American Journal of Botany and Bioengineering
https://biojournals.us/index.php/AJBB
ISSN: 2997-9331

Review Article about Arduino Microcontroller Based Syringe Infusion Pump

Mohammed Abdulaal Ramadhan
AL-hadba University / Engineering College Department of Medical Instrumentation

Maher Mohammed Ramadhan
AL-kitab University / Engineering College Department of Medical Instrumentation

Saad Najeh Hameed
Bilad Alrafidain University /Medical instrumentation engineering

Hasanain Kadhim Lafta Alkharsan
Medical Instrumentation Techniques Engineering / Al Hussain University College

Omar Hadi Khudhair Abbas
Al-Kitab University/ Engineering of Medical Device

Received: 2025 19, Apr
Accepted: 2025 28, May
Published: 2025 09, Jun

Annotation: An infusion pump is a medical device used to deploy fluids, medications, or nutritious supplements into a patient's body in regulated amounts. It is an automated device that mimics the working of a standard syringe, where a plunger is pushed in to introduce fluid and that plunger is pulled back for a withdraw. Syringe infusion pumps presently in use are configured in a way that can push a fluid only in a single user-controlled, set flow rate. Not all infusions of medications can occur in the same pace, as different drugs are used at different flow rates. There is also a possibility of human error while configuring and setting up the infusion pumps, which can lead to mal-formations or incomplete infusions. An infusion pump is a device that can introduce fluids into a patient's body without the need of physician's assistance. Infusion pumps can use standard syringes to draw any volume of fluid from any vial size, preserving the sterility of the fluids and retracting the syringe after infusion. It can infuse any amount of fluid into the patient after an account of time, which can be varied per case, unlike current infusion pumps that have a fixed flow rate. It has a bigger display interface for easy use and understanding by physicians for a quick read and set-up, and has the option of pre-installed patient profiles for even quicker set-up of the device. Standard syringes can be used to inject and infuse fluids in a regulated manner with minimal chances of error. The system can use any syringe size, any vial size, any administering position, any infusing timer, or any infusion volume. There are several types of infusion pumps ranging from simple, stand-alone devices, to complex, computer-controlled pumps. Stand-alone infusion pumps are used for pulse generator type syringe pumps. High and low speed infusion pumps are constructed using open source electronics with understanding of fluid dynamics. Safety infusion pumps and tubing infusion pumps are also available. Automated infusion pumps such as software-controlled infusion pumps and fluid delivery sensor-controlled infusion pumps are also used. Stepper motor controlled devices are made use of for heavy duty applications such as missal testing, but face issues of cost efficiency and a broad vigilance range.

1. Introduction

The origin of syringe pumps can be traced back to the world of medical equipment. These pumps are designed to manipulate a liquid contained in a syringe. Application range from drug infusion in clinics to fluid metering in research laboratories and bioengineering. Therefore, syringe pumps have been around for many decades, but their costs often exceed their simplicity. In addition, some applications need multi-channel precision pumps or higher cost-height advantages. This combination of complexity implies a higher cost for buying or using them. As a result, many researchers and engineers do not consider buying high-priced products. There rise laboratories demanding open source hardware and software designs contemplating their limits to modify.

Syringe pumps can be classified into two categories on the presence or absence of motors. The motor-free kind is simpler, but it is impossible to control flow rate accurately. Control is limited to manually-operated suction and discharge, usually characterized by operator experience. The other kind is called motor-driven syringe pumps, which use gear transmissions to drive the plunge of the syringe straightly. Precise control of flow rates can be achieved either by controlling the rotation angle of the stepping motor or the supply voltage of the servo motor. However, there are still many shortcomings. Some old NC machines adopt DC motors with motor drivers, which are cumbersome to adjust, in addition to being difficult to maintain. A customized syringe pump worked well but provided limited piston strokes.

To ultimately reduce design cost, a simple syringe infusion pump was developed based on a microcontroller board. Four parts are composed of the infusion pump, a syringe, combined connectors, and supporting plates: all standard, off-the-shelf products readily available. There may not be any similar work regarding the hardware and software used together. Performance tests are carried out to attain results accurate enough for industrial use. Applications suitable for Arduino-based infusion pumps are discussed as well. [1][2][3]

2. Overview of Syringe Infusion Pumps

A Syringe Infusion Pump (SIP) is an adjustable device for delivering thoughtful and steady infusions of a drug into a patient. A statement best describes the syringe infusion pump as an “endless need” device for the benefit of society in medicine, research, and biology. There are several types of infusion pumps such as syringe infusion pumps, circumferential infusion pumps, peristaltic infusion pumps, etc. The syringe infusion pump demonstrates that relatively high-performance flow control can be achieved by combining low-cost hardware and electronics with fundamental control algorithms. Because of the acceptable volume of the motor driven syringe, a standard multi-channel recruitment infusion pump body is constructed to arrange several single-channel pumps close to one another [4]. The infusion accuracy of the developed multi-channel micro-pump is estimated by conducting experiments, and experimental results demonstrated that the pump can satisfactorily deliver liquid infusion with a maximum flow-rate error rate of 6.52%. As a significant entity of drug administration in healthcare applications, low-cost syringe infusion pump (LSIP) plans are fabricated from easy cypress components and simple software for wireless and hardware flow rate controller actuation [5]. The usage of sophisticated components has been kept to a complete minimum so that all the elements can be purchased under $30. Alternative plans target essential software and electronic designs, which can be developed to test single-channel usage functioning according to a coded wired keypad before transferring to wireless, multi-channel analysis. The SIP was developed from cypress components that can be simply implemented by junior undergraduate students.

3. Arduino Microcontroller: An Introduction

This project implements a microcontroller based controlled syringe infusion pump using a microcontroller unit (MCU), which is used to push an already inserted syringe to control the amount of solution discharge. Potential effects of this project can eliminate the risk of attendant’s error of dispensing the wrong dosage amount. The Digital-to-Analog Converter (DAC) in conjunction with low power Operational Amplifier (Op-amp) is used to generate analog control voltage. The operational characteristics of the syringe infusion pump are changed effectively by controlling the target control voltage, which is converted by the DAC with 256 levels (8 bits). An infusion pump steadily controls the amount of solution dispense by adjusting the dosage amount based on the average moving control period. Automatic control of syringe infusion pump using smart microcontroller. A fully automated off-line pressure sensing syringe infusion pumps are made using an 8 bit ATMega168 microcontroller, which is used to monitor the exerted pressure on the syringe and fuel accordingly [4]. This prototype is made for external empirical studies of the actuator performance and offers easy-to-use software. A low-cost pressure-transducer-based syringe pumps enable the easy use at a large-scale and real-time laboratory of research with the capability of five axes independently at sub-microliter sensitivity.

An Automated Syringe Filling System using a TI Hercules Launchpad was developed. The system is designed with several inputs, outputs, and control components that are all connected to the microcontroller to fulfill the design specifications [6]. Detailed specifications of the system components were analyzed and presented, including the TI Hercules Launchpad microcontroller, display unit, fluid level sensor, valve relay board, and syringe assembly. A general overview of how the hardware components are connected to one another using various communication protocols along with introductory testing results are also presented. Lastly, a base description of how the built-in timers on the Hercules microcontroller will be used for coding functions that require time control are discussed.

