DESIGN OF AN AUTONOMOUS SMART SHOWER WITH SENSORS AND ACTUATORS

Tareq Khan

School of Engineering Technology, Eastern Michigan University, Michigan, Ypsilanti, United States.

ABSTRACT

At the beginning of taking a shower, the user needs to manually adjust a rotational handle or the ratio of cold and hot water to get the desired water temperature and the flow rate. In this paper, a temperature and flow rate sensor feedback smart shower is proposed which takes the target water temperature and flow rate from the user as input, and then automatically adjusts the ratio of the cold and the hot water during the shower to keep the temperature and flow rate fixed - even though there is fluctuation of supply water temperature and pressure. The proposed system contains distance sensor and automatically turns off the shower when the user is away for soaping or shampooing. The system generates a report on water usage and shower time – to promote awareness on saving water. An embedded system based prototype of the proposed shower has been developed and tested.

KEYWORDS

Ball valve, Embedded system, Finite State Machine, Sensor, Servo motor.

1. INTRODUCTION

In the United States, most of the people use the single rotational handle where they have to fine-tune the knob by rotating it to get a comfortable shower temperature. In this process, the user runs the water, keep adjusting by touching the water until it reaches the desired temperature. This manual process to get the target temperature is time-consuming and not efficient.

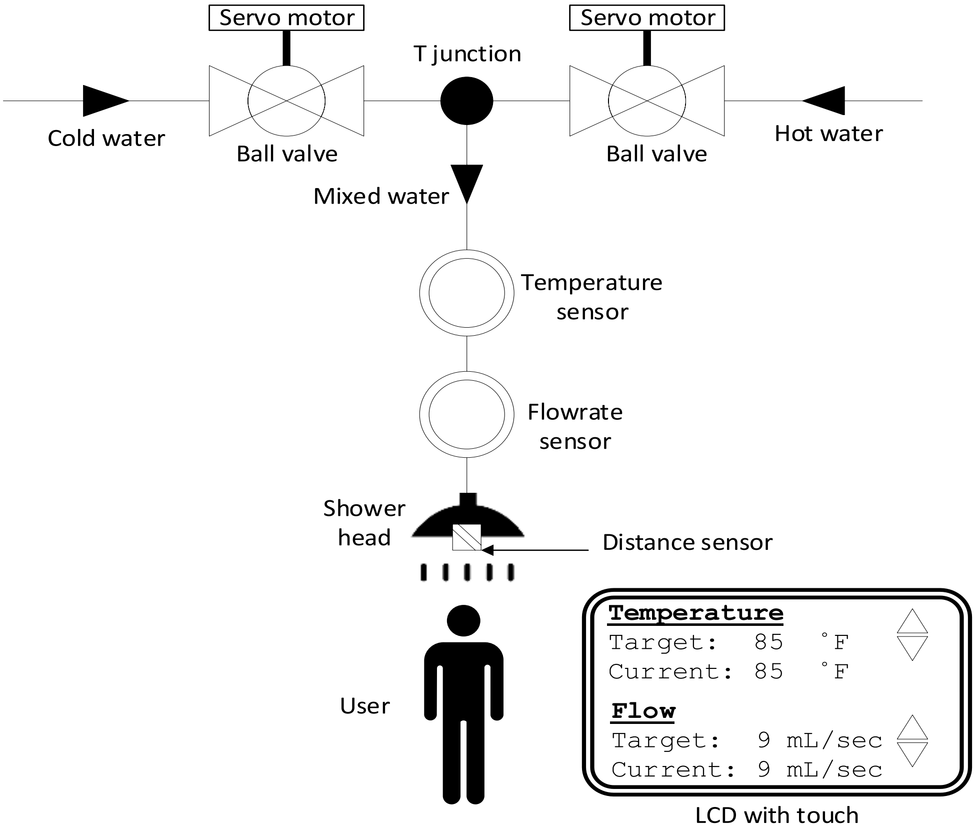
In this paper, an embedded system based closed-loop autonomous shower is proposed. The overall block diagram of the proposed shower is shown in Fig. 1. In the proposed system, the user sets the target temperature and the flow rate on a touchscreen-based liquid crystal display (LCD). When the *Start* button is pressed, the system automatically adjusts the ratio of cold and hot water to get the target temperature and flow rate. Two servo motor operated ball valves [1] are used to control the hot and cold water ratio, as shown in Fig. 1. The advantage and significance of the proposed system are mentioned below.

- In the proposed system, the display shows the current temperature of the water, thus the user can start using the shower when it has a comfortable temperature – avoiding a sudden burst of hot or cold water on the body.
- During the shower, the temperature and the flow rate of the water may fluctuate and it can be an uncomfortable shower experience [2]-[4]. When domestic water lines are shared by

DOI : 10.5121/ijesa.2018.8301

multiple apartment units in a building, temperature and pressure are affected by other users in the building, especially during peak use times. If the cold water supply is reduced by other uses – such as toilet flashes, dishwasher or laundry, the cold water pressure in the shower goes down. If the cold water decreases - the shower water gets too hot, and vice-versa. If a water heater is shared among several units, that can also cause drastic temperature swings as the hot water is being shared among many units, with both hot and cold taps being turned on and off throughout the building. The proposed system continuously adjusts the ratio during the entire shower session, so the output temperature and flow remain fixed - even though the supply fluctuates. This produces a comfortable showing experience - without the hassle of manual adjusting.

- In an average American home, showers are typically the 3rd largest water use after toilets and laundry. The average water usage is 65.1 liters (L) and lasts for 8.2 minutes at average flow rate of 7.9 liters per minute (L/min). To save water, it is recommended to turn off the shower while soaping or shampooing [5]. The proposed shower contains a distance sensor to detect the presence of the user directly under the shower. The system automatically *pauses* and *resumes* the shower to its previous temperature and flow settings when the user is away for soaping/shampooing or near the shower namely. This feature can help to save a significant amount of water.
- To save water, it is recommended to shorten the showering session to 5 minutes [5]. Behavioral research shows that self-observation and monitoring can cause the desired behavior [6]. The proposed system produces a report at the end of the shower session showing the total water usage and shower duration. This report will make the user conscious of saving water and reduce the time spent during the shower as much as possible.



