Smart Liquid Mixing System

Sabitha Gauni* and Vuddagiri Krishna Vamsi
Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India
Email: sabithag@srmist.edu.in

Abstract. Smart drink appliances with a wide area of different control specifications have become a popular research area. Most of these systems are expensive and takes a lot of time to replace the whole system. The proposed method below will help us upgrade the existing appliances that make liquid mixing possible and a feature where we can control it with the help of voice. This system will help the present technologies upgrade them for automation, which will be easy since this is done using simple methods while having an ease of access. The system runs on an android smart phone, connected with the help of Bluetooth to raspberry pi 3 nodes. The application analyses the commands given through voice and sends the data to the Raspberry Pi. Based on the motors connected to it, the application searches for keywords in the command and takes the required control action.

Keywords: Automation, Smart home appliances, industries, liquid mixing

1. Introduction
Today, many sophisticated technologies make household jobs much simpler and save a lot of time. Generally, home automation systems control appliances like fans, lights, television, and others. They are also used to open up the doors and windows [2].

In this pace of increase in the number of household appliances, humans expect to work in automation to make their jobs easier and decrease the amount of time they spend doing things or household chores. One such thing is making their custom-made juices or liquids at their homes, such as health drinks.

Also, in industries such as the medical industry, paint industry, chemical industry, pharmaceutical industry, when two or more kinds of miscible liquids or fluids are required to be mixed uniformly, they generally use filling stations to mix different kinds of such liquids. But the present conventional method, when shifted towards automation, can cause a huge difference in efficiency since the old methods are not highly efficient for automation purposes for address constructing an automation system, especially to mix liquids according to the user-based inputs with the help of gestures, touch, short message service (SMS) [4] [6]. In this study, the system is operated using voice as the command input.

In the present day, many people with a smartphone and who all are adapted to the smart home environment solely depend upon the voice assistants such as Google Assistant [10,12], Amazon Alexa [13], Microsoft Cortana [14], and many more, as said by Bohouta and Veton in [11]. Since these smart Assistants and many others are going to be a part of the human daily-life ecosystem, this model will be implemented to be compatible with different voice assistants and controlled by voice. That is why we
need to integrate the model with the voice assistants to make it future-proof and put this and the movement towards the Internet of Things.

The voice commands are translated using an application in the Android smartphone to control the motors. The command is then sent using the Bluetooth feature of the smartphone. Thus, the received command will be decoded by Raspberry Pi. Upon validation of the received data, the Raspberry Pi sends the motor driver's control signals to run the motors [3]-[5].

A resend request is sent back to the transmitter to resend the data in case of errors in the received data.

To demonstrate the project's work, 5 kinds of different liquid mixtures are shown using raw materials like Soda water, any 2 juices, milk, and other different raw materials to make home-made health juices.

2. Related Work

In [2], the implementation used to operate the doors and windows is mainly based on the controller in the application in a smart phone connected to the android device using a Bluetooth module. We can use RF connection and GSM for voice home automation. But in [2], Bluetooth is decided for executing this home automation since it is more frequently used and available in all the latest smart phones.

In [1], a liquid mixing system is proposed using PLC to automatically fill bottles in industries and other areas where liquid mixing is needed. But when we need the same model in a miniature version such that it can be used in a small-scale application, this method is difficult to implement, and the cost is higher.

In [8][9], A voice-controlled robot was made with the help of a Raspberry Pi and a smart phone connected to the Raspberry Pi with the help of a Bluetooth module. An application was created such that when a voice command was given through that application in the smart phone, the voice command will be converted to text, which can be used to control the actions that can be customizable in the Raspberry Pi and control the robot.

In [15][16], the authors published their analysis of how the smart voice assistants work in day-to-day life, their work in the backend, and how they analyze our voice to be the best version of voice recognition. They also explained how smart assistants, or voice assistants, can have a huge impact on how we control things and be able to do daily chores, and make our lives easier in many ways.

3. Problem Definition

We will be controlling the whole system using a smart phone with the help of an application where the smart phone will be connected with the help of Bluetooth. We have chosen this method since the smart phone is the most popular gadget at the present technology pace.

