

OCULAR DRUG DELIVERY SYSTEM USING OPEN-SOURCE SYRINGE PUMP

SURYA V, SRUTARTHA BOSE, SHREYA KULKARNI, KATHIRVELU D*

Department of Biomedical Engineering, Sri Ramaswamy Memorial Institute of Science and Technology, Chennai, Tamil Nadu – 603 203, India. Email: kathir297@gmail.com

Received: 06 December 2017, Revised and Accepted: 22 February 2018

ABSTRACT

Objective: The objective of this study is to design an automated ocular drug delivery system controlled by an open-source syringe pump, which in turn is to be controlled using Arduino UNO.

Methods: The Arduino UNO is the microcontroller used to control the entire setup. The drug delivery is executed using the open-source syringe pump. A 12V NEMA 17 stepper motor is used to execute the pump movements. To facilitate the ease of use, an Android application developed using Android Studio has been developed, where inputs are given in the form of the required drug dose. The application plays the role of converting the requested drug dose into the number of steps required to run the stepper motor. The stepper motor works based on simple lead-screw mechanism.

Results: The designed syringe pump has been tested using water. It was determined that for 42 steps a drug volume of 30 μl is dispensed by the designed device. Typically, a drop volume ranges from 33.8 μl to 63.4 μl and the device that has been discussed in this article can deliver a minimum volume of 30 μl .

Conclusion: Using this device, excessive use of drugs can be minimized, thus reducing the capital investment put behind ocular drugs.

Keywords: Arduino UNO, Drug delivery, Glaucoma, Leadscrew mechanism, Syringe pump.

© 2018 The Authors. Published by Innovare Academic Sciences Pvt Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>) DOI: <http://dx.doi.org/10.22159/ajpcr.2018.v11i6.24151>

INTRODUCTION

The most common ocular drug administration in today's world is in the form of eye drops. Patients suffering from various ophthalmic diseases such as red eyes, dry eyes, cataract, and glaucoma are prescribed with a certain amount of medication in the form of eye drops, either to cure or to improve their conditions. Diseases such as cataract and glaucoma, however, are pretty severe and are major contributors to the worldwide population of blind people. The typical dosage pattern for patients suffering from such diseases ranges from administering the specific drug at least 2–3 times/day on a daily basis [1]. The most common drugs used in such cases are Travatan, Xalatan, Beta blockers, and Lanosterol [2].

There are several side effects caused due to such drugs, for example, burning, stinging, ocular irritation, decreased eye sensation, and ptosis (drooping of the eyelid). The chances of these side effects being prevalent increase with the amount of exposure to these ocular drugs. It is common to observe such diseases in aged population. In which case, accurate drug administration becomes a topic of concern, since such patients encounter physical difficulties, which lead to inaccurate drug dose delivery to the eye [3]. Improper delivery of drug can lead to treatment failure, especially in case of glaucoma patients [4]. If it was for a more temporary condition such as random eye irritation, it would not have been a problem. However, for patients suffering from cataract or glaucoma, the error in the drug dose delivered builds up over time and contributes to the prevalence of major side effects. Apart from the side effects, the expenditure is also a factor of concern. For typical 2–3 drops per eye daily regimen, the approximate cost per year is \$427.69 for Travatan alone [5]. The lowest cost per day therapy was determined to be \$0.55 and the costliest being \$1.35 [2]. In most cases, there is a lot of spillages when too many drops are administered at one go. This leads to wastage of the drug, which further leads to increased expenditure.

The present article introduces an ocular drug delivery system (ODDS) that will promote precise drug delivery to eradicate the discussed