4. Components of an Arduino Based Syringe Pump

For experimentation, a working prototype was designed and developed using an Arduino board and a syringe pump. The syringe infusion pump consists of a syringe holder, a DC motor that can hold a syringe, a thumb of a syringe that can move in the forward and reverse direction, and the Arduino board that helps drive the motor to move the syringe thumb. The syringe pump operation is asked in just two ways, one in clockwise and another in anticlockwise direction using a switch with the help of relays. It uses the IR sensor to detect the low fill of the syringe. On tumbling of the switch button, the motor moves the syringe thumb in the forward direction and locking the switch. If a small high voltage is applied in reverse direction it will knock the switch on the off position and the thumb of the syringe stops and no voltage will be applied making it latch off. When the syringe is filled properly, the IR sensor works to sense the low fill of the syringe. The motor control uses the Arduino Mini and a ULN2003 Darlington array IC to drive the motor. A 50W grid tie inverter circuit is constructed that works with PWM control and cannot be interrupted very easily by external noise [4]. Arduino Mini get power from a 12V battery. The battery is charged using a 12V battery charger. In the case of grid failure, there is an LED which indicates the grid failure, immediately the inverter switches ON and provides the AC power supply. When the grid voltage returns to normally start up, the inverter takes several seconds delay and the normal AC supply takes over, and it automatically switch OFF. The entire circuit is constructed on a 17mm x 10mm PCB which fits very easily in the allotment. Two ways to implement a digital lock circuit are presented. The first way has the advantages of low-voltage operation low power consumption and small size, yet the numeric input is somewhat complicated. In contrast, the second approach is larger, very robust in everyday applications, power-consuming, and quite simple at the same time [7].

4.1. Microcontroller Selection

The microcontroller in the design is a widely used ATmega2560 microcontroller which is from AVR family of microcontrollers. The microcontroller is programmed using Embedded C. The required libraries for programming the microcontroller are installed in the Arduino IDE. It is a low power design and is a low cost microcontroller. The programming code is made reusable and the code is uploaded via USB chip connection with the development board. The programming is done in a modular approach so that the program can be reused in other projects. Each subroutine is handling one separate task so it can be tested independently.

A C compiler is used to compile all the higher level language code into equivalent machine code for programming the microcontroller. All the interfaced units with proper connections use GPIO pins of the microcontroller to communicate with the code. Interfaced units such as LCD, motor controller, pressure sensor (with relay driver) are provided with C subroutines for the code, which can be called at any time to execute specific tasks. The C compiler is a standard code development kit and most used for programming the microcontrollers. It has powerful built functions and a very simple to use structure to develop big code.

During the technical implementation section, the technical as well as the software chemicals used and their role are formulated with ratings. Components were chosen after careful calculation so there is very little chance that the project will fail on implementation. Factors such as voltage, current and rating were taken into consideration before selecting the components. All the pieces of equipment were chosen in order to build a working prototype for testing and demonstration purposes. The barrel pump is chosen to be of TLV type and the pipe size before and after the pump is also selected accordingly. The pressure sensor is of a specific brand with a model number. Relay drivers are used to power the 24V AC compressor pump and other relay drivers to activate/deactivate the valves. A transformer is good enough for the pump and relays. The relay drivers that will control the valves and the barrel pump, MOSFET drivers are used to demodulate the input signals at higher speed levels. A circuit driver and optical driver are used to drive the relay coils safely while lengthening their lives.

4.2. Syringe Mechanism

The syringe placement mechanism is done using a DC motor. The motor will rotate a shaft and to this shaft a connecting rod is attached. The other end of the rod is connected to the nut of the vice which enables it to be perpendicular to the vice and parallel to the wall. As it rotates, it pushes downward or upward the vice depending upon the direction of rotation. This vice is used to accommodate syringes of different volumes. The syringe is fixed in place using the vice. The syringe is placed with the face cover facing the outside. The user has to place a filled syringe with the face cover rotation in anti-clockwise motion. The syringe must be fitted below the broad end of the clamp because it holds the syringe crus.

The syringe at first is placed in the broad end of the clamp, and once the mechanism rotates in a vertical position it gets fixed and can't come out of the clamp. Then the other end of the cam is fixed to an L-shaped arm which holds the pin, the pin is fixed to the stepper motor. The syringe mechanism is made to move in a semi-circle. As it rotates, the smaller end of the syringe coat is pressed and the fluid-filled out from the syringe. The face cover can also be rotated in the clockwise direction so as to unclamp the face cover and take out the syringe. The advantage of this mechanism is it can accommodate syringes of different volumes and widths too.

At first the stepper motor is rotated in clockwise motion and the weight is lifted up, After reaching the bunny position, it stops, and it is rotated in the anti-clockwise direction, upon rotation the syringe mechanism also rotates along with it, and it discharges the droplet on the slide. After completion of the patient one dose is discharged, and the vice clamps the new one this happens in a pre-determined interval of time as appointed by the doctor and the arm of the syringe mechanism presses it for some time to ensure the entire fluid is discharged.

4.3. Motor Drivers

A motor driver is a relay, transistor circuit, or integrated circuit that controls a motor. The types of motor drivers available these days are H Bridges for DC and Stepper motors and unidirectional drivers of DC motors with PWM control. Motor drivers serve to increase the current supplied to the motors and thus need external power input. The driver provides the correct electrical signals to control the operation of the motors and reduce the workload on the microcontroller.

A DC motor is used to drive the mechanism of the syringe pump. The syringe drive mechanism consists of a DC motor, belt, and pulley combination with a syringe holder. The motor rotates the pulley by applying voltage across it; due to which, the belt is pulled. The syringe holder is attached to the system of the belt; hence whenever any motion is generated in the belt, the holder is moved which leads to a change in the volume of the syringe.

The driver is used to control the DC motor of the syringe drive mechanism in the system. It is a 16-pin dip package motor driver IC and consists of four H-bridges. An H-bridge circuit can control the direction of motion of the motor. Each H-bridge requires two signal inputs for controlling the motor. The H-bridge only works with the enable signal being high. Each H-bridge can control a single motor. Therefore, a total of four motors can be controlled simultaneously with a single IC. Most of the ports in the IC are internally connected to ground or VCC; only one of two Enable signal ports is used to control the motor driver. A total of four signal inputs are provided to control the two motors. The motor drivers are enabled with a low-active enable signal. Each input signal port is connected to the Arduino port through an opto-isolator. The opto-isolators act as voltage level shifters since the operating voltage of the driver is +5V and that of Arduino is +3.3V.

The DC motor drive circuit is switched on, and the current flows through one of the four MOSFETs and that of the left side in the H-bridge; hence the motor rotates in one direction. The switching would be reversed by controlling the signals applied at the input, which would lead to switching on the other MOSFETs of the H-bridge. [8][9][10]

4.4. Power Supply Requirements

The main power source is recommended to be a wall adapter. Since most wall adapters have variable voltage DC outputs one can supply 5v to the pump to power the Arduino or the motor drivers. If only 5 volts supply is used then the pump will just run on the battery backup. This is the recommended method so that one can control the run time and the voltage. A 6-24 volts supply through the motor driver and Arduino will take care of the other components with the help of MOSFETs.

The whole system is tested with and without wall adapter. The motor drivers had a higher supply voltage than a wall adapter hence all the control, queue, energy requirements, timings will be kept in the voltage configuration memory. Powering the motor drivers directly from the wall adapter or power supply on the rack ensured a lower supply to the Arduino over the specified sequence. The waste codes in the system are tested too. The limits of queue length and syringes are also verified. The different cases of queue timings such as queue overlap and urgency conditions are satisfied. If the urgency on to the syringe does not fix the urgency it nor the comments given to the sensor it ensures that no syringe pump is setup and returns back to setup.