Generally, we can implement a system where the system can be connected to Bluetooth as a media connection to control the whole system. But during the speech recognition application design, though smart phones have good processors, they consume a lot of battery while doing tasks that involve high processing power. So that choice is not considered.

Sometimes in liquid mixing, we need to mix two kinds of liquids made of different viscosities. But this makes the design of the system difficult to design since the motors used will be the same for all the liquids in this system for ease of making. This causes the motor's time to dispense the liquid at different times for liquids of different viscosities. For example, if we are using this system as a smart home appliance to make juices at home, the milk will be having a viscosity value, while the juice with which the milk will be mixed will have a different viscosity. So, same time attributes cannot be assigned to the motors that are pumping these liquids.

We need to work on the timings of different liquids that dispense at different times so that a single motor can be used while having different time constraints based on the type of liquid we will be using.

So, that has to be done to experiment with the time to dispense the liquid with the help of the motor for a constant amount of 30 milliliters.
Since we are thinking of implementing the same system to make it work with the smart home assistants, we are trying to prepare a model that can connect to Wi-Fi or have its GSM module with data connectivity. But this can lead to problems when there might be no data transfer due to lack of signal or power outage. So, the GSM module is not a suitable component to implement this. Since the whole model will be a part of the home automation system, Wi-Fi will be a better option.

If the pipe's output is at a level below the input of the pipe, which is the motor, then the water starts to flow freely without the effect of the motor even if it works. This is called as "Siphoning effect." These mechanical elements need to be considered to avoid some unwanted output from the model.

4. **Formulation of Hypothesis**

The main method to explain the working of the system depends on the motors used and the amount of time each motor works. All the motors are rated with liters per hour rating to let us know the amount of liquid they can pump within an hour. But this applies only to water. If we want to use any other liquid such as milk, the viscosity may change, causing it to change the value of the rate at which the motor produces the output. So, to avoid this, the value of time allotted to each motor for a certain amount of quantity is varied for different kinds of liquids. So, we need to analyze each liquid's value so that we can calculate the amount of liquid the motor dispenses for a minute. If the liquid dispensed for 1 minute is 1200 milliliters, the time the motor needed to run to give 300 milliliters is calculated by taking the volume-time proportions. That gives us 15 seconds.

This value will be changed for different kinds of liquids. So, the liquid used in the system should be predetermined such that when the command or the voice input is given, then the defined amount of liquid will be dispensed to the container.

5. **Proposed System**

This proposed design will give the voice command through a smart phone connected to Google API to recognize the voice easily. If the connection between the smart phone and the Raspberry Pi is done with the help of Bluetooth, then we need an application in the mobile which can send our voice data that was converted from text-to-speech to the Raspberry Pi, which will trigger based on that.

If we are going to use Wi-Fi technology, then the whole setup needs to be connected to the internet where the Raspberry Pi can access the web request sent through the internet for the respective trigger.

The figure in the figure shows a brief view of the overall process for the project to work. Figure 1 and Figure 2 shows the model’s proposed models to be designed in full scale.

A **Components Used**

To implement this system, many hardware components are used:

- Android smart phone having the latest Android 5.0 or above as the operating system.
- Wi-Fi connection or GSM Data Connection
- HC-05 Bluetooth module to send data from the smart phone to Raspberry Pi 3. (If the model is decided to work solely based on Bluetooth)
- Raspberry Pi 3 is the main hardware component. It is a microcontroller board based on the Cortex A53 (ARMv8) 64-bit SoC [7].
- Motors to pump the liquid and will be controlled with the help of motor drivers connected to Raspberry Pi.
- Motor Driver
- Relays

Based on the information above, the proposed automation system is shown in the figure 1:
Figure 1. The controlling side of the proposed automation system

Figure 2. The receiving side of the proposed automation system

This is how the proposed system is implemented mainly with Raspberry pi as the processing center. Figure 3 shows the overall layout and the block diagram model of the proposed system.

Figure 3. The Block Diagram of the proposed automation system

6. Result & Discussions

The proposed system will give different inputs to make different drinks in various fields of liquid mixing. Here we will be showing how to make different color combinations using base colors so that as we mix them in different proportions, we can achieve any number of colors we require to create the colors for paints. These base colors are liquids with a fixed viscosity all the time.