drawbacks of the typical eye drops. The principle of the system is that of an open-source syringe pump [6]. The syringe pump is actuated based on leadscrew mechanism so that precise movements can be executed [7]. Typically, closed-box syringe pumps are more complex and expensive [2(6)]. Hence, a simple cost-effective open-source syringe pump has been designed. The pump is run by a 12V NEMA-17 stepper motor, which controls the movement of the leadscrew in the form of steps entered. Previously, 8051 microcontroller has been used to run the stepper motor [8]. However, that is too complex so Raspberry Pi was used to make it easier. The free open-source software that has been used to run the stepper motor in this design is Arduino UNO as it is a very versatile, cost-efficient, and user-friendly microcontroller [7]. Arduino ATMEGA328 microcontroller has also been used to control a syringe pump [9]. However, Arduino UNO is a much user-friendly version. The volume of drug dispensed can be controlled by the number of steps required to run the stepper motor. Typically, one drop is equivalent to 25–70 μl . With large volumes, there is a risk of adverse side effects due to absorption of the drug through nasal mucosa [10]. Different drop bottles dispense drop volumes ranging from 33.8 μl to 63.4 μl [11,12]. Microneedles have been used for targeted drug delivery to the eye [10]. However, nozzles, on the other hand, are totally non-invasive. Hence, to avoid larger drops, a nozzle of 3×10^{-4} m diameter has been used to carry out precise dispensing of the drops [13].

METHODS

Hardware components

Arduino UNO is a microcontroller board that has 14 digital input/output pins, 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header, and a reset button. The main application of this board is to control the stepper motor through the stepper motor driver as well as transmit and receive input/output data between the user interface and the Arduino integrated development environment (IDE) software. The board runs on a 5V battery and its functions can be controlled by means of a C++ code in the Arduino IDE software. HC-06

is a bluetooth module which is compatible with the Arduino UNO board. The purpose of this module is to connect the cell phone user interface with the Arduino UNO board so that data can be transmitted with ease. It has 34 pins and has a wireless working frequency of 2.4 GHz. It can achieve up to 80 m of super long-distance communication under open environment. It requires an input DC power supply of 2.0–3.6 V.

A NEMA 17 bipolar stepper motor has been used to execute the syringe pump movements. This stepper motor runs on a 9 V battery, it has 200 steps/rev, and the phases are split into 1.8° each. The stepper motor rotational movement is further converted into translational movement by means of a lead screw mechanism. To connect the stepper motor with the leadscrew, respective coupling has been done.

L29 three-dimensional (3D) chip is required to drive the stepper motor based on the commands given by the Arduino UNO board. It comprises of a dual H-bridge circuit which enables the stepper motor to rotate in clockwise or anticlockwise direction. Referring to the pin diagram given below, the Pins 3, 6, 11, and 14 are connected to the stepper motor wires, respectively, and the Pins 2, 7, 10, and 15 are connected to the Arduino UNO board.

To carry out the pump movements, a strong support structure was needed. Hence, a setup has been constructed using aluminum metal, as shown in Fig. 1, in such a way that it connects the stepper motor with the 5 ml syringe by means of leadscrew, where the syringe is stationary and the plunger is moveable. To provide damping for the stepper motor vibration, this entire setup was built on an aluminum base. The setup comprises of the motor plate, front piece, lead piece with bearings, and the end piece. The stepper motor is attached to the setup through the motor plate, and the lead screw runs through the setup from the front piece till the end piece, at the center. The syringe is held in position by the lead piece and the end piece. The setup has been built in such a way that the position of the syringe can be adjusted by means of a knob, and hence, it is also replaceable.

Software components

The open-source Arduino software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and code used to run the board can be written in C++ programming language. The code is typed on the pad called a sketch. It is then verified and uploaded on to the respective Arduino board. The sketch comprises of two main parts called as the setup and the loop. The setup function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch. After setup has been called, function loop is executed repeatedly in the main program. It controls the board until the board is powered off or is reset. The advantage of this software is that it is compatible with all Arduino boards and hence very versatile and user-friendly.

An android application has been built to solve the purpose of a user interface. Android studio was used to build the android application. This software has a gradle-based build support and Android-specific refactoring and quick fixes can be easily executed. It has a rich layout editor which allows us to drag-and-drop UI components and preview layouts on multiple screen configurations. It also provides a deep code analysis that gives a detailed explanation about an exception based on the annotation added and also inspects code in the whole project. There are several reasons why we preferred this software over others. It helps in easy and quick navigation so that if we want to open any recently opened files, we can directly jump to it. Furthermore, it has an autosave feature that allows you to save files by default. The previews of the XML files can be viewed adjacent to each other so that changes can be made simultaneously. Other features include smart autocompletion, color coding, typo checks, and integrated command lines in the IDE.