Overall a prototype worked on volt meter regardless of the pump just to check the positioning control. As the motor drivers took a negative voltage the other motor driver circuits used were not integrated for the wall adapter. On reconfiguring another Arduino nano was used with simple MOSFETs and transducers hence better environmental controls and decent sensitivity limit. All the vacuum pump and sequential control systems were demonstrated and a proper motor driver circuit configuration using low cost IC depending on the external supply and teeth count was designed for a new ATM. So, all the components, architectures, CAD for design, hands on skills, chips etc. were generated/surfaced, hence all the chain components were able to cause the effect in the system.

5. Design Considerations

The infusion pump includes three crucial components that ensure accurate delivery of medicine to the patient: a microcontroller, a motor driver circuit for the stepper motor control, and the syringe holder. Each of these components is described in the following sections: the Arduino Uno microcontroller, assembly, and coding, the motor driver circuit, and the syringe holder assembly. To operate the Arduino, the Arduino IDE must be previously installed on the computer and the Arduino Uno connected to the computer via the USB cable supplied with the kit. A software library is included in this report which contains the code for the Arduino Uno microcontroller [4].

In order to run this program, the library should first be opened in the Arduino IDE. The Arduino IDE will then compile the library. If no errors are present, run the library by clicking on the tick button in the menu bar. A secondary window will appear listing the compilation options. Click OK in this window to continue. After a few seconds, the compile window should appear again displaying progress and possible errors. If an error is present, the red error message on the bottom of the window should be noted. After the library is compiled without error, the library should be uploaded to the Arduino Uno by first checking the communication port in the Arduino IDE and ensuring it matches the port in use by the correct Arduino Uno microcontroller. Then, the arrow button in the menu bar should be clicked to upload the library to the Arduino microcontroller.

Syringe pumps use either a stepper motor driver or a ferromagnetic stepper motor to precisely control the movement of a plunger within a syringe barrel. Here, the Liquid Crystal Display (LCD) interface 16x2 is used to present the mark and feedback to the user. LCD on embedded system is used to present warning and display the number and appropriate action to avoid the machine being dangerous to everyone.

5.1. Flow Rate Calculation

To accurately measure and control the flow rate of an infusion pump, precise hardware and software that can calculate flow rate must be designed and implemented. The software design consists of two parts: the flow rate calculation part and the sensor measurement part that integrates user input and displays the measured values. The flow rate design is described first. The software must take inputs from the sensor and user inputs, integrate them into suitable data structures, and calculate the state variables for the infusion flow rate and infusion volume. Therefore, the pump is controlled on the basis of state variables.

The hardware design is based on an Arduino microcontroller, L298N motor driver, and dose and inertia velocity measurement sensors. The hardware design must have optical sensors to check if there is an obstacle in the track, a multi-turn potentiometer to change the angle of the motor, a movement motor to divert the track, and an Arduino shield to connect the optical sensors, potentiometers, velocity measurement sensor, camera, etc. The servo motor is controlled by the Arduino motor driver L298N, which is enabled by the Arduino. A bolt is fixed with a gear that is attached to the motor shaft. Another gear is attached to this gear so that the loader track can rotate and curve. Two optical sensors measure the reload position such that the track can be adapted to the shaft angle.

A multi-turn potentiometer measures the angle of the servo motor. The Arduino checks if any of the optical sensors are active. If either is true, a zero-degree angle is set for the potentiometer. Otherwise, the angle is updated every hundred milliseconds Integrating User Input and Measurement Display. The hardware design consists of two large boards and quite a few circuits, which have been tested and verified individually. It is possible to piece them into a complete circuit. Each component was verified individually in programming and debugging before assembly [4].

5.2. User Interface Design

The user interface of the test system was designed using MATLAB's App Designer. The main screen of the interface is shown in Fig. 36. It contains two sections: section (i) displaying the video feed from the webcam and section (ii) displaying the analysis and results, such as the percentage similarity of contours and diameter measurement in millimeters [6]. With the push of the "Start" button, the test is initiated and the resulting image is displayed on the live feed subplot. The recorded image gets transferred to a handler function that can process this image, and the results are displayed accordingly. The background of the user interface can be changed to dark grey along with its components, during the testing, to give it a better look.

There are several mentions of possible regions of improvement, which were discussed prior to coding and while designing the program. Some areas of suggested improvement include having the analysis process highlighted/noted on the live feed, better-contoured images so that results are not missed, and a more thorough investigation of measuring diameters of sectioned vials.

User Interface elements (labeled in Fig. 36) Section (i) displays the live video feed from the webcam. This is achieved by using the camera, a side function, and a timer object in MATLAB. The "Start" button initiates the video feed. During the feed, captured images get processed. A subfunction captures and processes 20 frames switching between analysis functions every 5 frames. This provides instant feedback to the user, so they know the program is running. The testing subfunctions are contained in a separate file, so the main function runs smoothly and has a clean layout. A new analysis window opens containing the analysis image, generating the warning, "Please wait while the image gets analyzed…" during processing. Once done, results are displayed in section (ii) and a "Save Graph" and "Clear Live Image" buttons become active.

5.3. Safety Features

The infusion pump has some important safety features that should be clearly defined and explained. All of these features work in a way to protect the patient and the system. If the patient is protected the system will be protected. While this is just a prototype there are plans to go further with the project. With further developments, the key safety features should be developed further and implemented in the final design. For the prototype, however, a few key features are important. These features are emergency stop button, accuracy of the motor, and movement limits.

First and foremost, there is the emergency stop button. Safety switches are very important for any system. Any system that interacts with a human must have a way to immediately stop all motion. In the case of this device, the emergency stop switch will cut power to the stepper motor, disabling it. Once the system is turned back on the motor will home to the start position. With the use of relays instead of directly hooking up the motor drivers to the Arduino, it is easy to add additional features like the emergency stop switch [6]. Another important thing to consider is the accuracy of the motor. It is easy for a motor to lose steps. This usually happens because either the motor is not getting enough power or it is trying to turn too fast. Both of these are something the software should monitor. The software should also check that the stepper motor hasn't lost any steps by getting feedback from the motor.

The last main safety feature to consider is the movement limits. The motion of the syringe should never exceed the limits of the motor. The current prototype is built up in a way that makes it impossible for the syringe to exceed the limits of the motors. The vertical motion is limited by the shape of the system. The horizontal speed is limited to avoid damaging the syringes. By combining these features together it is easy to see how and why the prototype is effectively designed with safety features in mind.

6. Programming the Arduino

In order for the microcontroller to accomplish the above tasks the microcontroller code needs to be generated. A combination of the Code Composer Studio IDE for development work and the Halcogen tool made by TI for bringing in drivers opening the whole microcontroller's capabilities. Halcogen allows the user to select the specific drivers to bring over and select some of the basic configuration settings. Once this all happened, Halcogen generates all of the include files to drive the exact modules of the microcontroller chosen inside Halcogen. These files were then copied over to CCS and placed into a placeholder project where user-written code would be placed in conjunction with the driver files. The entire project is C/C++ based with project files and include files to run everything. The microcontroller's surface mount form factor required a custom printed circuit board to mount all components to.