Before we start the system, we need to analyze the time in seconds in which the liquid dispenses 30 milliliters. This will be taken as a standard. Let us consider for this system, the viscosity of all the paints are in such a way that the motor dispenses 30 milliliters of paint for 2 seconds. If we need 60 milliliters, then the motor works for 4 seconds, or it can vary the motor speed based on the circuit used. The “Siphoning Effect” should be considered as this can affect the performance of the model.
Table 1: The time each motor works for different combinations.

| Name of the color | Amount of time the motor works (in seconds) |
|-------------------|--------------------------------------------|
|                   | Red | Yellow | Blue | Black | White |
| Orange            | 2   | 2      | 0    | 0     | 0     |
| Dark Orange       | 4   | 2      | 0    | 0     | 0     |
| Purple            | 2   | 0      | 2    | 0     | 0     |
| Green             | 0   | 2      | 2    | 0     | 0     |
| Dark Red          | 4   | 0      | 0    | 2     | 0     |
| Grey              | 0   | 0      | 0    | 2     | 2     |
| Brown             | 0   | 4      | 0    | 2     | 0     |
| Pink              | 2   | 0      | 0    | 0     | 2     |

The above, Table 1, shows various combinations to obtain different shades of colors that will be mixed in the last such that the liquids are mixed properly.

The processing occurs in the Raspberry Pi, where the code required to run this program is stored in the Raspberry Pi’s on-board memory. The coding is done with the help of Python IDE in which we can make changes in the code, or we can put the code for the Raspberry pi to function according to the command given through the smart phone.

Suppose ever there is any change needed in the quantity of liquid needed in the system. In that case, the code that was one in Raspberry Pi with the help of Python IDE needs to be changed in such a manner so that the control signal will be generated accordingly based on the input given as the command.

The flowchart from figure 4 shows the working representation of the Liquid mixing system, which we designed for the transmission side of the system, i.e., mobile in this proposed model. The receiving part of the system is at the Raspberry pi, where the whole system is connected to the internet with a Wi-Fi connection. As soon as the connection is established and necessary settings are done, Raspberry Pi can now be made ready to execute the code. For the present system, Raspberry Pi’s data transmission can be done through the Wi-Fi built in the Raspberry Pi. The receiving end of the system can be explained with the help of a flowchart from figure 5.

The below flowcharts show us how we can implement the model using the Google API and integrate it with Google Home Ecosystem. This is very useful in this liquid mixing model and any other smart home device connected to the internet. The flow charts describing the flow of how the system works shown in figure 4. With the flow charts’ help, we can show how the implementation takes place in the proposed system. Figure 4 shows how the implementation takes place in the smartphone to be transmitted to the internet.

In Figure 5, with Internet Connection, the Raspberry Pi gets the user input data and processes the data sent from the cloud server. According to the data, the viscosities of the liquids assigned to different liquids are taken. Based on the data, the control signal is generated on sent to the motor drivers to dispense the liquid.
Figure 4. Flow chart of the controlling side of the proposed automation system

First, we will be creating our voice implementable Applets using the website "If This Then That" (https://ifttt.com/) [18]. We can create an application called applet that can be directly interacted with or connected to the account attached to the application on this website. It has various applications like Gmail, Google Assistant, Amazon Alexa, and many others.

For this project, we will be using the applet to attach the trigger URL to the voice command given to the Google Assistant as it was the most used voice assistant in smart phones.

In this applet, we will be adding a public URL that is created to send the data to the Raspberry Pi that can trigger the assigned GPIO pins on the board based on the URL given. This public URL can be created inside the Raspberry Pi using the "ngrok" application.

"ngrok" is an application installed in the Raspberry Pi that can be used to create Public URLs for different purposes [3]. In this proposed system, we will be accessing the ngrok to access the local host server created in the Raspberry Pi to make the code accessible. In the execution code, we will be accessing the code with a PIP called "Flask," which will connect our code to port 80. We are using Flask since it is used to create a lightweight web server and can execute some complex applications. Flask is also easy to use and understand.