Temperature
Target: 85 °F
Current: 85 °F

Flow
Target: 9 mL/sec
Current: 9 mL/sec

Figure 1. The proposed smart showering system.

2. RELATED WORKS

To keep the temperature and flow fixed, one mechanical solution is to use thermostatic valves [7][8]. Inside this valve, a wax element expands and contracts as it reacts to heat and thus controls the hot and cold water ratio. However, the user can not set a particular temperature in Fahrenheit unit or flow rate in L/min unit in this system as it does not contain digital sensors and displays. It is also unable to show current water temperature and keep track of water usage during the shower. Microprocessor based embedded systems and wireless networks are now becoming popular to monitor home appliances [9], weather [10], and even patients [11]. In [12], an adaptive assistive smart shower system is proposed which is capable of detecting the user's abilities and disabilities using a classifier, and then provides the necessary aids automatically. The work in [13] proposes a smart water heating scheme to ensuring the minimal waste of water during shower using an open-source Asterisk-based IP-PBX on cloud computing technology. In [14], a case study of identifying hot water fixtures including the kitchen sink, bathroom sink, and shower by using support vector machines (SVMs) is reported.

The commercial product in [15], lights up the water spray with different colors depending on the amount of water used – to promote awareness of water usage. In [16], an LCD shows the current temperature; and LCD’s background and water spray change colors depending on the temperature. The products in [17] and [18] show both water usage and temperature on a display. In [19], the current temperature is shown on a LED display. However, [15]-[19] do not have any automatic control to adjust the temperature and flow. The work in [20] uses a distance sensor and changes water flow levels based upon the user’s location in the shower. It also shows water usage and shower time by connecting with a Smartphone app using Bluetooth. The smart shower in [21] connects with Smartphone using Wi-Fi and allow users to start the shower from phone to have the shower the desired temperature pause at initial start-up. It displays the current temperature and also adjusts temperature even if the supply water fluctuates. However, it cannot control or adjust the flow of the water, it can only turn on and off the flow.

Compared with the related works, the proposed smart shower takes the temperature and flow rate from the user as input and automatically adjusts both the temperature and the flow even if the supply water fluctuates. The proposed shower shows the current temperature and flowrate in real time. Using distance sensor, the system automatically turns off the shower when the user is away for soaping or shampooing – to save water. It also generates reports on water usage and shower duration to promote awareness on saving water.

3. METHODOLOGY

3.1. Calculating the Pulse Width for Servo Motor w.r.t. Valve Open Area

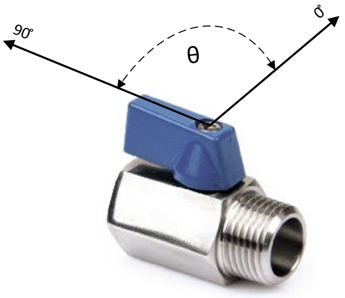


Figure 2. Ball valve and the angle of the handle.

In this project, two servo motor operated ball valves [22] are used to control the ratio of hot and cold water. A ball valve is a form of quarter-turn valve which uses a hollow, perforated and pivoting ball to control flow through it. As shown in Fig. 2, when the valve handle is 0° - the valve is fully closed, and when it is 90° - it is fully open. Depending upon the alignment angle of the valve handle, *θ*, the open area inside the valve can be changed and flowrate can be controlled.

The open area inside the valve can be modeled using the intersection area of two overlapping circles as shown in Fig. 3. The radius of circle A and circle B is *r*, and the distance from the centers of the two circles is *d*. The open area in the valve is the intersection of the circles – shaded in gray. When *d* is 0 - the circles completely overlap and the valve is fully open; when *d* is 2*r* – the circles are completely disjoint and the valve is fully closed. The linear relationship between *θ* and *d* is shown in Fig. 4 and (1).

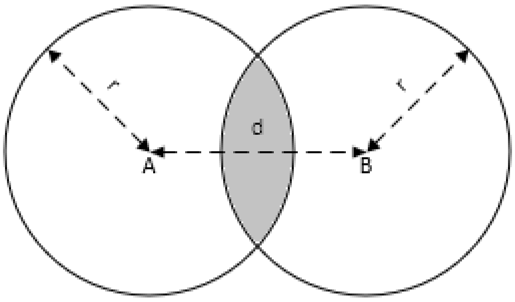


Figure 3. Intersecting circles to model the open area of the valve.

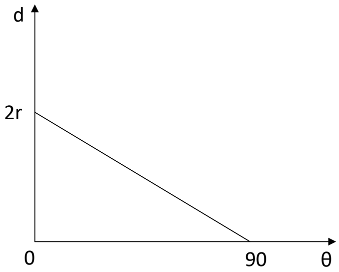


Figure 4. The relationship between valve handle angle, *θ*, and the distance between the circles, *d*, as shown in Fig. 3

$$d = \frac{-2r}{90}\theta + 2r \qquad (1)$$

Now, the intersecting shaded area of the circles, *Area*, w.r.t. *d* as shown in Fig. 3 can be calculated using (2) [23].

$$Area = 2r^2\cos^{-1}\left(\frac{d}{2r}\right) - \frac{d}{2}\sqrt{4r^2 - d^2} \qquad (2)$$

The valve handle is controlled using a servo motor [19] that is mounted at the top of the valve handle. From the top view perspective of the servo motor, it needs to rotate 90 degree clockwise to bring the valve handle from the fully closed to the the fully open position. The required pulse duration, *W*, for the servo motor w.r.t. θ is expressed in (3) where W_{max} is the pulse duration for θ to be at 0° and W_{min} is the pulse duration for θ to be at 90° position.

$$W = W_{\max} - \frac{(W_{\max} - W_{\min})\theta}{90} \qquad (3)$$

From (1), (2), and (3), a lookup table is generated to convert *Area* to *W*. A plot of the lookup table is shown in Fig. 5, where *r* = 10, W_{max} = 1500µs and W_{min} = 700µs. For r = 10, the maximum overlapping area is $\pi r^2 = 3.14 \times 10^2 = 314$. The lookup table converts each integer values of *Area* from 0 to 314 to their corresponding pulse duration, *W*. Using this table, *Area* can be set properly by rotating the servo motor in the exact angle.

