Sentinam Innovations

School of Engineering Science
Simon Fraser University
Burnaby, British Columbia

November 9, 2015

Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6

Re: ENSC 440 Design Specifications - Remote Monitored IV for Home Care

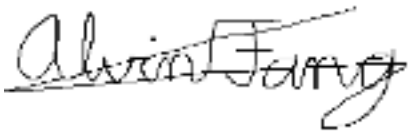
Dear Dr. Rawicz,

Please find attached our design specifications for Sentipump, a Remote Monitored IV for Home Care, which provides details of our technical design guidelines for our ENSC 440W Capstone Engineering Science Project. Sentipump is an infusion pump that can be remotely controlled through mobile applications to safely administer IV therapy in a home environment, while being affordable.

This document describes design specifications that apply to Sentipump. In this phase of development, only the prototype version of Sentipump will be implemented. We discuss the design for features included in future versions of our project, but they will not be implemented in this development phase. The design for the overall system, down to individual components, will be explained in this document. A detailed test plan is also included.

We look forward to your review and feedback of our project. If you wish to contact me, please reach me at kpfang@sfu.ca. Thank you for your time and consideration.

Sincerely,



Alvin Fang
Chief Operational Officer
Sentinam Innovations

Enclosed: *Design Specifications for Remote Monitored IV for Home Care*

School of Engineering Science
Simon Fraser University
Burnaby, British Columbia

Sentinam Innovations

Design Specification for a Remote Monitored IV for Home Care

Project Team: Alvin Fang
Chelsey Currie
Pranav Malik
Baljinder Singh
Nyann Moe

Contact Person: Alvin Fang
kpfang@sfu.ca

Submitted to: Andrew Rawicz – ENSC 440
Steve Whitemore – ENSC 305
School of Engineering Science
Simon Fraser University

Issue Date: November 12, 2015
Revision: 1.1

Table of Contents

List of Figures

List of Tables

Glossary

GUI	Abbreviation for Graphic User Interface.
Infusion Pump	A medical device that delivers fluids, such as nutrients and medications into a patient's body in controlled amounts.
IV	Abbreviation for intravenous, is the infusion of liquid substances directly into a vein.
FDA	Food and Drug Act.
SDK	Abbreviation for software development kit.
LCD	Liquid Crystal Display.
EM	Electro-Magnetic.
EMI	Electro-Magnetic Interference.
SNR	Signal to Noise Ratio.
PCs	Polycarbonates.
BPA	Bisphenol A
ABS	Acrylonitrile Butadiene Styrene

Executive Summary

This document provides details of the design specifications for the Sentipump, an infusion pump with remote monitoring capabilities, and support for sensor attachments and home care usage. In this document, the design and implementation details for the prototype will be discussed in detail. We will also discuss the design improvements we plan to include in the final versions of Sentipump. We implement our system with the requirements specified in our functional specification document in mind, and provide detailed comparison and justification of our design decisions. Prototype features are marked by I and II in our functional specifications document, while the final version features are marked by II and III.

In our document, we explain the purpose of our system, and discuss the overall system design. We then break down our overall system design into key sections for more in- depth analysis of the design, and the implementation details of each component: the IV pump, flow rate sensor, control unit, user interface unit, and the mobile application design.

A test plan is also provided in this document, which our team will use to ensure the quality of Sentipump. In our test plan, we describe unit and integration tests we will use to ensure our system design meets our high functional standards and expectations. Tests will be performed most rigorously on the highest priority features, such as the accuracy of the infusion rate our system delivers. After the main features are tested, we will then test the side features, such as additional sensor attachments and WiFi connection to mobile/desktop applications. As features are modified, connected features will have to be retested as well to ensure new changes do not break other components.

Our team has done thorough research for this project, and we believe we have come up with a solid product design. Safety and reliability are the two important features we incorporate into our product design and we are aware of the safety standards required to manufacture a medical device. Our team proposes to design and build the IV

system and mobile app with great reliability and effective testing. The finished product will improve on features of current IV systems, have an affordable price, and include remote monitoring capabilities.

1. Introduction

The Sentipump is an intravenous infusion pump designed to infuse medications at a set rate, and is targeted for at home use. The integrated mobile app allows the infusion to be supervised and modified remotely by healthcare practitioners or caregivers. The Sentipump will have a closed loop flow control, which will allow the infusion rate to be set and maintained precisely. We endeavour to make an infusion pump which allows patients with chronic conditions to receive their intravenous medications at home safely, and is easy to set and monitor for caregivers with no medical background. This design specification includes the technical details of all the components of the Sentipump.

1.1 Scope

This design specification covers the design of the Sentipump, and explains how the functional specifications laid out in the Functional Specification will be met [1]. This document focuses on the requirements for a prototype and as such, only the functional requirements labelled I or II are discussed.

1.2 Intended Audience

Members of Sentinam innovations will use this document to ensure that all design requirements are followed for the prototype. Test engineers will use the test plan included in this document to test the Sentipump and ensure that it conforms to all requirements.