As the Public URL linked with the code is initialized at the IFTTT website, the code is then executed with the Terminal help. The voice command that needs to be given to a certain trigger is programmed in the IFTTT website, where we can assign the voice command to the specific URL. The attachment that needs to be added at the end of the public URL is coded in the execution file coded to control and access the GPIO pins through which the motors are connected.
Figure 5. Flow chart of the receiving side of the proposed automation system

Table 1 shows the outputs that we can present if we want to mix colors and by controlling the timings of each motor. In the table 1, each motor's timings are defined to obtain a glass of liquid of certain command given. The working can be explained as follows.

In the beginning, we will be initializing the "ngrok" server in the Raspberry Pi, which will be giving us a forwarding address that will connect to our server that was running in the Raspberry Pi can be observed in Figure 6.

![Figure6. Initializing NGROK in Raspberry Pi.](image)
As shown in the figure, the address in the forwarding column in the "ngrok" column is used to communicate with the Raspberry Pi and the code running inside it. This address is then copied and used at the IFTTT website, where we will be using it to assign the input method to access the link. Here, we are using Google assistant as an input.

At the IFTTT website [18], we will be defining what the input is given on the phone. On this website, we will assign the voice command that will be given to activate the specific Public URL with that command. The input given through the IFTTT website can be seen in Figure 7.

![Figure 7. Screenshot of IFTTT website for voice command](image)

This voice command will now be attached to the Public URL using an application called "WebHooks" [17,19]. This application is useful to link any public URL to the input application. Since the input application is Google Assistant, the Google Assistant's commands are linked with the Public URL given in the WebHooks application. The image of how WebHooks application looks is shown in Figure 8.
In the real-life implementation of this system, we can change the model according to the specifications needed. Changes can be done, such as changing the motors that can pump water for more capacity for longer times, and the voltages can be changed according to the system designed. Since this is designed around a Raspberry Pi, we can change the code to use the system according to the ingredients or recipe or the way we need by changing the times that each motor will be used to pump the liquid and also increasing the number of liquids to use by increasing the count of motors.

Since the output even depends on the viscosity and the tubes or pipes used for the system, these should be tried and experimented on several times. Then the timing and material should be decided for the model to be designed. The picture shown in Figure 9 can be seen to have an idea of how the project was implemented during the project work.

But for the proposed model, we will be using a 5 Volt motor to pump the liquid, which has a rating of pumping 120 liters per hour. These are connected to a DC 5 Volt rating relay acting as a switch based on the Raspberry Pi trigger. The relay supply is modulated with a bridge rectifier’s help, which can convert the 230 Volt DC from the Transformer to either 12 volts or 5 volts. The Transformer is a step-down transformer that will convert 230 volts AC to 230 volts DC.
The whole circuit for the proposed model is encased in a wooden structure built to house all the bottles containing the liquids that need to be pumped. The circuit explained above, a funnel to mix the liquids and glass to dispense the mixed liquid. The whole wood structure is based on a footstool where a layer of plank is added in the stool's midway parallel to the floor.

In the proposed system, each motor can pump liquid of 180 milliliters for 10 seconds. So, if we need a quantity of 1800 milliliters, we will be turning the motor on for 100 seconds and then turning the motor off.

Coming to the model's software, we will be using the NGROK, Flask, "If This Then That" website, and Python. While the basic coding was done using Python, we will be using Flask web Framework to make the code connect to the internet using the server with a URL.

Flask is a web framework, i.e., a third-party library used to develop various web applications. Flask is used in this code to connect the functions that we implement to run the motors to be implemented using the request from the server.

NGROK is a web development tool used to create a public URL to host the web application. In the NGROK application, the Public URL will be created, which can be used to access the data from the Raspberry Pi through request mapping. This request mapping allows us to access the Python code to run the different functions designed to run the respective motors for a particular time according to the code given. This URL allows us to access the code in the Raspberry Pi as long as it's connected to the internet.