The schematic shows the components required for generating the power for the system, supplying the stepper motors with direction and stepped signals, and finally the components required for interfacing to the resistive touch LCD display. The resistive touch LCD if connected to a 5V power supply and using a serial interface, receives power from the 5V DC to DC converter and signals from the GPIO pins used configured as pin output. The power domain for the power MOSFETs to control the three stepper axis can be boosted up to 30 volts from a higher voltage source using the DC to DC converter. This system is a switching converter with step up capability necessary for correctly providing an ample voltage to the power MOSFETs which switch at frequencies approaching 16KHz. The power circuitry was made with ease of expansion in mind, should the system ever need more stepper axes additional power circuitry beyond just another extraction from the DC to DC converter would not need to be added. The power is then sent to six N Channel MOSFETs to send a high side PWM signal to each stepper controller circuitry which interprets the signal and generates step pulses for each stepper. The steppers exert push and pull forces via fused mechanical hybrid mounting clamps stuck to vibration and acoustic isolation viscoelastic material. Two thin 5V passive piezoelectric elements drive steel plates from the clamps at a natural frequency of 30.93 Hz.

6.1. Basic Arduino Programming

If the automated syringe pump and microcontroller control code were to be produced in the real world, a thorough working plan illustrating how the pump would be produced would be made priority. This would include a breakdown of parts needed for production and in greater detail how the microcontroller code is set up. With the default microcontroller code, it would be difficult to decipher and utilize in different hardware setups unless the code itself is modified. The control code would need to be function commented throughout and a description of the specifics of the table driving the LCD would need to be attached as documentation [6]. Ideally, a text box would be added allowing the user to enter some parameters in order to translate directly to tables. Additional functionality could easily be added to the control code once properly commented and organized for use.

The PC computer program was written with Java utilizing the Processing interface. Processing interprets Java into a program that runs outside of the IDE by utilizing libraries specifically written for interfacing hardware from USB ports. This processor automatically walks through the needed configuration setup for UART communication that is ultimately managed by the selected libraries. With libraries accurately configured, messages can be automatically sent and received from the hardware being utilized. A small issue that came up was sending an Error message to the computer program. Simply sending an Error message to the PC for reporting did not suffice. In order to prevent potential confusion in incoming message format, it was decided that sending a specific character be first implemented to indicate an Error message was about to be sent.

6.2. Libraries for Syringe Pump Control

The open-source libraries selected for the project control were the AccelStepper library and the DCCServo library. These are open-source libraries for controlling stepper motors and servos. Arduino IDEs come with many useful built-in libraries, some of which are required to be included in any Arduino code [6]. Some other libraries might need to be downloaded and installed manually. The detailed process is not repeated here. Authorities of open-source libraries are referenced, and these libraries may change over time. Usage and specific function references should refer to the library sources directly for the latest credible understanding and help. The libraries can rather easily be adapted for custom motions by editing the library files directly, however, one must keep in mind that this affects all programs using these libraries [4]. Both libraries are needed in the control code. The control function creates a DCCServo object named tiltServo for a DCCServo object. DCCServo(int ser_num, int pin0, int pin1, int pin2) is the constructor. The pin numbers set to this DCCServo object are connected to pins 9, 3, and 10 on the Arduino board. The control function also uses the setServoRange(int min_val, int max_val) function to set the allowed servo angles. It sets the min angle to be 0 degrees and the max angle to be 180 degrees. Furthermore, working more closely with the DCCServo library, the pin modes for use with pulse output rate limits need to be set in the control function, with this example set to a 10 millisecond maximum pulse output rate on the attempt to retain its 38 millisecond period, and the setServoPulseMax(int mex_pulse_us, int max_pulse_ms) can filter pulse length for each servo within the DCCServo object.

6.3. Implementing User Inputs

Once the microcontroller powers up, it initializes the display and configuration settings to the system’s defaults, which include 18 gauge needle size and a user-unlocked system. The syringe fill menu, which allows the user to choose from either a syringe fill or user setup option, is blinked on the screen in blue and white. This will continue until the user touches a location on the display at which point the fill menu will freeze in that state and the user will be prompted to perform an action [6].

At the beginning of most task selection menus, the information display screen will clear previously displayed information, and display the task prompt, while the keyboards text entry and other function buttons reset. The currently displayed text entry area and keyboard are displayed in light blue. Depending on the users input all keyboard buttons either have no action or have a defined function. An action display menu, which is similar to the fill menu but will have its own prompts, will also freeze on the stage selected. The upper half of the screen will display the prompt, and entered information, while the lower half of the screen will display the keyboard and other function buttons. The function buttons will all have the same definitions as previously mentioned in the other display, and this elevator type action cannot occur until all necessary fields have been filled. Once all fields are filled, the keyboard area is removed revealing a continuing display with the entered data, and a confirm and edit option will be displayed as the last function buttons.

The confirm button will cause the information supplied to be saved by the microcontroller so the syringe filling process can be initiated. Confirmation prompts are sent in blue text. Prompts to show the data still saved in memory will occur and blinking a semi transparent text box will remind the user to input the information as properly as possible. The led text display will enter pause mode as a field is being confirmed and then reveal the data stored already. On the desktop at least two separate sets of information each containing inaccessible import or account numbers is required. Accessing the cylinder programming menu without entered user credentials will interrupt the functioning of the system until a proper user is entered. [11][12][13]

7. Calibration of the Infusion Pump

The requirements for calibrating the infusion pump are to have a digital display to show the readings, an input method for the user to enter a calibration measurement, a ground zero, and to have the input controls for the syringe type (medium and plunger volume). Due to the previous example using rotary encoders to measure movement, this idea of feedback was taken and then evolved into using non-contact Hall Effect sensors. Since there wasn’t proper positioning feedback anywhere else on the design, this was a crucial point to add to the design for a great final product. The Hall Effect sensors were chosen because they are simple to implement, have readily available and cheap off-the-shelf parts, and they can be used not only for feedback but also for input controls to the microcontroller, thus reducing the number of switches and buttons. The thought was that with the simplicity of the sensor and embedded coding, the integration would be easy and allow for a great consumer product with a modern look. The first part of integrating the Hall Effect sensors was to design a PCB for the sensors. The PCB was not only meant to give a platform for the sensors but also to directly plug into the Arduino microcontroller development kit breadboard and made to fit precisely to the spacing and layout of the magnet holder in the shell housing [6].

For the embedded coding portion, the first step was to gather libraries needed to read the sensors. A simple code was then developed that softly biased the reading so there were no readings right away and then allocated a range of values that dictated a specific product when latched. Clamping the reading to particular values during a check loop dictated which product was being read. This process of reading products was very tedious and took hours of switching sensors in and out of the prototype and taking data and readout values. Once the data was taken, the adjustment in coding was easy and could be changed with just a few lines of variables; however, this is a good example of an issue that could be considered a non-function component once the final design takes form.

The product sizing, scaling, and overall consumer feel of final product prototypes were not only crucial but also difficult to iterate through on the current design. With the consumer feel of the final product being dubbed as critical to the success of the product, controlling this aspect of the design was imperative. With a candidate product already designed and on the market, it was important to capture the same aesthetic appeal and feel in pumping so product decisions needed to be made carefully. Immediately upon prototyping the pumping case for the previous plastic design, the outer material was thought to be taken from the current pumping case. After drilling out the front face for the LCD screen and fiddling with some basic PCB holder clips with the machine shop, the process of finalizing the product cases had begun.