Methodology

The Arduino UNO microcontroller, as shown in Fig. 2, is the heart of the designed ODDS. It executes the commands given through the C++

code entered into the Arduino IDE software, which is available on the personal computer. The Arduino board sends the respective commands to the L293D stepper motor driver, which further executes those commands to control the NEMA 17 stepper motor. As the stepper motor rotates, the leadscrew converts it into translational motion, thus moving the syringe plunger in the respective direction.

An android application has been developed to solve the purpose of the user interface. To enable transmission of data between the application and the Arduino IDE software, HC06 bluetooth module has been used. The connections for the bluetooth module are given separately on the breadboard. The android application has been developed in such a way that it can make the syringe plunger move in both directions, i.e., forward and backward. A 5 ml syringe has been used to contain the drug while it is being delivered to the eye. As it has already been mentioned, the connections are given on a breadboard, as shown in Fig. 3. The Pins 9, 10, 11, and 12 on the Arduino board are used as the input pins which are connected to the Pins 15, 10, 7, and 2 of the L293D IC, respectively.

Every stepper motor comprises of four lead wires that come with a certain color code. The NEMA 17 stepper motor that has been used for this system has the color code 1 which has red-, blue-, green-, and black-colored wires. According to the color code, the green and black wires

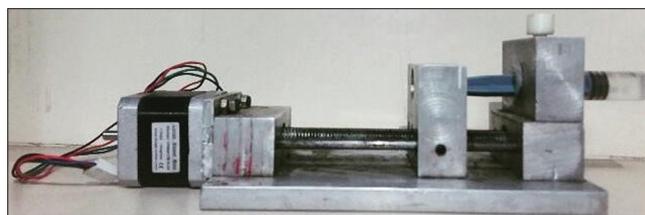


Fig. 1: Syringe pump setup

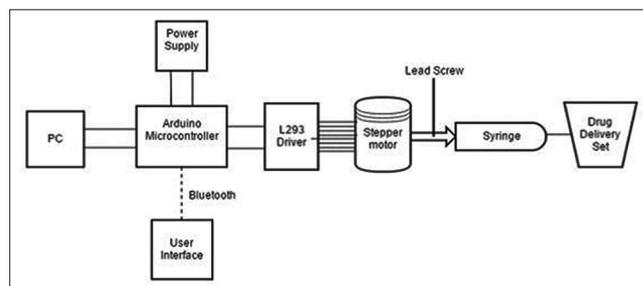


Fig. 2: Block diagram of the designed ocular drug delivery system

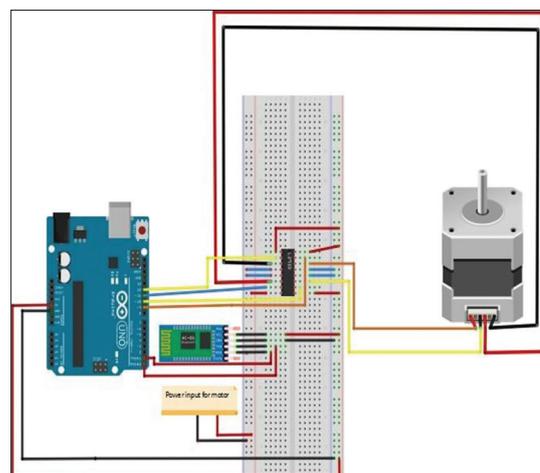


Fig. 3: Circuit diagram of the designed ocular drug delivery system

belong to the bipolar drivers A and C, whereas the red and blue wires belong to the bipolar drivers B and D, respectively, as shown in Fig. 5.

The code used to run the stepper motor mainly comprises of a setup and a loop. In the setup, the pins have been assigned their respective functions, whereas the commands given within the loop is run repeatedly till the number of steps to be covered reaches the specified number of steps. The program has been encoded in such a way that it applies a delay of 20 ms for the coils to energize and de-energize properly [3].

By giving the connections as shown in Fig. 3 and uploading and running the code, it was possible to run the stepper motor according to the number of steps entered in the serial monitor. However, to make the system more user-friendly, the android application Audrino has been developed as the user interface, and this application can communicate with the Arduino software by means of the bluetooth module. Hence, a separate circuit, as shown in Fig. 4, has been created for connecting the HC-06 bluetooth module with the Arduino software. For the bluetooth module, the TXD and RXD are connected to the Pins 0 (RX) and 1 (TX) of the Arduino board, respectively. After which, the code for this module alone is uploaded on to the Arduino board and run.

