*(2), 100907. https://doi.org/10.1016/j.jics.2023.100907

[4] Ueyama, A. (2019). *Want to know more! Basics of thermo-fluid analysis 12: Chapter 3 flow 3.2.1 velocity, speed, flow rate, and acceleration*. Retrieved from: https://www.cradle-cfd.com/media/column/a145

[5] Kumar, S., Chandra, S. K., Shukla, R. N., & Panigrahi, L. (2023). Industry 4.0 based machine learning models for anomalous product detection and classification. In *2022 OPJU International Technology Conference on Emerging Technologies for Sustainable Development*, 1–6. https://doi.org/10.1109/otcon56053.2023.10114045

[6] Boynton, W. V. (1984). Cosmochemistry of the rare earth elements: meteorite studies. *Developments in Geochemistry*, *2*, 63–114. https://doi.org/10.1016/b978-0-444-42148-7.50008-3

[7] Jan, F., Min-Allah, N., Saeed, S., Iqbal, S. Z., & Ahmed, R. (2022). IoT-based solutions to monitor water level, leakage, and motor control for smart water tanks. *Water*, *14*(3), 309. https://doi.org/10.3390/w14030309

[8] Electrocredible. (2022). *How YF-S201 Water Flow Sensor Works? Pinout, Interfacing Explained*. Retrieved from: https://electrocredible.com/water-flow-sensor-pinout-working-explained/

[9] OpenStax. (2016). *Flow rate and its relation to velocity*. Retrieved from: https://pressbooks-dev.oer.hawaii.edu/collegephysics/chapter/12-1-flow-rate-and-its-relation-to-velocity/

[10] Garud, K. S., Tai, L. D., Hwang, S.-G., Nguyen, N.-H., & Lee, M.-Y. (2023). A review of advanced cooling strategies for battery thermal management systems in electric vehicles. *Symmetry*, *15*(7), 1322. https://doi.org/10.3390/sym15071322

[11] Hossain, A., Islam, S., Rahman, Z., Mohammad Salahuddin, A. Z., & Mollah, A. S. (2019). An intelligent flow control system of coolant for a water reactor based cooling tower. *Energy Procedia*, *160*, 566–573. https://doi.org/10.1016/j.egypro.2019.02.207

[12] Mishra, D., & Samantara, T. N. (2024). Current advancements in fluid-flow problems: A brief review. *World Journal of Advanced Research and Reviews*, *21*(1), 393–398. https://doi.org/10.30574/wjarr.2024.21.1.2718

[13] Kim, J., & Bewley, T. R. (2007). A linear systems approach to flow control. *Annual Review of Fluid Mechanics*, *39*(1), 383–417. https://doi.org/10.1146/annurev.fluid.39.050905.110153

[14] Peter, O., Pradhan, A., & Mbohwa, C. (2023). Industrial internet of things (IIoT): opportunities, challenges, and requirements in manufacturing businesses in emerging economies. *Procedia Computer Science*, *217*, 856–865. https://doi.org/10.1016/j.procs.2022.12.282

[15] Vallejo-Gómez, D., Osorio, M., & Hincapié, C. A. (2023). Smart irrigation systems in agriculture: A systematic review. *Agronomy*, *13*(2), 342. https://doi.org/10.3390/agronomy13020342

[16] Keck, J. C. (1990). Rate-controlled constrained-equilibrium theory of chemical reactions in complex systems. *Progress in Energy and Combustion Science*, *16*(2), 125–154. https://doi.org/10.1016/0360-1285(90)90046-6

[17] Shi, Y., Wang, P., Mu, H., Kou, H., Wu, A., & Su, L. (2023). Advances of interface, flow, and stress control for VB crystal growth: An overview. *Progress in Crystal Growth and Characterization of Materials*, *69*(2–4), 100605. https://doi.org/10.1016/j.pcrysgrow.2023.100605

[18] Vanags, E., Āboliņš, A., & Cābulis, U. (2023). Lipase catalyzed self-epoxidation of tall oil fatty acids in batch and continuous flow conditions. *Journal of Polymers and the Environment*, *31*(5), 2166–2176. https://doi.org/10.1007/s10924-022-02739-0

[19] Ahmad, N. N., Nik Ghazali, N. N., Rani, A. T. A., Othman, M. H. D., Kee, C. C., Jiwanti, P. K., . . . , & Wong, Y. H. (2023). Finger-actuated micropump of constant flow rate without backflow. *Micromachines*, *14*(4), 881. https://doi.org/10.3390/mi14040881

[20] Djalilov, A., Nazarov, O., Sobirov, E., Tasheva, U., Abdunabiyev, J., & Urolov, S. (2023). Research of water flow measuring device based on Arduino platform. *E3S Web of Conferences*, *401*, 04039. https://doi.org/10.1051/e3sconf/202340104039

[21] Han, Z., Hishida, S., Omata, N., Matsuda, T., Komori, R., & Chen, L. C. (2023). Feedback control of electrospray with and without an external liquid pump using the spray current and the apex angle of a Taylor cone for ESI-MS. *Analytical Chemistry*, *95*(28), 10744–10751. https://doi.org/10.1021/acs.analchem.3c01768

How to Cite: Ali, S., Gillani, S. M. D., Zain, S. M., & Khan, S. A. (2025). Controlled Fluid Flow Without Controlling Pump Through Arduino. *Archives of Advanced Engineering Science*, *3*(4), 241–251. https://doi.org/10.47852/bonviewAAES42022901