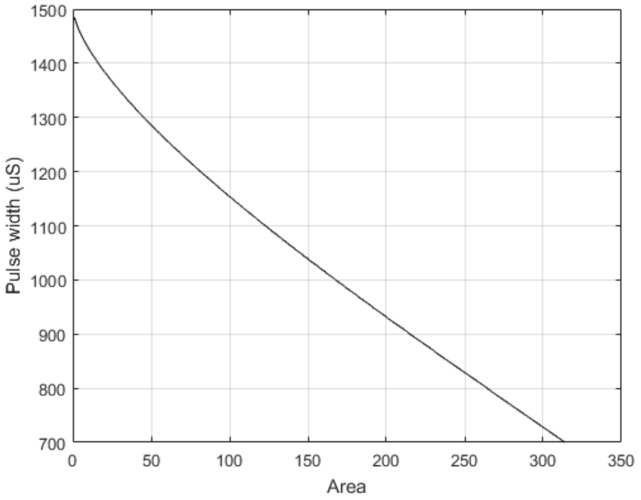


Figure 5. The plot of valve open area, *Area*, and pulse width, *W*, for the servo motor.

3.2. Adjustment Algorithms

The microcontroller of the system reads the current temperature, `CTemp,` and current flowrate, `CFlow,` of the water using the sensors and compares them with the target temperature, `TTemp`, and target flowrate, `TFlow` namely; and adjusts the valves to get the desired water temperature and the flow rate. When adjusting, the temperature adjustment has higher priority than flowrate adjustment. The pseudocode of the adjustment logic structure is shown in Fig. 6. This segment of the code runs in repeatedly under a loop at showering state.

```
if (abs (CTemp - TTemp) > TEMP_DIFF_THRESHOLD)
    if (isCTempUpdated)
        AdjustTemp ()
        isCTempUpdated := 0

else if (abs (CFlow - TFlow) > FLOW_DIFF_THRESHOLD)
         if (isCurrentFlowUpdated)
             AdjustFlow ()
             isCFlowUpdated := 0
```

Figure 6. Pseudocode of the adjustment logic structure.

3.2.1. Temperature Adjustment Algorithm

In Fig. 6, the TEMP_DIFF_THRESHOLD is the tolerable difference between current and target temperature and it has the typical value of 0. The current temperature, CTemp, is updated every after TEMP_ADJ_DELAY_MS = 250 ms delay and isCTempUpdated flag is set. This delay is required as the mixing of water takes some time, and without the delay - it will keep changing valve area too fast before the water reaches the sensor area. The AdjustTemp () function adjusts the valve open area as shown in Fig. 7. and the isCTempUpdated flag is then reset.

```
if (CTemp > TTemp)
     if (AreaHot > MIN_HOT_AREA)
       AreaHot := AreaHot - 1
       RotateServoHot (ToPWM_Width(AreaHot))

     else if (AreaCold < MAX_COLD_AREA)
         AreaCold := AreaCold + 1
         RotateServoCold (ToPWM_Width(AreaCold))


if (CTemp < TTemp)
     if (AreaHot < MAX_HOT_AREA)
       AreaHot := AreaHot + 1
       RotateServoHot (ToPWM_Width(AreaHot))

     else if (AreaCold > MIN_COLD_AREA)
         AreaCold := AreaCold - 1
         RotateServoCold (ToPWM_Width(AreaCold))
```

Figure 7. Pseudocode of the temperature adjustment algorithm.

In Fig. 7, AreaHot and AreaCold variables stores the valve open area of the hot water valve and cold water valve namely. The maximum area for hot water valve, MAX_HOT_AREA,and for cold water valve, MAX_COLD_AREA was set to 314. The minimum area for hot water valve, MIN_HOT_AREA, and for cold water valve, MIN_COLD_AREA was set to 1. The ToPWM_Width () function takes the *Area* as an argument and returns the pulse width, *W*, to rotate the servo motor using the lookup table as shown in Fig. 5. RotateServoHot () and RotateServoCold () functions take *W* as the input argument and rotate the hot water controlling servo motor, *S_Hot*, and cold water controlling servo motor, *S_Cold*, according to the pulse width namely. Assuming same pressure in hot and cold water supply, the temperature of the mixed water can be expressed using (4), where *TempHot* and *TempCold* are the temperatures of the hot and cold water supply namely.

$$CTemp = \frac{AreaHot \times TempHot + AreaCold \times TempCold}{AreaHot + AreaCold} \qquad (4)$$

When water is too hot, i.e. current temperature is higher than the target temperature, then the temperature needs to be reduced. To do that, the AreaHot is decreased by rotating *S_Hot*. However, if the AreaHot reaches the MIN_HOT_AREA, then AreaCold is increased and the *S_Cold* is rotated accordingly. In the case, AreaCold reaches the MAX_COLD_AREA, further adjustments cannot be possible.

3.2.2. Flow Adjustment Algorithm

As shown in Fig. 6, the flow is adjusted only after the temperature is adjusted, as temperature adjustment has higher priority. It has a similar structure of temperature adjustment. The `FLOW_DIFF_THRESHOLD` is the tolerable difference between current and target flow and it has the typical value of 0. The current flow rate, `CFlow`, is updated every after `FLOW_ADJ_DELAY_MS = 250 ms` delay and `isCFlowUpdated` flag is set. This delay is required as it takes some time to get the effect of valve area change and recalculate the flowrate. The `AdjustFlow ()` function adjusts the valve open area as shown in Fig. 8. and the `isCFlowUpdated` flag is then reset.

```
if (CurrentFlow > TargetFlow)
     if (AreaHot < AreaCold)
       if (AreaHot > MIN_HOT_AREA)
         N_AreaHot := AreaHot - 1
         N_AreaCold := round (N_AreaHot * AreaCold / AreaHot)

         if (N_AreaCold >= MIN_COLD_AREA)
           AreaHot := N_AreaHot
           AreaCold := N_AreaCold

           RotateServoHot (ToPWM_Width(AreaHot))
           RotateServoCold (ToPWM_Width(AreaCold))

     else      //AreaCold is smaller
       if (AreaCold > MIN_COLD_AREA)
         N_AreaCold := AreaCold - 1
         N_AreaHot := round (N_AreaCold * AreaHot / AreaCold)

         if (N_AreaHot >= MIN_HOT_AREA)
           AreaHot := N_AreaHot
           AreaCold := N_AreaCold

           RotateServoHot (ToPWM_Width(AreaHot))
           RotateServoCold (ToPWM_Width(AreaCold))
```

Figure 8. Pseudocode of the flow rate adjustment algorithm when current flow is greater than target flow.