The bluetooth module has been encoded in such a way that asynchronous parsing is executed. The main digital interface of the HC-06 module is the UART interface. Packets of data are sent as an asynchronous serial streams of bytes. This module is mainly used as a DHT sensor. However,

we did not need that feature, and hence, we only used it for the purpose of transferring the respective data from the Android application to the Arduino board and vice versa. That is why only the basic pins, i.e., the TXD, RXD, VCC, and GND are connected, as shown in Fig. 4.

Since using the Arduino software directly to run the motor can be a little tedious for patients and other users, we have developed a very simple and user-friendly Android application named Audrino, as shown in Fig. 8. This Android application allows the users to enter the amount of drug volume that they need to take into the syringe or to dispense. The entered amount is converted into the number of steps, and the stepper motor is needed to run and then fed into the Arduino board through the program code. The negative and the positive blocks available on the application are to indicate the direction of the motor rotation. If the negative block is selected, the motor rotates in a direction that makes the leadscrew pull the syringe plunger backwards, whereas is positive block is selected, it makes the syringe plunger move in the forward direction.

The algorithm used for this application is shown in Fig. 7. The reason we have kept both μ l and ml is because, most of the times, when the device is to be used for multiple number of patients, a large amount of the drug is required to be filled in the syringe. So, in that case, the user can mention the volume in the range of milliliters while taking the drug into the syringe and while dispensing, they can mention the volume to be dispensed in microliters. The relationship between the number of steps required and the volume of drug was established from the calculations that have been discussed in the following section.

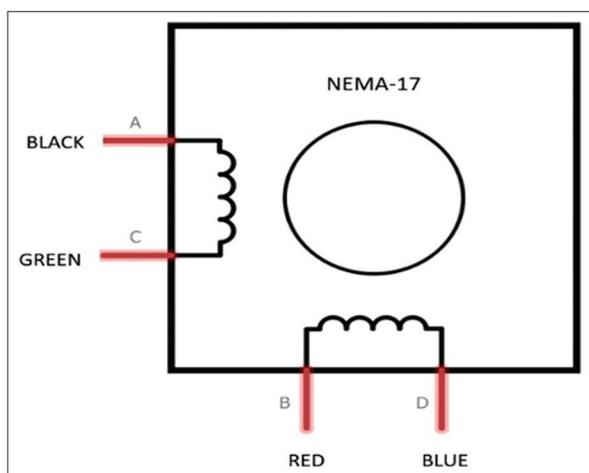


Fig. 4: Wiring schematic of NEMA 17 stepper motor

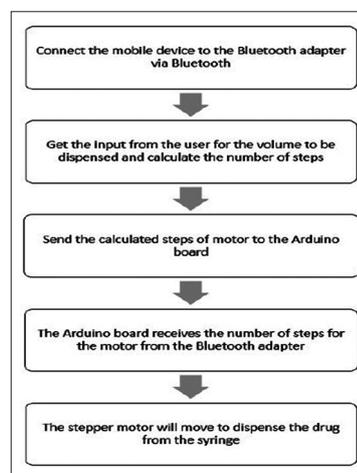


Fig. 6: Algorithm for the coding of the Arduino UNO board

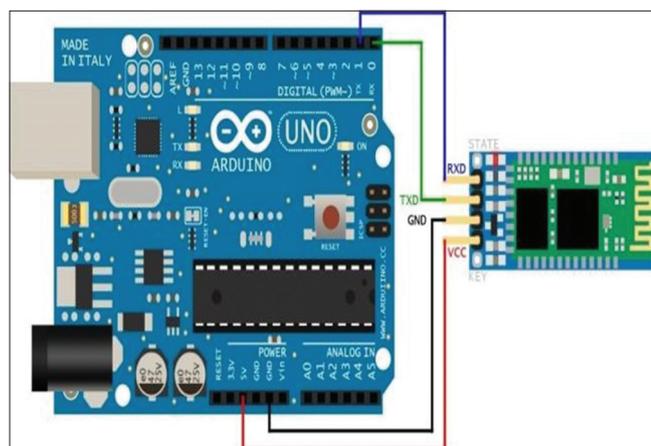


Fig. 5: HC-06 bluetooth module connections with Arduino integrated development environment software