7.1. Calibration Procedures

As a research prototype, the syringe pump requires a suitable setup be developed for calibrating the linear motion of the thrust rod, in order to have a known volume of infusion as a result of a known linear motion of the thrust rod. The resolution of linear motion is determined by the compound gear system that increases the accuracy of the stepper motor. The stepper motor is located in the stepper motor assembly, together with the gears. The output shaft of this assembly rotates in ratio with the stepper motor's rotation and moves the thrust rod linearly (horizontally). The linear motion of the thrust rod is measured by a linear potentiometer. The calibration setup incorporates a syringe variable within the range of 10-100 mL, steps to manual infusion, a trigger for initial positioning, and a button to reset the linear motion.

The test is conducted as follows: the pump routine is run based on the timing, the upward motion of the thrust rod is held for a few seconds (to prevent interrupted readings), and the downward motion of the thrust rod is performed. After sequencing for 14 steps, the calibration result is processed (proportionality coefficient of linear motion vs. infusion volume is determined) and stored. The calibration resolution is measured in the same manner and results in how many steps of motor movement correspond to how many units of linear motion (thrust rod displacement).

7.2. Testing and Validation

Model validation is a technique to check if a model is reliably producing accurate outputs. For a syringe pump, this means that the user would like to see rate and volume measurements to ensure satisfactory performance. To meet performance specifications, the Syringe Infusion Pump was required to reach the rates of 2-10 ml/hr over a 20 ml syringe and a duration of multiple hours for a 60 ml syringe at a rate of 50 ml/hr. To check if the specifications were satisfied, the SIP was tested using a 20 ml syringe with a de-ionized water solution. As an initial control test, it was concluded that the valves were closing satisfactorily to prevent backflow. The dispensing valve was turned on for 5 minutes. To measure the volume precisely, a timed and controlled test was performed using a syringe pump. The expected value of the test was calculated: ml was anticipated over 300 seconds, and very accurately achieved by the SIP. It was also concluded that a designed 60 ml syringe could be built to comply with the pressure requirement for human use.

In another experiment, two tests were designed to simulate a push on a 60 ml syringe with 20, 40, and 60 ml loads imposed, one at pump velocities of 1 and 2 mm/sec, respectively. One experiment failed to push all the loads as the loading only allowed 400-450 grams of maximum thrust. All other tests worked satisfactorily. Digital controls were validated by developing a communication protocol and creating an application for a smartphone. Validation was completed by first having a phone transmit simple information to the SIP controller via Bluetooth. A test was then completed to see if the pump was able to accept a command from the app to run the second amplification method. After some initial challenges, the app triggered the pump at the intended frequency. Validation was completed for both the camera and the circuit built to enhance the input. To reduce some program run time, the serial input character buffer option was changed to not use a background task. While still waiting for transmitter tests, it was noted that several next steps might be completed simultaneously to expedite project completion and performance prediction. Notes on the mechanical design of the Robotic Mini-Babbler are included, multiple ranges of motion may be calculated and built for the Robotic Mini-Babbler. [14][15][16][17]

8. Applications of Arduino Based Syringe Pumps

A syringe pump is an automated device that can dispense fluids at a set flow rate and volume. When the limits are hit, it will stop automatically. The need for this invention arose from the desire to automate tasks associated with measuring out doses of liquids when conducting experiments and testing patient safety in clinical settings. There are many different industries that require the accurate dispensing of fluid [18]. The medical industry is one in particular that greatly requires the use of syringe pumps. Automated syringe pumps are especially advantageous in clinical settings because they can increase efficiency, improve injection technique, increase precision, and reduce operator fatigue. Many healthcare professionals regularly inject individuals with Botox, skin filler, or varying medications with a syringe; however, there are currently no ergonomic devices designed for this purpose. Improper injection technique can lead to undesired patient reactions, bruising, or longer recovery times [4]. If the injector is tired or bored, it is easy for mistakes to be made that can alter the flow rate, concentration, or dosage.

The syringe pump eliminates the need to manually inject, allowing injection properties to be stored and recalled with a few button presses. One of the major tasks of a clinician is to interact with patients, but it can be difficult to both inject fluid and maintain eye contact. The design allows clinicians to give their full attention to their patients, which can build trust and contribute to a more pleasant experience. Such a device can also increase injection precision; this is important because a missed dose of a medication can have adverse effects on a patient. Patients will benefit from a consistent dosage of fluid and less pain from the procedure. Compared to using conventional syringes, the automated syringe pump is consistent in the rate and volume of fluid given. When the rate and volume are constant, the pressure at which fluid is injected will also remain constant. While the benefits of fluid consistency could improve the experience of patients getting injected, there are also benefits that the industry could provide. Less scratches and missed doses mean that there will be fewer side effects after the patient procedure is complete, more repeat patients, and fewer lawsuits filed for errors.

8.1. Medical Applications

A mechanism for a syringe pump infusion with a microcontroller by Arduino, which is controlled to achieve an appropriate infusion rate in the medical field with the integration of controlling functions, monitor selection, and alarm warning. The idea encountered is to create an Arduino-based infusion pump that is efficient, cost-effective, and suitable for low- and middle-income countries. The technology will be capable of setting the flow rate and the infused volume. Once it starts, an alarm will be stated if the pump is disconnected. Moreover, ready alarms will state the time for turning off the pump and completion of infusing volume. New investigations are directed to the area of developing embedded control to cover medical applications [6]. An infusion pump is a medical device that administers fluids, such as nutrients, medications, or blood products, into a patient's body in a controlled manner. Infusion pumps are widely used in hospitals, home care, and veterinary applications to deliver specific amounts of fluids over a specified period. By optimizing flow channels, microchannel-based infusion pumps can also be used for infusing products into other fluids or for mixing two different fluids at discrete flow velocity ratios. However, few inexpensive microcontroller-based efforts have been made toward the design of inexpensive infusion pumps suitable for wider use. Microcontroller-controlled infusion pumps have recently emerged as a robust and low-cost alternative to conventional pumps.

The primary focus of this work is on the design, construction, and characterisation of an Arduino microcontroller-based infusion pump designed for low-cost operation. The infusion flow rate and infused volume are controlled via a menu-driven interface using a keypad and LCD display and can briefly be stopped for patient monitoring. Once the infusion starts, the control algorithm continuously monitors the flow rate by measuring the vessel parts to accurately compute the infusion duration. Control logic calls trigger an alarm state if the pump leaves the designated flow range or is inadvertently disconnected. Data logging via a serial link is handled throughout the infusion, and a summary is printed on completion. A five-hour 240-ml flow test showed an average flow rate of 804±1μl/min, and the pump lost 2.2% of the delivered volume.

8.2. Research Applications

The scope of the system is extensive and varies from routine clinical to research uses. For application in the biomedical field, the syringe infusion pump offers advantages such as affordability, ease of access to a wide range of components, ease of construction, and simple programming due to the widely available libraries. It may be implemented in ongoing espresso-drip studies, investigations of behavioral pharmacology, anaphylaxis, and numerous research studies on the effects of drugs such as social and non-social buffering behaviors.

Research investigators designing units with multi-syringe microfluidics, unique parameters concerning size, material, cost, and syringe type should consider units employing the dual-pump program. Typical microfluidics testing experiments use millimeter-scale chips made of Polydimethylsiloxane and glass. Such materials are incompatible with pyrolysis and combustion methods of infusion testing on the order of micro- and nanoliters, as needed for comparable control in other systems. This permits a novel infusion test of pressure and flow in silicone-based micro- and nanoliter pool chips, direct contact with the syringe infusion pump, and study of polymer-composite chips to replace glass for less elastomeric polyurethanes. For flow measurements, a pressure-output control loop (PID) is included to facilitate future scaling of the output pressure control to pump systems with nanoliter-topicoliter flow rates.