To change the flow rate without affecting the current temperature of the mixed water, the ratio of the hot and cold water needs to remain fixed. The following relation in (5) needs to be maintained when changing flowrate:

$$\frac{AreaHot_{new}}{AreaCold_{new}} = \frac{AreaHot}{AreaCold} \qquad (5)$$

The pseudocode of the flow rate adjustment algorithm when current flow is greater than target flow is shown in Fig. 8. In this case, the flow rate needs to be reduced to get the target flow rate. The smaller area between `AreaHot` and `AreaCold` is first chosen – its value is reduced by 1. For instance, when `AreaHot` is smaller than `AreaCold`, `AreaHot` is reduced by 1 and stored in the new variable `N_AreaHot`. Now, to keep the temperature fixed, the new cold area, N_AreaCold is calculated using (5). After checking with its minimum value, the new variables are assigned to AreaHot and AreaCold; and then motors are rotated. In the case, when AreaCold is smaller than AreaHot, the similar logic structure is implemented.

The flow rate adjustment algorithm when current flow is lesser than target flow is implemented in a similar way using complementary logic. In that case, the flow rate is increased to get the target flow rate without affecting the temperature.

4. SYSTEM ARCHITECTURE

A microcontroller based embedded system is implemented to design and develop the proposed autonomous shower. The hardware and firmware part of the system are briefly described below.

4.1. Hardware

The overall block diagram of the microcontroller based hardware unit is shown in Fig. 9. A brief description of each component is given below.

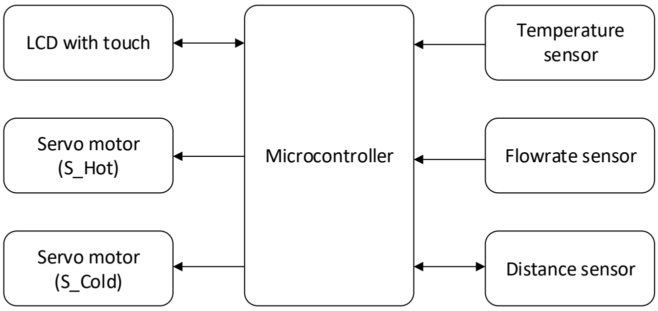


Figure 9. Block diagram of the hardware unit.

4.1.1. Microcontroller

An 8-bit AVR microcontroller (ATmega328) [24] is used as the processing unit. It has 32 kB of program memory, 2 kB byte of SRAM, and 1 kB byte of EEPROM. It also has programmable general purpose input/output (GPIO) lines, timers, universal synchronous/asynchronous receiver/transmitter (USART), analog to digital converters (ADC), external interrupt pins, and pulse width modulation (PWM) channels.

4.1.2. Temperature sensor

A voltage divider circuits as shown in Fig. 10 is implemented where R1 is a thermistor [25] and R2 is a fixed 10kΩ resistor. The thermistor has negative temperature coefficient and its resistance value at 25° C is 10 kΩ. It is used to read the temperature of the mixed water as shown in Fig. 1. The proportional voltage between the junction of the resistors is fed to an ADC port of the microcontroller. A 0.1 uF capacitor is connected across R2 to reduce noise.

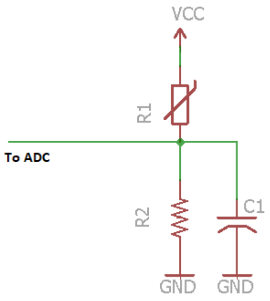


Figure 10. Voltage divider circuit for the thermistor.

4.1.3. Flowrate sensor

A hall effect based flowrate sensor [26] is used to measure the flow rate of the shower water. It provides a square wave pulse output whose frequency is proportional to the flow rate. Its output pin is interfaced with an external interrupt pin of the microcontroller. The sensor can measure flowrate up to 500 mL/sec.

4.1.4. Distance sensor

The ultrasonic distance sensor, HC-SR04 [27], is used to measure the proximity of the user under the shower. This economical sensor provides 2cm to 400cm of non-contact measurement functionality. The sensor module includes an ultrasonic transmitter, a receiver and a control circuit. The sensors Trig (Trigger) pin is connected with an output GPIO pin and the Echo pin is connected with an input GPIO pin of the microcontroller.

4.1.5. LCD with touch

A 2.4″ graphical LCD [28] capable of displaying 320 × 240 pixels using 262,144 colors is used to display the real-time temperature and flow rate and to display reports and graphs. The display also contains resistive touch area, enabling advanced and interactive user interface. The LCD connects to the microcontroller using hardware USART.

4.1.6. Servo motor

Servos motors allow to precisely control the rotation angle of the output shaft. Two high torque 180-degree counter-clockwise rotation servo motors [29] are used to rotate and control the ball valve handles of the hot and cold water supply as shown in Fig. 1. The servo can take pulse width input from 700µs to 2300µs at its control pin for the rotation angle of 0 to 180 degree namely. The control pin of each servo is interfaced with a PWM channel pin of the microcontroller.

4.1.7. Power supply

For powering the sensors, LCD and microcontroller - a DC power adaptor of 9V output having a capacity of 500mA current is used. Using the onboard power regulator chips of the microcontroller board, 5V and 3.3V supplies are also generated. As one servo can consume a maximum stall current of 1100mA, a separate DC power adaptor of 5V output having a capacity of supplying 2000mA current is used as the power supply for the servos.