"If This Then That" website is a website that consists of many applets that we can use to customize our experience with many of our day-to-day smart applications such as Google Assistant, Smart home devices, and many others.
For this project, we will be using the Google Assistant applet, which connects our Google account such that we can input our custom voice command and then also define an action when that voice command is triggered. The URL created with the NGROK and the Request mapping for the Flask Framework function is then linked to a particular voice command in the Google assistant applet. This connects the particular voice command to a particular method in the Python code. So, as the particular voice command is given through the Google Assistant from the connected account to the "If This Then That" website, then the particular request mapped function executes through the server's request. It then starts the submersible motors according to the time given in the functions. The Raspberry Pi is run on Raspbian OS to have the flexibility in using the Raspberry Pi.

Drawbacks

Often, the system or the model may not work properly while using the voice commands because there might be connectivity issues that can cause the system from receiving the command properly. Since the system works solely based on the ping or the request Raspberry Pi request from over the internet, if there's a delay or any communication, the output may not be properly produced by the model.

In the same manner, if ever there is a communication gap such as when the voice command is not said, or the voice command was said in a different accent where the Google Assistant, which we are using in this model or any other voice assistants could not understand or recognize, this can cause to the inefficiency of implementation of the system, or this can lead to activating a different voice command which can cause to an undesired output from the model system.

Such sort of undesired output can cause a loss to the ingredients used in the model and a wastage caused to the resources used. We should also consider the cost that can affect the model and the budget in which it is implemented. This flaw can lead to a lot of unknown loss if ever this was implemented.

To avoid this mistake, we need to think of a method to authenticate the voice command. We need to create a method to check if the voice assistant's command is the one we need. We can create a user interface such that the assistant can confirm the command it receives from the person. When confirmed, the system only implements the command and starts the motors according to the command given.

Sometimes, when we stop using the model for longer periods, the liquids that are present inside the pipes connected to the motors get dried up in the pipe, which may lead to choking of the pipes and cause the output to be different than what was expected to disrupt the recipe and lead to a loss of investment towards the ingredients. This problem can also lead to the pipe's damage that transfers the liquids into the funnel or the mixing setup.

One of the other problems related to the pump motors used in the system's model is reliability. They sometimes work to give a varied amount of quantity due to various factors. One of them is the viscosity of the liquid, air bubbles that may enter the pipe while the motor was pumping the liquid to dispense into the funnel or the mixing system, choking off the pipes due to some impurities in the liquids that are going to be dispensed which can lead to clogging of the pipes. One more factor to consider while considering drawbacks from the submersible motors is the reliability or the lifetime of the motors. If it ever stops working, the submersible motors are tough to replace, and it is not advisable to replace them very often.

Overcoming Drawbacks

The drawbacks mentioned above can cause a lot of loss to money and other entities such as time and many other aspects of life can lead to a lot of inefficiency and improper use of resources. These drawbacks can be overcome by many different methods, such as the following.

Many methods can resolve the problem regarding the connectivity issues, such as receiving the wrong command. One of them is specifying the related voice command to the Google assistant in the "If This Then That" website such that the voice command given can be distinct such as like adding a
name to the command to not to confuse among other commands that Google assistant or any other voice assistants generally mistake with.

The communication gap or the delay for the system can be decreased using many ways, such as implementing an efficient way to get the voice commands using a system that does not need to connect to an Internet while just using a server in the home network can host this server in the Raspberry Pi such that it can decrease the latency, or the time required for the model to implement the command given to it.

One of the other methods to check the user's command is to create a User interface that can confirm the user's command and validate it. This can help stop the clogging of the pipes and clean the pipes, which can increase the pipes' lifetime.

The submersible motors or whatever motors that are going to be used in the model should be decided such that they are more reliable and can last for a longer number of usage times.

7. Future Enhancements
The casing of the system is to be made preferable for various uses.
The number of motors and liquids is to be increased for more options.
Different sensors can be used for level pressure sensing.
Blenders can be added for thorough mixing of liquids.
Voice automation can be done in the system itself for convenience.
This system can be made a part of a Home Automation system.
System triggers can be given through other voice assistants like Amazon Alexa.