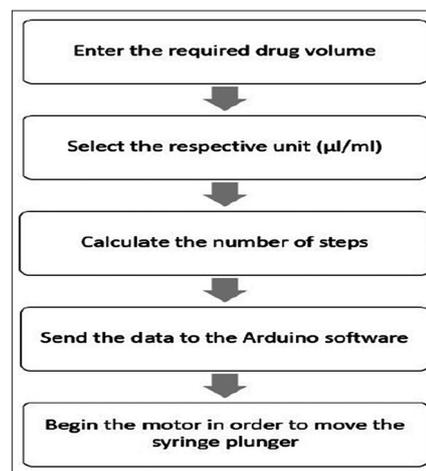


Fig. 7: Algorithm for the Android application

Calculations

To estimate the number of steps required to run the NEMA 17 stepper motor for a particular volume of drug, we need to first calculate the plunger movement. The plunger movement is the distance covered by the leadscrew for one whole revolution of the stepper motor. It was calculated using the following formula:

$$M = \frac{\text{Pitch of lead screw}}{\text{Total number of steps per revolution of motor}} \quad (1)$$

Where M - plunger movement.

The cross-section area of the syringe was calculated after measuring the syringe dimensions, and it was determined to be 161 mm². Using the calculated value of plunger movement, the volume delivered per step of the stepper motor was calculated as follows:

$$V_s = \text{CSA} \times M \quad (2)$$

Where V_s - Volume delivered per step

CSA - Cross-sectional area of the syringe.

To verify the calculations, a random number of 5000 steps was considered and the device was tested. Using the above formulas, the estimated theoretical value of the volume of drug that will be dispensed by running 5000 steps was determined to be 6037.5 µl. Practically, when the stepper motor was run for 5000 steps, the volume of drug that was dispensed out of the syringe was found to be 6 ml ≈ 6000 µl. Since there is a dead space, the slight variations in the readings have occurred. To eradicate this dead space for the sake of accurate results, we have calibrated the device in such a way that an additional 0.21 steps have been added as error correction.

Initially, we thought of delivering the drug to the eye by using a pipe as the device is a bit heavy to be lifted and used. Hence, calculations for the ideal dimensions of the pipe were carried out. However, there were certain aspects that caused hindrance to this feature. According to fluid dynamics, it was not possible to fill the syringe entirely if a pipe was to be used to take the drug into the syringe. When we experimented on this, it was observed that, irrespective of the volume of the drug pulled in, the pipe was always filled with the drug, whereas the syringe was partially filled, i.e., there were a lot of air bubbles. Hence, hypothetically we considered that the drug intake volume will be split into equal amount between the pipe and the syringe. Hence, if suppose we consider a volume of 6 ml to be pulled into the syringe in the presence of the pipe, then the optimum pipe length would be calculated as follows:

Since 6 ml volume will be split equally, syringe volume = pipe volume = 3 ml = 3000 µl

$$l = \frac{\text{pipe volume}}{\text{CSA}} \quad (3)$$

Where l - Pipe length.

This estimated pipe length, however, is not feasible enough for practical application. Different dimensions of the pipe were considered, and it was observed that a pipe of any other dimension would only lead to a lot of wastage of the drug because of non-ideal dimensions. If the pipe dimension was to be compromised, then the total number of steps that is needed to fill in the entire syringe, in the presence of the pipe, will exceed the maximum number of steps that can be used for this particular device, i.e., 5000 steps. Hence, using a pipe to deliver the drug to the eye is not a feasible approach in this certain scenario.