4.2. Firmware

The firmware is built on two layers – the driver layer and the application layer. The application layer access the hardware by calling the functions of the driver layer. A brief description of each layer is given below.

4.2.1. Driver layer

The driver layer consists of low-level firmware for accessing different hardware peripherals. The flow rate sensor's pulse output pin is connected with an external interrupt pin of the microcontroller. An interrupt service routine (ISR) is called at the falling edge of the pulse. The ISR routine calculates the pulse period in millisecond unit by subtracting the previous pulse arrival tick time from the current pulse arrival tick time. The period is then added in a circular fast-in fast-out (FIFO) buffer of size 8. The ISR routine also increments the variable `TotalFlowPulse` by one - to keep track of total water usage in the shower session. Some of the important driver functions and their brief descriptions are shown in Table 1.

Table 1. Driver functions and their brief descriptions.

Driver function name	**Description**
`ReadTemp ()`	Returns the temperature value of the water in Fahrenheit unit. The function reads the ADC pin, converts the ADC value to temperature, and also sets the flag `isCTempUpdated` every after `TEMP_ADJ_DELAY_MS.`
`ReadFlow ()`	Returns the flowrate of the water in mL/sec unit. Every after `FLOW_ADJ_DELAY_MS, t`he function calculates the average of the pulse periods in millisecond (`Tavg)` from the circular buffer, calculates the flow rate as `round(1000 * 1000 / (Tavg * FLOW_SENSOR_PULSE_PER_LITER)),` and sets the flag `isCFlowUpdated.`
`GetDistance ()`	Returns the distance of the nearest object from the distance sensor in Centimetre unit. The function sends a 10μs pulse at the *Trig* pin, measurers the *Echo* pin's pulse high time, and then convert the pulse travel time to distance in cm.
`IsAway ()`	Returns TRUE if the distance of the user from the sensor is more than AWAY_DISTANCE_CM continuously for the duration of AWAY_WAIT_MS.
`IsNear ()`	Returns TRUE if the distance of the user from the sensor is less than AWAY_DISTANCE_CM continuously for the duration of NEAR_WAIT_MS.
`ToPWM_Width (int area)`	Returns the pulse width, *W_us*, for the servo motors for a given `area` from the lookup table as shown in Fig. 5. The lookup table is stored in the program memory of the microcontroller to save RAM space.
`RotateServoHot (int W_us)`	Sets the pulse width, `W_us`, for *S_Hot* servo motor and rotates it accordingly.

`RotateServoCold (int W_us)`	Sets the pulse width, `W_us`, for *S_Cold* servo motor and rotates it accordingly.
`ReadButtonPress()`	Read the touch input from LCD. Depending on the coordinates of the touch, the function returns NO_PRESSED, TEMP_UP_PRESSED, TEMP_DN_PRESSED, FLOW_UP_PRESSED, FLOW_DN_PRESSED, START_PRESSED, PAUSE_PRESSED, STOP_PRESSED.
`DrawBackground ()`	Draws the lines, rectangles, buttons, and texts on LCD to show the graphical user interface (GUI) by sending the commands using USART port.

4.2.2. Application layer

The application layer is designed using a finite state machine (FSM) structure as shown in Fig. 11.