8. Conclusion
The proposed system is upgradable to other automation devices since implementing the present system is done using Raspberry Pi, a capable processor to run the programs. With the help of perfect equipment, we can connect other devices to operate through the same system. Or, we can attach this system to an already existing automation system setup that uses the Raspberry Pi.
The voice commands' reaction time depends on the quality of service, connectivity, and distance since Wi-Fi or Bluetooth is used here (If the Bluetooth model is used). But it can work perfectly with a home automation system making it one of the smart home appliances for the future [5]. This system's uses are in many fields like pharmaceutical, paint, home uses, or liquid manufacturing factories.
The system is easy to compile, cost-effective, environmentally friendly, uses low power, and has high efficiency.

References
[1] Mini Sreejeth, Shilpa Chouhan, “PLC based Automated Liquid Mixing and Bottle Filling System,” 1st IEEE International Conference on Power Electronics. Intelligent Control and Energy Systems, 2016.
[2] Reynaldo Lino Haposan Pakpahan, Dodi Wisaksono Sudiharto, Aji Gautama Putrada Satwiko, “The Prototype of Automated Doors and Windows by Using Voice Commands,” International Seminar on Application for Technology of Information and Communication, 2016.
[3] Annisa Sarah, Theresia Ghozali, Geraldo Giano, Melisa Mulyadi, Sandra Octaviani, Alfin Hikmaturokhman, “Learning IoT: Basic Experiments of Home Automation using ESP8266, Arduino and XBee,” IEEE International Conference on Smart Internet of Things (SmartIoT), 2020.
[4] Dhiraj Sunehra, Vemula Tejaswi, "Implementation of speech based Home Automation System using Bluetooth and GSM," International Conference on Signal Processing, Communication, Power and Embedded System, 2016.
[5] Markus Schinle, Johannes Schneider, Timon Blöcher, Dr. rer. Nat. Jochen Zimmermann, Dr.-Ing. Sebastian Chiriac, Prof. Dr. rer. Nat. Wilhelm Stork, "A modular approach for smart home..."
system architectures based on Android applications," 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, 2017.

[6] VLK. Bharadwaj Manda, Voona Kushal, N. Ramasubramanian, "An elegant home automation system using GSM and ARM-based architecture," IEEE Potentials, September/October 2018.

[7] Renuka P. Kondekar and Prof. A. O. Mulani, "Raspberry Pi based voice-operated Robot," International Research Journal of Engineering and Technology (IRJET) Volume: 02 Issue: 12 | December 2017.

[8] Piyush Vashta, Joginder Pal Singh, Pranav Jain, and Jitendra Kumar, "Raspberry Pi based voice-operated personal assistant (Neobot)," Proceedings of the Third International Conference on Electronics Communication and Aerospace Technology IEEE Conference, 2019

[9] About Raspberry-Pi 3 model B-plus https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus

[10] Gamal Bohouta and Veton Këpuska, "Next-Generation of Virtual Personal Assistants (Microsoft Cortana, Apple Siri, Amazon Alexa, and Google Home)," 2018.

[11] P. Mane, S. Simone, N. Gaikwad, and J. Ramteke, "Smart personal assistant using machine learning," International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, pp. 368-371, 2017.

[12] P. Dempsey, "The teardown: Google Home personal assistant," Engineering & Technology, vol. 12, no. 3, pp. 80-81, 2017.

[13] H. Chung et al., "Alexa Can I Trust You?", Computer, vol. 50, no. 9, pp. 100-104, 2017.

[14] G. López, L. Quesada, L. Guerrero, "A. Alexa vs. Siri vs. Cortana vs. Google Assistant: A Comparison of Speech-Based Natural User Interfaces," International Conference on Applied Human Factors and Ergonomics, pp. 241-250, 2017.

[15] A.A. Arriany, M.S. Musbah, "Applying voice recognition technology for smart home networks," Engineering & MIS (ICE) International Conference on, pp. 1-6, 2016.

[16] Bharath Ravi Prakash, Megha S, Niharika SR and Megha M G, "Super Smart Home Automation using Internet of Things," 4th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT), 2019.

[17] Mayur Wani, Prashant Ahire, "Real-Time Smart Mirror System Using Internet on Things" International Conference on Computing Communication Control and Automation (ICCUBEA), 2019.

[18] Google API from “If This Then That” https://www.ifttt.com

[19] Flask - lightweight web application Framework https://github.com/pallets/flask