RESULTS

The target of this project was to deliver a drug volume within the range of 30–45 µl to reduce wastage of drug and thus reduce the expenditure on the same. As calculated before, the volume dispensed per step is 1.2075 µl. Hence, for a volume of 30 µl, the number of steps required was found to be 24.845 steps. However, the dead space was a factor of concern, especially in the case of small volumes of drug. Furthermore, since a drop has a minimum volume of 30 µl, it is not possible to deliver a lesser volume without the use of any sort of needle that has a reduced surface area at its orifice. Experimentally, it was possible to dispense a minimum of 30 µl by giving 42.5 steps. Hence, a correction factor was calculated by experimentally determining the number of steps for 30 µl, 60 µl, and 90 µl, respectively. The volume per step was determined to be 0.7058 µl. Hence, there was a difference of 0.5017 µl between the theoretical and experimental values. Calibration was needed to match both the theoretical as well as the experimental values. The calibration factor was determined to be 0.5 µl. To verify the results experimentally, the syringe was filled with water completely, and respective volume of fluid was entered into the Android application to deliver it. The syringe has an entire volume of 5 ml, and exact 5 ml was dispensed using the Android application to enter the value. Similarly, the same thing was done for 2.5 ml too, and the accurate result was obtained as shown in Fig. 9. Following that, different volumes of fluid in the range of microliters were dispensed into microcentrifuge tubes and then measured using micropipettes of the range 10–100 µl. To study this error properly, a graph was plotted as shown in Fig. 10.

Table 1 includes the observation readings that were used to plot the graph, respectively. From the graph, it can be seen that for a particular number of steps, the experimental value of drug dispensed is lesser than the theoretical value. This error is calibrated using a calibration



Fig. 8: Graphical user interface for the developed Android application

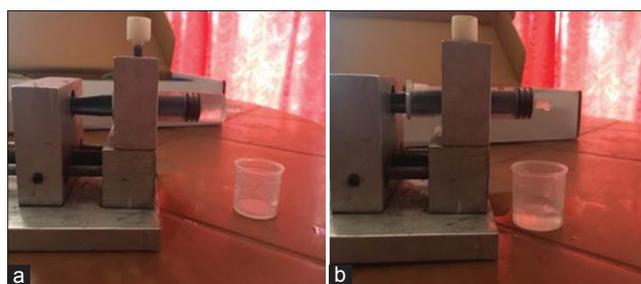


Fig. 9: (a) 5 ml syringe entirely filled with water before dispensing, (b) 2.5 ml of water dispensed into a measuring cap after running the device

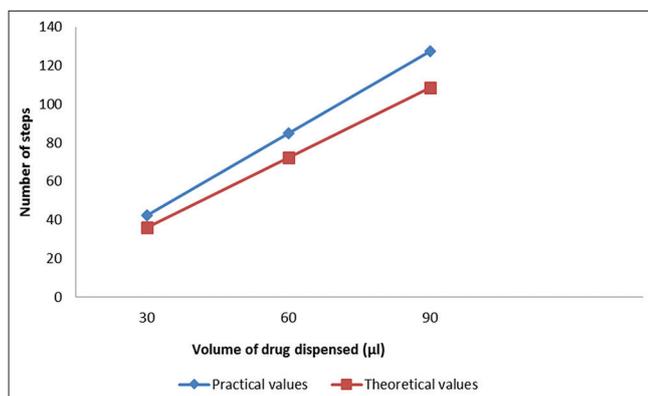


Fig. 10: Graph plotted for a number of steps against the volume of drug dispensed

Table 1: Number of steps required for respective volumes

Volume dispensed (µl)	Number of steps required	
	Experimental values	Theoretical values
30	42.5	36.22
60	85	72.45
90	127.5	108.67

factor of 0.5 µl. The error can be justified as there is always some dead space inside the syringe. To get accurate results, the entire syringe, up till the external edge of the orifice, has to be filled and every time, drug is dispensed using this ODDS.

DISCUSSION

There are several approaches for drug administration that has been discussed by Chen [14-16]. Moreover, cost-effective methods of treatment and diagnostic methods are the need of the hour for Indian population [17-19]. The author has particularly discussed new developments in this field that focuses mainly on clinical trials. Some approaches include drug-eluting puncta plugs, implants, and contact lens. These methods are quite efficient. However, they cause discomfort to the patient and also some of them are invasive like, for example, surgery is to be done to fix the implants in the respective position. In contrast to this, the ODDS discussed in this article is completely non-invasive and causes no discomfort to the patients.