2. System Specifications

The Sentipump will control the flow of medication at a preset rate. It will be highly accurate and will maintain its setting automatically through closed loop feedback with a flowmeter. The settings on the Sentipump will be set through the user interface on the device, or remotely via the integrated mobile app interface.

3. System Overview

The flow process chart shown below in *Figure 1* illustrates how these directives would work in our system.

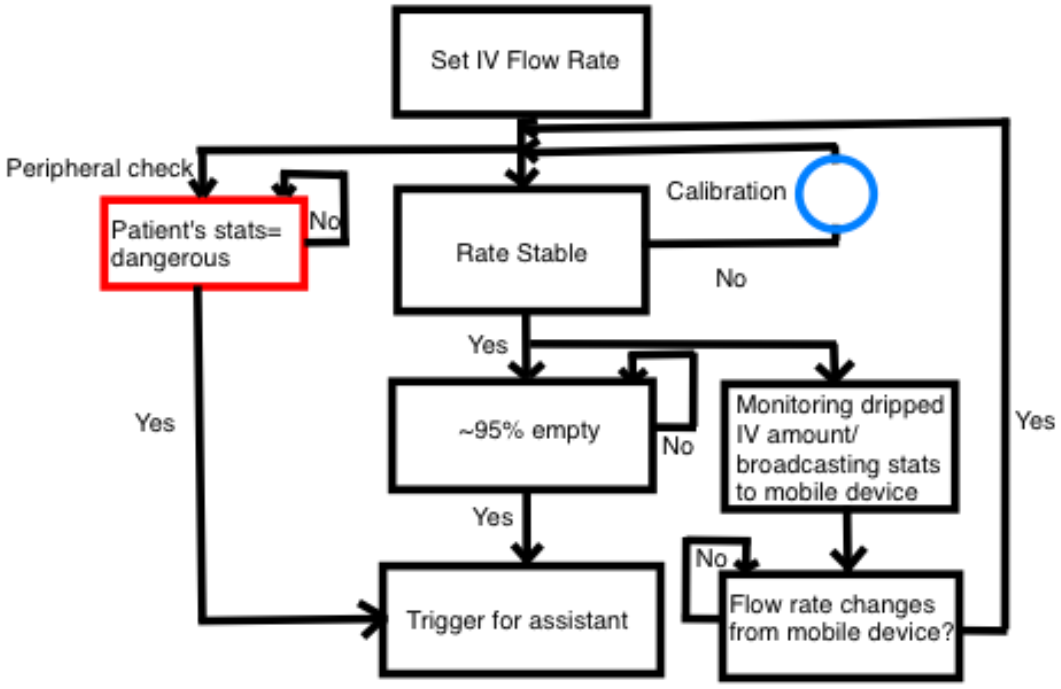


Figure 1: Process flow chart.

In the system overview model of Sentipump, shown below in *Figure 2*, the roller clamp is modified to include a peristaltic pump and flow rate sensor. The precise rate can be set by simply entering the desired rate on a mobile device or adjusting the potentiometer on the control unit. Once the flow rate is set, the peristaltic pump will

adjust the flow rate slowly, until it reaches the preset value detected by the sensor from the tube. The system will also ask the user to input the volume of the IV bag in order to generate a completion time at that flow rate. For the purpose of clarity, the drip rate displayed on the system will be in drops per minute (gtt/min), which is the current standard unit for IV drips.