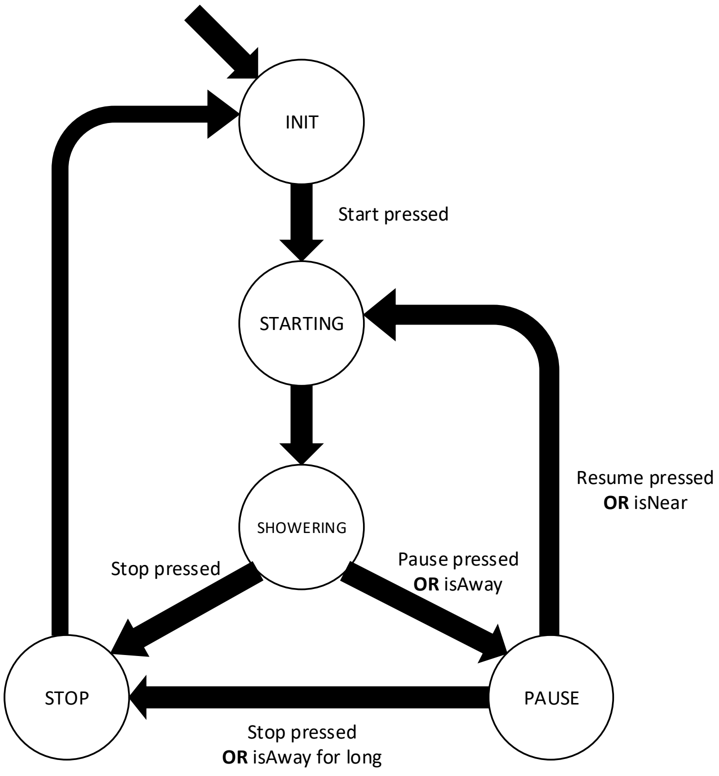


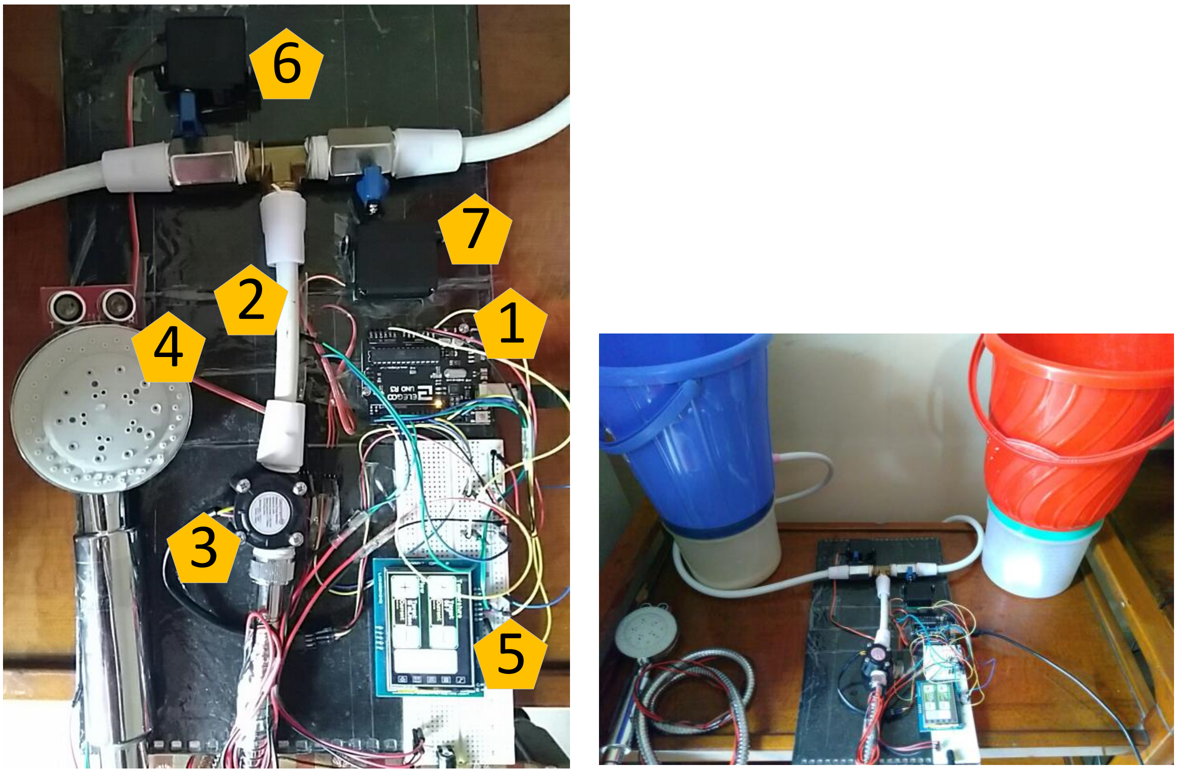
Figure 11. FSM of the application layer.

The system starts at INIT state. At this state, both the ball valves are closed. The user can set the target temperature and flow rate using the plus (+) and minus (-) buttons on the LCD. This state shows the ‘Start’ button on the LCD and pressing this button takes the system to the STARTING state. At STARTING state, both the ball valves are equally opened slightly to sense the current temperature and flow rate of the water. After 5 seconds, the system automatically goes to the SHOWERING state. At the SHOWERING state, the system reads the current temperature and flow rate, and continuously adjusts the hot and cold water ratio to get the target temperature and flow rate according to the discussion in Sec. 3.2. The current temperature and flow rate are displayed and the user can modify the target temperature and flow rate by pressing (+) and (-) buttons in real-time. The system also keeps track of total showering time using a *Timer*. At this state, the 'Pause' and 'Stop' buttons are displayed. If the user presses the 'Pause' button or the user is away directly from the shower for soaping/shampooing continuously for more than AWAY_WAIT_MS = 2 sec, the system goes to the PAUSE state. If the user presses the 'Stop' button, the system goes to the STOP state.

At PAUSE state, both the valves are turned off. The 'Resume' and 'Stop' buttons are displayed on LCD. If the user presses the 'Resume' button or the user stays near the shower continuously for more than NEAR_WAIT_MS = 2 sec, the system goes to the STARTING state. If the user presses the 'Stop' button or the user is away from the shower for a long time – 5 minutes for instance - the system goes to the STOP state. At STOP state, both the valves are turned off. The system shows the report of the shower session – the water usage, the showering time, and bar graphs showing the last 5 water usages and showering times. The system then automatically goes to INIT state after 30 seconds.

5. Result

A prototype of the proposed smart shower system is developed, calibrated and tested successfully. The photograph of the prototype and the experimental setup are shown in Fig. 12.



(a) (b)

Figure 12. (a) Photograph of the prototype: (1) microcontroller board, (2) temperature sensor inserted inside the pipe, (3) flowrate sensor, (4) shower head with distance sensor attached on top, (5) graphical LCD with touchscreen, (6) servo motor for rotating cold water ball valve handle, (7) servo motor for rotating hot water ball valve handle; (b) Photograph of the experimental setup with buckets containing cold water (left) and hot water (right).

5.1. Calibration of temperature and flowrate sensors

The temperature sensor is calibrated to get the temperature in Fahrenheit unit from the proportional ADC value of the circuit shown in Fig. 10. A plot is made with ADC value in the x-axis and the measured temperature using a digital thermometer in the y-axis for different temperatures of water. The linear equation derived from the plot is expressed in (6), where t_f is the temperature in Fahrenheit and n is the 10 bit ADC value.

$$t_f = 0.14n + 4 \qquad (6)$$

The flow rate sensor is calibrated to get the accurate flow rate and the total water usage report. Several experiments are conducted by passing a known amount of water through the sensor and logging the total number of pulses generated by the sensor. From these experiments, the pulse per liter is found to be 225. It is observed that after the wheel in the sensor stops – it takes a certain amount of flowrate to start the wheel spinning again due to inertia. In the same way, when the wheel is running at high speed – the wheel speed does not decrease immediately due to the inertia of the wheel - even if the flow rate is decreased. This phenomenon may cause a delay in calculating the accurate flow rate by the proposed system.