Another fascinating technique that has been discussed by Kumar *et al.* involves the application of a fine glass capillary tube inside the dropper tip to reduce the volume of drug that is to be dispensed [10]. This is also an efficient method and can deliver a drug volume of 20 µl at the minimum. The main drawback of this technique is that it uses the common dropper bottles for containing the drug. As it has already been discussed above, the majority of the patients suffering from glaucoma belong to the aged population. They have physical difficulties such as shaky hands and less strength, so using dropper bottles is not an easy task for them. This problem can be solved using the ODDS we have designed for our project as it is an automated user-friendly device. Considering the fact that the drug is dispensed by the open-source syringe pump, we would like to discuss previous works that involved designing and construction of an open-source syringe pump. Wijnen *et al.* have constructed an open-source syringe pump which uses parts developed using a 3D printer [6]. Since they have used a 3D printer, the cost of construction was less. Moreover, the microcontroller that they have used is Raspberry Pi. Another similar work was done by Akash *et al.* who also used Raspberry Pi to control the syringe pump [7]. The algorithms used in both these works are similar to that of what we have used for our ODDS. However, they have focused more on controlling the rate of the stepper motor, whereas our main focus was to deliver precise volumes using the stepper motor. We have used Arduino UNO as our microcontroller instead of

Raspberry Pi as it is a more convenient choice depending on the ease of use as well as the cost efficiency. Appaji *et al.* have also designed an open-source syringe pump [8]. They tried to predict the flow properties using the Pascal's law and Bernoulli's theorem. The microcontroller used by them was an Intel 8051 microcontroller. This is a very complex type of microcontroller as encoding for it is a tedious job. Hence, to avoid any sort of errors while coding, we preferred Arduino UNO as it comes along with a free open-source software called as the Arduino IDE software. Furthermore, none of these previous works have developed an Android application. Hence, user-friendliness is a matter of concern for these particular syringe pumps.

From the study conducted by German *et al.*, it is evident that droppers are not a very reliable method of ODD [11]. Different dropper tips have different dimensions and hence deliver different volumes of drug. This variation can cause ineffective of the treatment or lead to side effects. Since there is no standard implemented so far for the amount of volume to be dispensed by a dropper tip, droppers are not entirely reliable for ocular disease treatment. Hence, the ODDS discussed is a wiser option that not only helps in delivering precise volumes of drug but also is very user-friendly as it is automated.

In the support of our project, we were able to build on the works done by Ball and Schneider, Rylander, and File. Ball and Schneider had estimated the cost per drop and yearly cost put behind ocular drugs [2]. It was determined that cost can go up to 278.15\$ for 5 ml bottles. Hence, a need for a cost-efficient drug delivery was identified. Rylander *et al.* too studied and performed a cost analysis of ocular drugs [5]. β-blockers being the most inexpensive category of ocular drugs were found to cost within the range of 203.47–657.24\$. This was for a regimen of 3 drops per day basis. File and Patton confirmed that smaller volumes of drug are as equivalent larger volumes so typical droppers lead to a lot of wastage of drugs which leads to increased expense [20].

The amount of drug that is to be delivered to the eye can be controlled using the ODDS which is discussed in this article. The relationship between the number of steps to run the stepper motor and the volume of drug dispensed was determined and utilized to execute the application of this system. This ODDS can deliver a volume precise up to 30 µl. By doing so, users can deliver the precise amount of the prescribed drug to their eyes. This leads to the efficient application of the drug as well as reduced risk of suffering from the side effects caused by these drugs.

If we compare the discussed ODDS with typical eye droppers, this is a better approach as it is cost efficient, user-friendly, and precise. Thus, by controlling the stepper motor as discussed earlier, delivery of a precise volume of 30 µl was achieved using the ODDS discussed in this article.

We know that 1 drop of drug dispensed is equivalent to 30 µl. Since the average number of drops to be administered per eye is 3 drops, it can be estimated that for a total quantity of 5 ml, 56 patients can be administered by the drug at one go. For cases where both eyes of a patient require drug administration, 28 patients can use the total quantity of drug in the syringe at one go. The optimal volume of the dispensed drug per drop is 25 µl; however, the ODDS that has been discussed in this article can deliver a minimum volume of 30 µl per drop. To improve it, it is possible to insert a glass capillary in the dispensing nozzle of the device. By doing so, exact 25 µl drug volume can be dispensed. The system is simple and easy to use, especially because of the Android application which makes it similar to an automated system, where the user only has to enter the required value and the rest of the functions will be carried out by the device itself. However, since the device is a bit bulky, it might be difficult to lift it up to the eye and use it. Another drawback is the device bulkiness. It is understandable that the device cannot be lifted up to the eye level to dispense the drug. To solve this particular problem, a special stand can be designed for positioning the device in such a way that the patient eye lies just beneath the orifice of the syringe so that it is aligned to execute proper dispensing of the drug. By doing so, the device can be used as a patient bedside eye drop

delivery device in hospitals as well as in homes.