Tightfitting lids for sealed syringes that are porous to gas and viscous liquid allow a marketed pressure pump to limit pressure transients to 75 ms at baseline flows of 10 µL/h and to proportionality gains of 500 to 1000 h/Pa. This system achieves a lower noise flow baseline (0.01 µL/h from 300 s) and improved resolution from (10%) to below 0.5%. An expanded toolbox will permit a wider range of infusion testing applications and some ex ante unit testing [4].

8.3. Home Healthcare

Revolutionary programmable logic controller (PLC) LoRaWAN modem is used to provide cloud capabilities. The proposed cloud platform can be accessed through mobile devices using the technology, database service, and a cloud-based API synchronizer. Cloud-see and cloud-hear functions are online services that can visualize local data and remote-control devices using a bidirectional streaming platform. Geolocation automation systems are utilized globally. End users can obtain services in device internet communication, data acquisition, and control without the configuration and setup of a complex local gate.

Simulated real-time telemetry and controllability of a Soybean to Ethylene-Restorer bacteria co-habitation system are demonstrated. The nation’s safety and efficiency can be increased by identifying unknown sensors, operating remotely even during disasters, and using biowearable devices or drone automation. A comprehensive IoT-based smart healthcare system integrated with an elastic cloud computing platform is offered. The system consists of client devices and an elastic cloud architecture with a fog node. The biowearable devices can ensure real-time and accurate health monitoring and prediction by integrating various empirical and physical modelling methods and multi-modal deep learning algorithms.

A device-free mobile sensing method using wireless channel state information in the constructed indoor environment is proposed. The channel power spectrum density varies linearly with only the prespecified element of the array antenna over receding block transmission. By eliminating multipath contributions from the channel power spectrum density by the neural network channel decomposition, occupancy detection is provided. A method is proposed to enhance the robustness of an ultra-low energy depth camera. Together with the hardware design of an enhanced wide-angle lens, lattice grating, and adaptive infrared light source, a complete prototype of the intelligent maze-solving robot is developed. Further improvements and other applications are discussed. [19][20][21]

9. Challenges and Limitations

Pumping systems generate mechanical and pressure changes to create fluid motion. Continuous, steady-rate flows, alternating flows, and "peristaltic" mass transfer are achieved. Required conditions include pumps that disperse, compress, mix, aerate, spray, atomize, generate mist, filter, scrub, agitate, dose, inject, aspirate, and combine fluids with solids, gases, and electromagnetic energies. Equipment controls fluid flow, composition, rheology, agitation, treatment, and process requirements.

Pumping systems are essential in laboratories for pumping sample reagents into detectors. Fluid handling devices require electrovalves, solenoids, electronics, and sensors for the pneumatic apparatus to safely function. Syringe pumps inject sample solutions into microfluidic devices. Today's fuel pumps produce several heating and cooling circuit flushing channels to increase durability.

With technological advancement, hobbyists and researchers created therapeutic infusion devices and brain stimulators. Few advancements have been made in the domain, and many researchers rely on 3D printing fabrication.

Single-inline package encoder provides a simple and cheap continuous optical encoder suitable for direct fitting on radio-controlled servos. Acceleration and deceleration are improved in closed-loop servo control, which eliminates smooth start, restart, acceleration, and deceleration at an operational point. A closed-loop control electro-pneumatic soft gripper platform uses dual-mode single-chip microcontroller and dual-core inter-integrated circuit and serial peripheral interface, suitable for pneumatic branchless soft printed circuit boards.

There is still ample room to upgrade the low-cost syringe infusion pump platform. With further studies, the pump could be improved by improving the simple microcontroller to a more functional one, like an embedded system with a larger capacity, along with GUI-enabled software to enable smartphones and tablets for more effective and convenient control. A continuous recording display feature could record and truncate the pump-dependent experiments results, benefiting teachers and laboratory demonstrations. [22][23][24]

9.1. Technical Limitations

Because researchers frequently require precise delivery of minute amounts of liquids, syringe pumps have applications in the life sciences community. However researchers in academic labs typically report designs for air-driven syringe pumps, which do not provide comparable accuracy and are therefore not suitable for applications that require regulation of suction in addition to pressure [4]. Traditionally syringe pumps are expensive and bulky and rarely found in academic labs. Low-cost 3D printed syringe pressure pumps are fabricated and characterized for use in microfluidics applications. These pumps provide accurate, flexible, and low-cost tools for liquid handling in microfluidics applications.

13 cm x 13 cm x 7.5 cm enclosure was designed around the pump insert to strengthen the structure of the pump and organize the wiring and controls. The Display/Buttons PCB contains an LCD display with backlight, three push buttons, a piezo buzzer, and a 3.9 V to 5.1 V boost converter. The microcontroller PCB contains an ATMega 328 flashed with the Arduino bootloader. The controller includes a fuse to protect against the possibility of an external power supply being accidentally connected. The boost converter and piezo buzzer are connected through a transistor to allow software control of these components. The display and display button connections are made on a flat ribbon cable connector. To facilitate this a two-piece connector was 3D printed. The USB connection and programming button are accessible through holes in the top of the enclosure. The enclosure is designed to accept #2-56 threaded mounting screws. The hardware for the enclosure may be viewed in the attached bill of materials. Future work should journey from understanding the sequence of construction to understanding why machines are constructed to do operations as they were. As newfound knowledge of 3D

modeling software is applied, using the information acquired in both machine construction and 3D modeling should elicit a number of interesting thoughts, which may in turn lead to new knowledge.

9.2. User Acceptance

The user interface will consist of an LCD touchscreen interfaced with an Arduino microcontroller. Once powered on, the user will be prompted to enter a user name and verify the user name with a password. If this process is completed successfully, the user will be brought to the main screen where there are four options: “Start Infusion”, “View Infusions”, “Account Management”, and “Log Out”. Tapping one of these options will navigate the user through that process [6]. The user will be met with prompts that will aid them in completing the chosen option. Start infusion begins with the user entering desired infusion information. First, the user is prompted to select which motor will infuse the liquid. They will then be prompted to enter the infusion time. From here, the user will be given an option to input a slip interval of time in order to ensure the liquid stays in the syringe. If the user chooses not to enter this option, a default value will be used. The final step of the start infusion process is selecting the volume of liquid to infuse. Once all this information has been entered, the user will be brought back to the main menu. This option allows the user to create, edit, or delete a previously entered infusion. Once again, the user will go through the prompts that assisted them in filling out infusion information. Instead of being brought to the start infusion option, upon completion of this process the user will be brought to the main menu. Account management allows the user to add, edit, or delete other users. Managing other accounts includes the same prompts as creating a new account. Also upon successfully completing this, the user will be returned to the main menu. If at any point the user would like to log out, there is a log out button. Tapping this button will log out the original user and prompt them to enter a user name again. There are also two buttons on the main screen that display the current time. Tapping these buttons will bring up a clock for setting the desired time and date. The time will be displayed on the main mode screen and the current date will be displayed on a new “Date” option drawn on the main screen. [25][26][27]

9.3. Regulatory Issues

Pumps have been used in medicine for the administration of various materials. Syringe pumps have additional advantages over other types of pumps. Micro controllers can be used in conjunction with motors and other devices to provide drugs at desired rates. In addition to the control logic, the system can also be designed such that if an abnormal condition is detected; the system can shut itself off. Automating the processes of filling and locking the syringes is also possible. Syringe pumps are devices that are able to displace fluids from a syringe. Usually, a DC motor connected to a screw or needle arrangements moves the pump. In the hardware suggested, a stepper motor moves a syringe and controls the fluid flow rate. In addition to the control logic, the syringe infusion pump can provide a number of other functions. Security is important when dealing with medicines, drugs, or health information. This design includes a number of security features to prevent unauthorized access to the system and its stored information. In addition to the manual control functionality, alternate control methods could be added as part of the initial implementation. The most straightforward method would be a simple IR remote control.