5.2. Prototype testing: Phase I

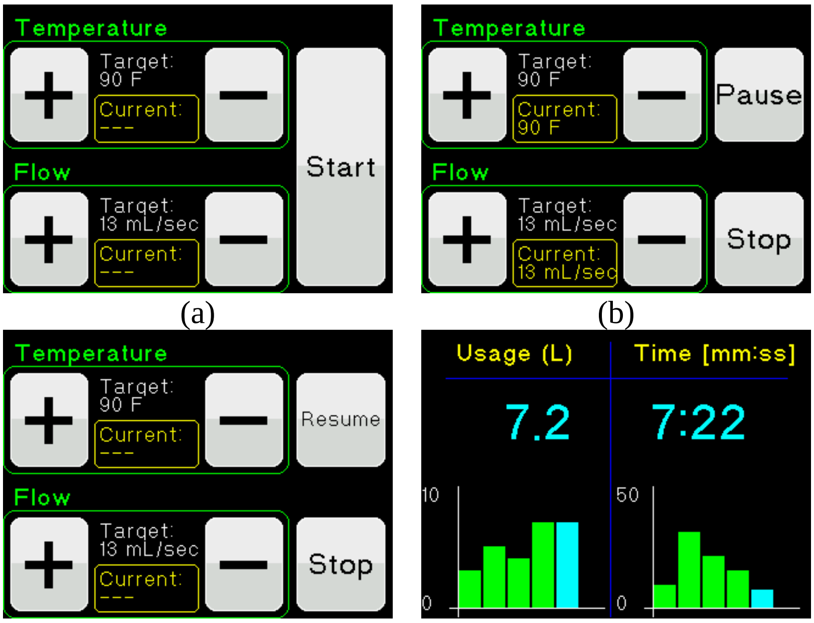
The proposed system was tested in two phases. In the first phase, the temperature sensor and the flow rate sensor were disconnected from the system’s microcontroller. Hot and cold water supply was also removed. A voltage divider circuit was made using a potentiometer and its middle pin was connected to the system microcontroller’s ADC port. Different voltages can be generated by rotating its knob mimicking different temperature. Using another microcontroller, square wave pulses were fed to the system's microcontroller to mimic different flow rates. The frequency of the pulses can be controlled externally by rotating the knob of a potentiometer. During the test, the behavior of the servo motors rotation is observed and verified for different temperatures ranging from 0 to 102˚F. and flowrates ranging from 0 to 266 mL/sec.

5.3. Prototype testing: Phase II

In the second phase of the test, the temperature sensor and the flow rate sensor were reconnected to the system’s microcontroller and an experimental setup as shown in Fig. 12 is used. Two buckets were used to hold the cold and the hot water. Several experiments are done to test the functionality of the proposed system for different amount (for creating different pressures) and temperatures of water in the buckets.

For instance, one experiment uses the cold water having a temperature of 77˚F and the hot water having ta emperature of 104˚F. The screenshots of the LCD at different states of the system is shown in Fig. 13. At the INIT state, the target temperature is set to 90˚F and flowrate is set to 13 mL/sec as shown in Fig. 13 (a). The ‘+’ and ‘-‘ button pairs are used to increase and decrease the target temperature and flowrates. After ‘Start’ button is pressed, the system goes to STARTING state and then goes to the SHOWERING state. In this state, the system adjusts the temperature and flowrate by rotating the servo motors to get the target values. A screenshot of the LCD at this state is shown in Fig. 13 (b). This state shows the current temperature and flowrate along with the target values. In Fig. 13 (b), we see that the system properly adjusted the ratio of the cold and hot water – making the target and the current values equal. The water coming out from the shower head is collected and its temperature is measured using a digital thermometer. The reading of the digital thermometer matched the current temperature shown in the LCD. Testing is also done by changing the target values in real-time during a shower, and the system is able to adjust the ratio to get the target values. The distance sensor is mounted on top of the shower head as shown in Fig. 12. During showering state, the proximity related functionalities are tested. The system successfully goes to PAUSE state when the user is away and resumes to STARTING state when the user is near.

When the system is at PAUSE state, the LCD shows the 'Resume' button as shown in Fig. 13(c). When showering is done and 'Stop' button is pressed - the system goes to STOP state and shows the water usage and showering duration report as shown in Fig. 13(d). All the water coming out from the shower head is collected in a bucket and then manually measured using a measuring cup. The manual measurement exactly matched with the reported value of 7.2 liters. The showering duration shown in the report is also verified using a stopwatch. The report also shows the history of the last five water usage and durations using bar graphs.



(a) (b) (c) (d)

Figure 13. Screenshot of the LCD at different states: (a) showing target temperature, target flowrate and 'Start' button at INIT state; (b) showing target and current temperature, target and current flowrate, 'Pause' button and 'Stop' button at SHOWERING state; (c) showing 'Resume' button at PAUSE state; (d) showing water usage and showering time with bar graphs at STOP state.

5.4. Comparison with other works

A comparison with other related works is shown in Table 2. It shows that the proposed shower has more advantages and features than the related works. The proposed smart shower takes the temperature and flow rate from the user as input and automatically adjusts both the temperature and the flow rate even if the supply water's temperature and pressure fluctuate. The proposed shower displays the current temperature and flow rate in real-time. Using distance sensor, the system automatically turns off the shower when the user is away for soaping or shampooing – to save water, and also automatically turns on the shower when the user is near. It also generates reports on water usage and shower duration using bar graphs of last five showers - to promote awareness on water savings.

6. CONCLUSION

In this paper, an embedded system based autonomous shower is proposed. Using sensors and actuators, the proposed shower outputs water with the user-defined target temperature and flow rate. The shower is paused and resumed automatically depending on the proximity of the user - to save water. It displays reports on water usage and showering durations at the end of each showering sessions to increase awareness. A prototype of the proposed smart shower is developed and tested successfully. Future work includes making showering profiles for users, Bluetooth connectivity with a smartphone, and developing the smartphone app.

Table 2. Comparison with other works

	Display temperature	Display flowrate	Auto adjust temperature	Auto adjust flowrate	Auto pause and resume on user proximity	Display total water usage	Display showering duration
Thermostatic valve [7]	No	No	Yes	Yes	No	No	No
Hydrao Smart Shower [15]	No	No	No	No	No	Yes, on Smartphone	No
DreamSpa [16]	Yes	No	No	No	No	No	No
WaterHawk [17]	Yes	Yes	No	No	No	No	No
Amphiro [18]	Yes	No	No	No	No	Yes	No
Yoo Mee [19]	Yes	No	No	No	No	No	No
EvaDrop [20]	Yes, on Smartphone	No	Yes, only at starting	No	Yes, changes flowrate on proximity	Yes, on Smartphone	Yes, on Smartphone
Moen [21]	Yes	No	Yes	No	No	No	Yes, a timer can be set.
Proposed	Yes	Yes	Yes	Yes	Yes	Yes	Yes

REFERENCES

[1] G. Gokilakrishnan, S. Divya, R. Rajesh, V. Selvakumar, "Operating torque in ball valves-A review," *International Journal For Technological Research In Engineering*, vol. 2, no. 4, pp. 311-315, 2014

[2] What causes an apartment building to have drastic fluctuations in shower water temperature?, [Online]. Available: https://www.quora.com/What-causes-an-apartment-building-to-have-drastic-fluctuations-in-shower-water-temperature, 2018.