CONCLUSION

The proposed system can minimize the excessive use of drugs as well as reduce the capital investment of ocular drugs. The device can be well suited as a bedside setup in eye drop delivery in hospitals as well as in homes.

AUTHOR CONTRIBUTIONS

Surya V: Design and implementation, Srutartha Bose: Design and implementation, Shreya Kulkarni: Design and implementation, Kathirvelu D: Technical advice and supervision.

CONFLICT OF INTEREST

None.

REFERENCES

1. Michael R. Device for measuring and improving adherence of eye drops; 2016.
2. Ball SF, Schneider E. Cost of β -adrenergic receptor blocking agents for ocular hypertension. *Arch Ophthalmol* 1992;110:654-57.
3. Van Santvliet L, Ludwig A. Determinants of eye drop size. *Surv Ophthalmol* 2004;49:197-213.
4. Hermann MM, Diestelhorst M. Microprocessor controlled compliance monitor for eye drop medication. *Br J Ophthalmol* 2006;90:830-2.
5. Rylander NR, Vold SD. Cost analysis of glaucoma medications. *Am J Ophthalmol* 2008;145:106-13.
6. Wijnen B, Hunt EJ, Anzalone GC, Pearce JM. Open source syringe pump library. *PLoS One* 2014;9:e107216.
7. Akash K, Kumar MP, Venkatesan N, Venkatesan M. A Single Acting Syringe Pump Based on Raspberry Pi-SOC; 2015. p. 1-3.
8. Appaji MS, Reddy SG, Arunkumar S, Venkatesan M. An 8051 microcontroller based syringe pump control system for surface micromachining. *Proce Mater Sci* 2014;5:1791-800.
9. Harisudhan R, Kumar MG, Prakash AU, Sathya P. Stepper motor control using Arduino-ATMEGA328 microcontroller. *Int J Sci Res Dev* 2015;2:778-80.
10. Kumar S, Karki R, Meena M, Prakash T, Rajeswari T, Goli D, et al. Reduction in drop size of ophthalmic topical drop preparations and the impact of treatment. *J Adv Pharm Technol Res* 2011;2:192-4.
11. German EJ, Hurst MA, Wood D. Reliability of drop size from multidose eye drop bottles: Is it cause for concern? *Eye (Lond)* 1999;13:93-100.
12. Sklubalová Z, Zatloukal Z. Study of eye drops dispensing and dose variability by using plastic dropper tips. *Drug Dev Ind Pharm* 2006;32:197-205.
13. Cunningham MG, O'Connor RP, Wong SE. Construction and implantation of a microinfusion system for sustained delivery of neuroactive agents. *J Vis Exp* 2008;17:7-16.
14. Chen H. Recent developments in ocular drug delivery. *J Drug Target* 2015;23:597-604.
15. Powar PV. Development status in the meadow of nanostructure magnetic drug delivery system and its promising applications. *Int J Pharm Pharm Sci* 2017;9:10-7.
16. Rajendran R, Balan R, Ganesan N, Thiruvengadam D. Recent modalities in drug delivery via inhalation therapy-An advanced treatment strategy for pulmonary carcinoma. *Int J Pharm Pharm Sci* 2015;7:8-21.
17. Vinupriya P, Hariharan M, Kathirvelu D, Chinnadurai S. Estimation of HbA1c using the complete blood count measures in the diagnosis of diabetes. *Asian J Pharm Clin Res* 2017;10:214-8.
18. Alex SM, Sreelekshmi BS, Smitha S, Jiji KN, Menon AS, Umadevi P. Drug utilization pattern of anti-diabetic drugs among diabetic outpatients in a tertiary care hospital. *Asian J Pharm Clin Res* 2015;8:144-6.
19. Adepu R, Madhu S. Influence of post discharge counselling on health outcomes in diabetic and hypertensive patients. *Asian J Pharm Clin Res* 2011;4:28-33.
20. File RR, Patton TF. Topically applied pilocarpine. Human pupillary response as a function of drop size. *Arch Ophthalmol* 1980;98:112-5.