Regulatory issues for this device will be minimal due to its potential certification as a Class I or II device. This will allow the design to be finalized and added to the production line without extensive redesign either mechanically or within the control circuit. The most likely regulatory agency will be the FDA, as this design in a clinical setting would fall under their Authority. In addition to a common understanding of management procedures, the FDA also places high levels of responsibility on manufacturers to ensure that devices meet their specifications. There will be a burden of documentation for design verification and test results, as well as ensuring the quality control of off-shelf components, assembly methods, and test procedures [6]. Off-shelf components will need to feature a clear history of quality control procedures. If suspected components cannot be traced, the entire device may need to be tested by hand. Any changes made after approval will also require extensive redesign. Monitoring systems can be created to ensure that devices meet safety standards at the lowest possible price. The design files for the 3D printed components will be released to the public along with assembly instructions, component lists, and circuit diagrams [4]. This will allow other researchers to use this pump as a cheap alternative, as it can be made with off-the-shelf components.

10. Future Directions

In the future, the electronic components should be placed on a printed circuit board (PCB) that can fit inside the shell and provide a consistent ground connection. A smaller and cheaper microcontroller should also be acquired to avoid problems in soldering wires and shrinking the overall volume of the electronic components [18]. The screw mechanism that holds the syringes need to be re-evaluated, as it is 3D printed with plastic and has to be carefully maintained. Coatings made out of metal would greatly improve long-term stability and function of these parts. Furthermore, a different version of the device shell should be modeled and manufactured. Most notably, a larger hand grip along the left side of the device should be modeled in order to produce a more ergonomic grip and improve usability.

The proof-of-concept design for a syringe pump capable of automated and accurate Botox injections has been developed. By controlling the flow rate and volume of injection into a phantom face model, the accuracy of the final injection volume was tested and shown to fall within the acceptable margin of error. Additionally, all necessary components were 3D printed and assembled into an aesthetically acceptable product to allow use and testing in a larger laboratory setting. Additionally, if it were possible to partner with a company in the fields of biomedical engineering or integrative systems, a more systemic approach could be taken to defining design parameters, such as budgeting the entirety of a plastic injection mold process. After initial designs had been created, feasibility tests of a working design could be quickly assessed and prototypes could be produced in larger quantities.

Almost all design functions that allow for prototyping and building industry-grade mechanical components must be mastered to go into a career in engineering. Classwork focused on design principles, CAD programs, rapid prototyping functionalities, and Finite Element Analysis pathways provided the basis for many of the concepts utilized in the final product. In conjunction with more specific courses on soldering circuits, 3D printing methods, programming languages, and practical engineering constraints, personal technical skills applicable to an engineering career are more honed than when entering the Capstone course.

10.1. Integration with IoT

In the current fast-paced, busy world, it is difficult for medical staff to keep track of every patient accurately and consistently. Keeping track of the total administrations of medications is the same. A doctor needs to keep a prescription of how many medications should be injected into which patient and how many milometers of that should be given to them, otherwise there is a chance that the patients can either take too much dosage or not have enough [28].

Overdoses can lead to serious health problems or even death, while not getting enough can leave the disease to progress. This way, a doctor will keep themselves in charge so that in case of emergency, they can easily check that data for every patient and be alerted if one of them is in danger. For these reasons, this infusion pump system is presented to take charge of infusing the prescribed medications into patients and give the doctor access to their health data. This system will consist of both hardware and software systems. The hardware side will control the motors and will need to take care of tracking the medications that are infused into patients (syringe control). It also needs a real-time clock to know the current time. Everything will be controlled under a microcontroller that has pins for all the above tasks and will need to be coded to function accordingly.

The software side will connect to the application programming interface of the code and store the health and medication data into the database. The software will also have to occupancy antialias filter to reduce noise before the data is passed to the computer (data reading). This result in communication from the Arduino to the server and back via this API. Lastly, this setup will allow the doctor to be notified about possible health risks. It will crawl the entire database every 30s and if finds a patient in danger, it raises a warning so that the doctor investigating this patient further.

10.2. Advancements in Sensor Technology

The standard syringe pump is an archaic design that consists of three standard 30 ml syringes, pneumatic valves, a relay board, and a microcontroller for control. The precise motion of the syringes is employed to modulate flow in PDMS microfluidic chips. The enclosed syringes prevent evaporation and minimize the amount of essential oil needed, maintaining a high quality of perfusion even after long-term experiments. The dual-syringe design enables continuous infusion by alternating the leading and trailing syringes on a time schedule of several days. The control of the pumps is handled by a standard Arduino microcontroller, which allows a simple interface and real-time control. The architecture permits an easy scaling of the pump system, with an unlimited number of additional syringes available under $200.

Syringes are inexpensive, disposable, and readily available. The custom-made syringe holders are designed with the 3D design software, and manufactured using a 3D printer. The syringe holder uses integrated hinges that can be manually opened and closed to secure the syringe. This doubles the speed at which the syringes can be exchanged, and there are no fragile components that can easily fall off like in previous designs [4]. The simple switch plate is designed to be user-friendly while still capable of complex commands. A joystick box and additional buttons can be used to manually control the pneumatic valves and the pumps, if desired. Using a remote desktop simulation program or control over a connection, real-time adjustments can be made during experiments.

10.3. Potential for Automation

Infusion pumps are widely used in healthcare institutions to guarantee the accurate and timely delivery of fluids necessary for hemodynamic stabilization in critically ill patients. Traditional infusion pumps provide time-controlled delivery. They allow clinicians to enter the total duration of infusion and the rate of infusion in mL/hr. Temporary intervention on the infusion pump may lead to potentially dangerous maneuvers if no supporting data are made available to the caregiver. Automated feedback-controlled doses infusion pumps are needed to provide clinicians with an additional post-administration automatic correction of the infusion rate. Invasive arterial blood pressure monitoring devices have proven to be effective to actively track hemodynamics in real-time. Nevertheless, hardly any infusion pumps can automatically adjust their volume delivery to the continuously updated set points. The present invention relates to an infusion pump designed to simplify mounting, which has been purposely designed for medical applications where sterile conditions must be guaranteed [6].

The inventive infusion pump comprises: a syringe support unit designed for releasably mounting a syringe the plunger of which is configured to move longitudinally of the syringe during a pump stroke cycle. A motor drive unit configured to actuate the longitudinal movement of the plunger during a pump stroke cycle. Programmable control and display unit. The syringe support unit and the motor drive unit are arranged at least partly in a common casing forming an integrated assembly. At least one of the syrup support and pump drive units on its side opposite of the common casing defines a connection end snack to cooperatively engage with mounting connector [18]. A method for simplifying mounting of an infusion pump where the infusion pump is done and where the peelably adherent sterile enclosure is provided. The peelably adherent sterile enclosure is peelably adhered to the infusion pump or a part thereof and is thus sealed as a whole to form a sterile enclosure between it and the infusion pump or the part thereof, which sealed sterile enclosure is then removed along with the peelably adherent sterile enclosure prior to starting operation of the infusion pump.