[3] Why the shower gets hot or cold when you flush, [Online]. Available: https://lifehacker.com/why-the-shower-gets-hot-or-cold-when-you-flush-and-how-508252782, 2018.

[4] Design considerations for water supplies in apartment buildings and flats, [Online]. Available: http://www.anglianwater.co.uk/_assets/media/Design_Considerations_for_water_supplies_in_apartment_buildings_and_flats_LED.pdf, 2018.

[5] Showering to Savings, [Online]. Available: https://www.home-water-works.org/indoor-use/showers, 2018.

[6] W. J. Korotitsch and R. O. Nelson-Gray, “An overview of self-monitoring research in assessment and treatment,” *Psychological Assessment*, 11(4), pp. 415-425, 1999.

[7] Delta Faucet T27967 Ara Angular Modern Monitor 17 Series Valve Trim with 6-Setting Integrated Diverter, [Online]. Available: https://www.efaucets.com/detail.asp?product_id=T27967, 2018.

[8] Thermostatic vs Pressure Balance Shower Valves: What’s the Difference?, [Online]. Available: https://www.yliving.com/blog/thermostatic-vs-pressure-balance-shower-valves, 2018.

[9] E. Chobot, D. Newby, R. Chandler, N. A. Mulaweh, C. Chen, and C. P. Ráez, "Design and implementation of a wireless sensor and actuator network for energy measurement and control at home," *International Journal of Embedded Systems and Applications (IJESA)*, vol. 3, no. 1, pp. 1 – 15, 2013.

[10] Iswanto and H. Muhammad, "Weather monitoring station with remote radio frequency wireless communications," *International Journal of Embedded Systems and Applications (IJESA)*, vol. 2, no. 3, pp. 99 – 106, 2012.

[11] C. Kavitha, A. V. Ramana, and S. S. Raj, "Embedded management system for out patient department," *International Journal of Embedded Systems and Applications (IJESA)*, vol. 2, no. 3, pp. 47 – 56, 2012.

[12] M. Ma, B. Hotrabhavananda, J. Hall and M. Skubic, "Assistive Adjustable Smart Shower System," *IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, Philadelphia, PA, 2017, pp. 253-254.

[13] D. De Freitas Melo, E. De Souza Lage, A. V. Rocha and B. De Jesus Cardoso, "Improving the consumption and water heating efficiency in smart buildings," *13th International Conference and Expo on Emerging Technologies for a Smarter World (CEWIT)*, Stony Brook, NY, 2017, pp. 1-6.

[14] Y. Gao, D. Hou and S. Banerjee, "Fixture identification from aggregated hot water consumption data," *2016 IEEE International Smart Cities Conference (ISC2)*, Trento, 2016, pp. 1-6.

[15] Hydrao Smart Shower, [Online]. Available: https://www.hydrao.com/en/products/hydrao-shower-aloe, 2018.

[16] DreamSpa® AquaFan 12-inch All-chrome Rainfall Shower Head with Color-Changing LED/LCD Temperature Display, [Online]. Available: http://www.ipshowers.com/dreamspa-aquafan-12-inch-all-chrome-rainfall-shower-head-with-color-changing-led-lcd-temperature-display, 2018.

[17] WaterHawk 6" Smart Rain Shower Head with Water Usage and Temperature LED Display, [Online]. Available: https://www.newegg.com/Product/Product.aspx?Item=0N3-00CA-00001, 2018.

[18] Amphiro a1 basic, [Online]. Available: https://www.amphiro.com/en/produkt/amphiro-a1-basic, 2018.

[19] YOO.MEE LED Thermometer Handheld Shower Heads, [Online]. Available: https://www.amazon.com/YOO-MEE-Temperature-Designed-Lifetime-Accessories/dp/B015MM9RIG, 2018.

[20] EvaDrop, [Online]. Available: http://evadrop.com, 2018.

[21] Moen U Shower Smart Home Connected Bathroom Controller, [Online]. Available: https://www.amazon.com/Moen-Connected-Bathroom-Controller-TS3302TB/dp/B01MY07CZG, 2018.

[22] 1/2" NPT, Stainless Steel 304 Mini Ball Valve Female X Male, [Online]. Available: https://www.amazon.com/dp/B0734QDGZW, 2018.

[23] Intersection area of overlapping circles, [Online]. Available: http://jwilson.coe.uga.edu/EMAT6680Su12/Carreras/EMAT6690/Essay2/essay2.html, 2018.

[24] ATmega328P Microcontroller, [Online]. Available: http://www.microchip.com/wwwproducts/en/ATmega328p, 2018.

[25] Thermistor, Vishay BC components, [Online]. Available: http://www.vishay.com/docs/29048/ntcle203.pdf., 2018.

[26] DIGITEN G1/2" Water Flow Hall Sensor, [Online]. Available: https://www.amazon.com/DIGITEN-Sensor-Switch-Flowmeter-Counter/dp/B00VKATCRQ/

[27] Ultrasonic Sensor, [Online]. Available: https://www.sparkfun.com/products/13959, 2018.

[28] SmartGPU 2 - LCD320×240 - 2.4", [Online]. Available: https://www.vizictechnologies.com/smartgpu-2, 2018.

[29] Servo Motor, [Online]. Available: https://www.sparkfun.com/products/11965, 2018.

Author: Dr. Tareq Khan is an Assistant Professor in the School of Engineering Technology of Eastern Michigan University, USA. To date, Dr. Khan has authored (and co-authored) 2 books, 1 book chapter, 17 peer-reviewed journals and 26 international conference papers. He currently has 2 US patents granted. His research interests include smart home, embedded systems targeting healthcare applications, Internet of Things (IoT), machine learning, capsule endoscopy, and image processing. He is a member of the Institute of Electrical and Electronics Engineers (IEEE).