11. Conclusion

The Arduino Microcontroller based Syringe Infusion Pump has been designed and developed on open-source platforms allowing easy access to open-source codes. The use of low-cost components along with 3D printed parts has led to a low-cost design which is capable of delivering the desired results in an efficient manner. Furthermore, the stepper motor pulses can be put into the User Interface code in order to further customize functionality. The pulse width, or duration of the “high” state on the pulse input, may be modified in order to adjust the duty cycle from low to high or high to low so that the driven motor rotation may occur more quickly or more slowly. Without modifying the delay timings, code min/max rates should safely ensure functionality at +/-5% from default performance.

Future work should include assembling the current design into a more traditional enclosure, as well as potentially redesigning certain components. Enclosure parts would vary in size depending on how many pumps would be enclosed. The enclosure would include a slot for a 100mm x 100mm non-conductive sheet to separate enclosure portion for circuitry and enclosure portion for plumbing/infusion. The circuit board/casing portions would also enclose the motors and machinery in order to remove them from exposure to splashing fluid. The enclosure should also include carpet/non-slip pads in order to reduce vibrations through the workspace.

Some considerations for future alterations include design consideration for passive valve envy, also known as on-chip valving. The capability would allow increased numbers of reservoir channels, and the respective circuitry to actuate them. Another exciting potential design alteration would be creating a new enclosure to house a rotatable multi-channel piezoactuator with moveable adhesive tape slides to accommodate the common pressures and speeds currently utilized in a multitude of lab-on-a-chip applications.

References:

1. B. Mallick and H. C. Mohanta, "Development of a Syringe Infusion Pump," Journal of Control System and its Recent Advances. researchgate.net
2. Z. Tariq, Z. Anjum, L. Memon, G. Rasool et al., "Fabrication of Smart Syringe Infusion Device: A Solution for Healthcare Industry," researchgate.net, . researchgate.net
3. Y. Wu, Y. Chen, and Y. Cheng, "Building an Arduino-Based Open-Source Programmable Multichannel Syringe Pump: A Useful Tool for Fluid Delivery in Microfluidics and Flow Chemistry," Journal of Chemical Education, 2024. [HTML]
4. J. R. Lake, K. C. Heyde, and W. C. Ruder, "Low-cost feedback-controlled syringe pressure pumps for microfluidics applications," 2017. ncbi.nlm.nih.gov
5. E. Chappel, C. Conan, S. Mefti, F. Cannehan et al., "Infusion Micro-Pump Development Using MEMS Technology," 2012. [PDF]
6. D. L III Keister, C. W Borntreger, E. M Oglesbee, and Z. M Stevens, "Automated Syringe Filler," 2016. [PDF]
7. R. Ze Gao, M. Hébert, J. Huissoon, and C. L. Ren, "µPump: An open-source pressure pump for precision fluid handling in microfluidics," 2020. ncbi.nlm.nih.gov
8. E. H. Aboadla, S. Khan, K. Abdul Kadir, and Z. Md Yusof, "Suppressing voltage spikes of MOSFET in H-bridge inverter circuit," Electronics, vol. 10, no. 6, 2021. mdpi.com

9. L. Liu, "Design of H-bridge circuit for CMOS stimulator and its application analysis," in *Proceedings of the International Conference on Image Processing and Computer Applications*, 2023. [HTML]

10. A. Adam, F. Kacar, and N. Mastorakis, "Enhancement design of eleven-level cascaded h-bridge motor driver application," Computers and Electrical Engineering, 2025. [HTML]

11. M. Rafiquzzaman, "Microcontroller Theory and Applications with the PIC18F," 2025. [HTML]

12. T. T. Ogunseye, G. B. Egbeyale, A. K. Bello, "Design and Construction of a Microcontroller Based Electronic Moving Message Display," arXiv preprint arXiv:XXXX.XXXX, 2021. [PDF]

13. K. Udeze, "Design of Microcontroller Based Security Door/Rain Detector Using Memory Card," Journal of Applied and..., vol. 2025, Mediterranean Publications, 2025. mediterraneanpublications.com

14. E. Mbunge, B. Muchemwa, S. Jiyane, and J. Batani, "Sensors and healthcare 5.0: transformative shift in virtual care through emerging digital health technologies," global health journal, 2021. sciencedirect.com

15. J. G. Duan, T. Y. Ma, Q. L. Zhang, Z. Liu, and J. Y. Qin, "Design and application of digital twin system for the blade-rotor test rig," Journal of Intelligent …, vol. 2023, Springer. [HTML]

16. J. Leng, M. Zhou, Y. Xiao, H. Zhang, Q. Liu, and W. Shen, "Digital twins-based remote semi-physical commissioning of flow-type smart manufacturing systems," *Journal of Cleaner Production*, vol. 2021, Elsevier. nih.gov

17. H. Lee, J. A. Lim, and H. K. Nam, "Effect of a digital literacy program on older adults' digital social behavior: A quasi-experimental study," *Journal of Environmental Research and Public Health*, vol. 2022. mdpi.com

18. A. Sezer, C. Marie Montesi, L. Taylor Gallagher, and T. James Mott, "Clinical Pipette," 2015. [PDF]

19. S. Tarapiah, "Maze Solver Robot," 2024. najah.edu

20. K. A. R. Youssefi and M. Rouhani, "Swarm intelligence based robotic search in unknown maze-like environments," Expert Systems with Applications, 2021. [HTML]

21. S. Alamri, S. Alshehri, W. Alshehri, H. Alamri, "Autonomous maze solving robotics: Algorithms and systems," in *Journal of Robotics*, 2021. researchgate.net

22. D. Loukatos, M. Kondoyanni, G. Alexopoulos, "On-device intelligence for malfunction detection of water pump equipment in agricultural premises: feasibility and experimentation," Sensors, vol. 2023. mdpi.com

23. T. Lupinski, M. Ludwig, S. Fraden, and N. Tompkins, "An Arduino-based constant pressure fluid pump," The European Physical Journal, vol. 2021, Springer. springer.com

24. D. Loukatos, I. Glykos, and K. G. Arvanitis, "Communicating the Automatic Control Principles in Smart Agriculture Education: The Interactive Water Pump Example," Robotics, 2025. mdpi.com

25. C. Ünsalan, H. D. Gürhan, M. E. Yücel, and C. Ünsalan, "LCD, Touch Screen, and Graphical User Interface Formation," in *Microcontrollers*, 2022, Springer. [HTML]

26. A. Goeritno, I. Setyawibawa, and D. Suhartono, "Designing a microcontroller-based half-duplex interface device drove by the touch-tone signal," Jurnal Infotel, 2021. ittelkom-pwt.ac.id

27. O. I. Adebisi, A. B. Ogundare, T. C. Erinosho, "Development of a microcontroller and resistive touchscreen-based speed monitoring and control system for DC motor," Int. J. Adv. Appl., 2023. semanticscholar.org

28. Z. A. Al-Odat, S. K. Srinivasan, E. M. Al-Qtiemat, and S. Shuja, "A Reliable IoT-Based Embedded Health Care System for Diabetic Patients," 2019. [PDF]