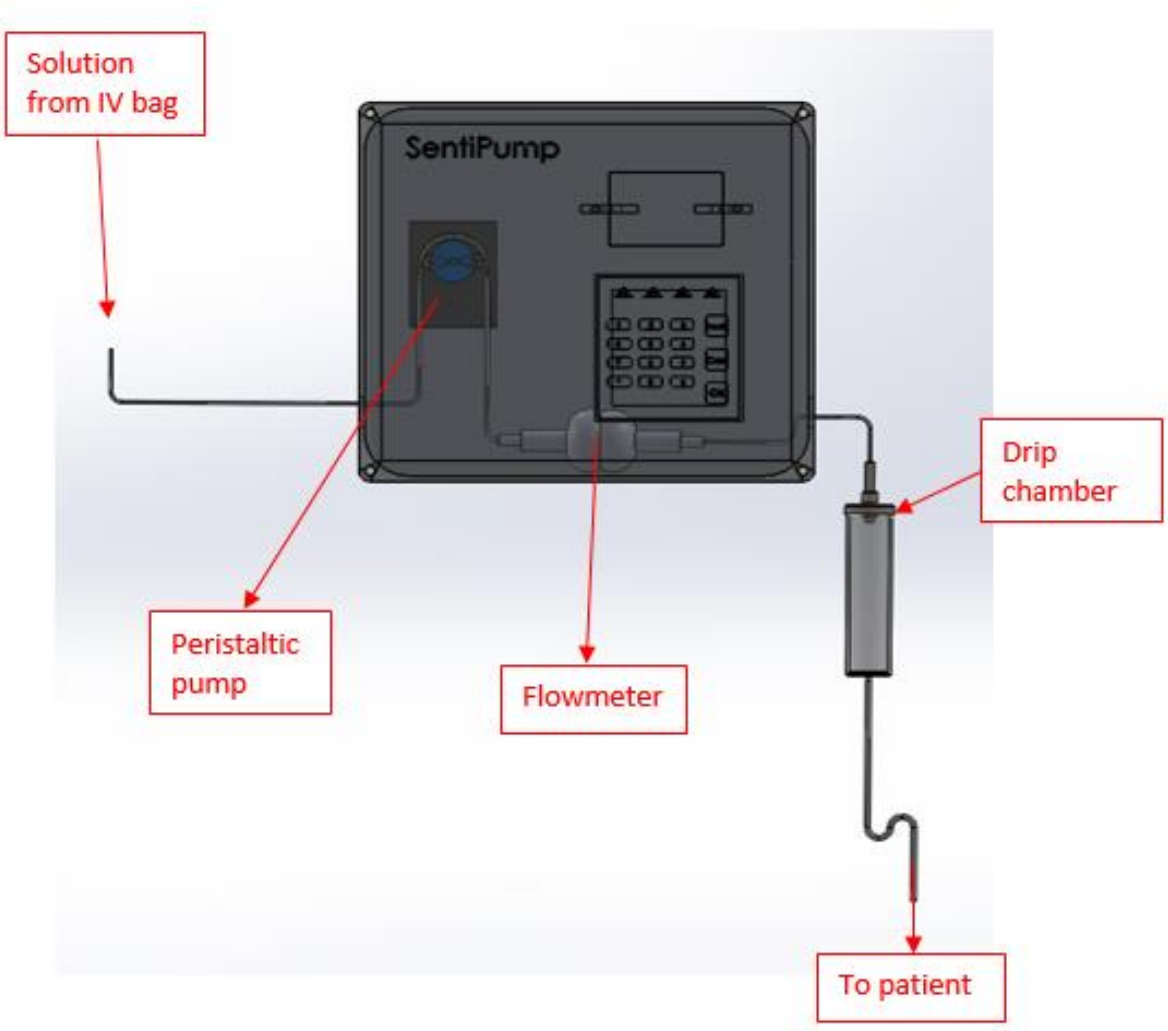


Figure 2: Overview of Sentipump Design.

3.1 Mechanical Design

These phases are the primary system overview and more applications will be added to manually calculate the flow rate. The foundation of our system is a simple closed loop feedback system. As shown below in Figure 3, the IV tubing enters the system, is constricted by the system to change the rate of flow, and the flowmeter then provides feedback on the accuracy of the setting.

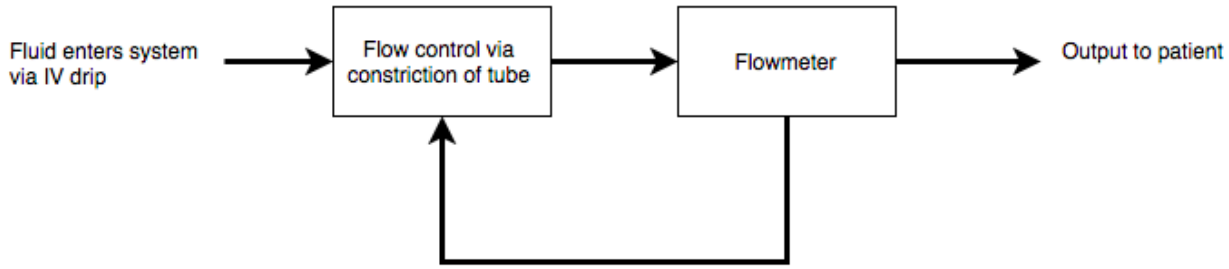


Figure 3: Simplified overview of the infusion pump system.

The only mechanical component in this device is the peristaltic pump. This pump controls the flow of the fluid by ways of compression and relaxation of the tube. Its operation is very similar to our throat and intestines [2]. There are some reasons for choosing this pump over other available pumps. The rollers in the pump compress the tube as it moves thus creating a vacuum and driving the fluid forward. This pumping action makes it suitable for applications requiring low volume and accurate dosage, much like our device. Also, the fluid being pumped does not come in contact with any of the parts of the pump, thus making it free of contamination, and easy to clean and sterilize. [2]

Another consideration while selecting the peristaltic pump is the number of rollers attached to the rotor for pumping the fluid. Higher number of rollers (3 or 4) means smaller amount of dosing, but it increases the stress on the tube. This will decrease the life of the tubing inside the pump. Keeping this in mind, our team decided to use the

pump with two rollers. A two roller design will allow effective use of the tube life cycle. Smaller amounts of dosage can be achieved by decreasing the rpm of the pump slightly more than required.

3.2 High level System Design

As seen in *Figure 4* below, analog sensors, digital sensors, the user interface (LCD and keypad), remote control through mobile devices, and current measurement are the system inputs. The inputs will be conditioned later using filters, amplifiers, Analog/Digital converters, USB shield, debouncing software and encoders before being processed by the Arduino Uno control unit.

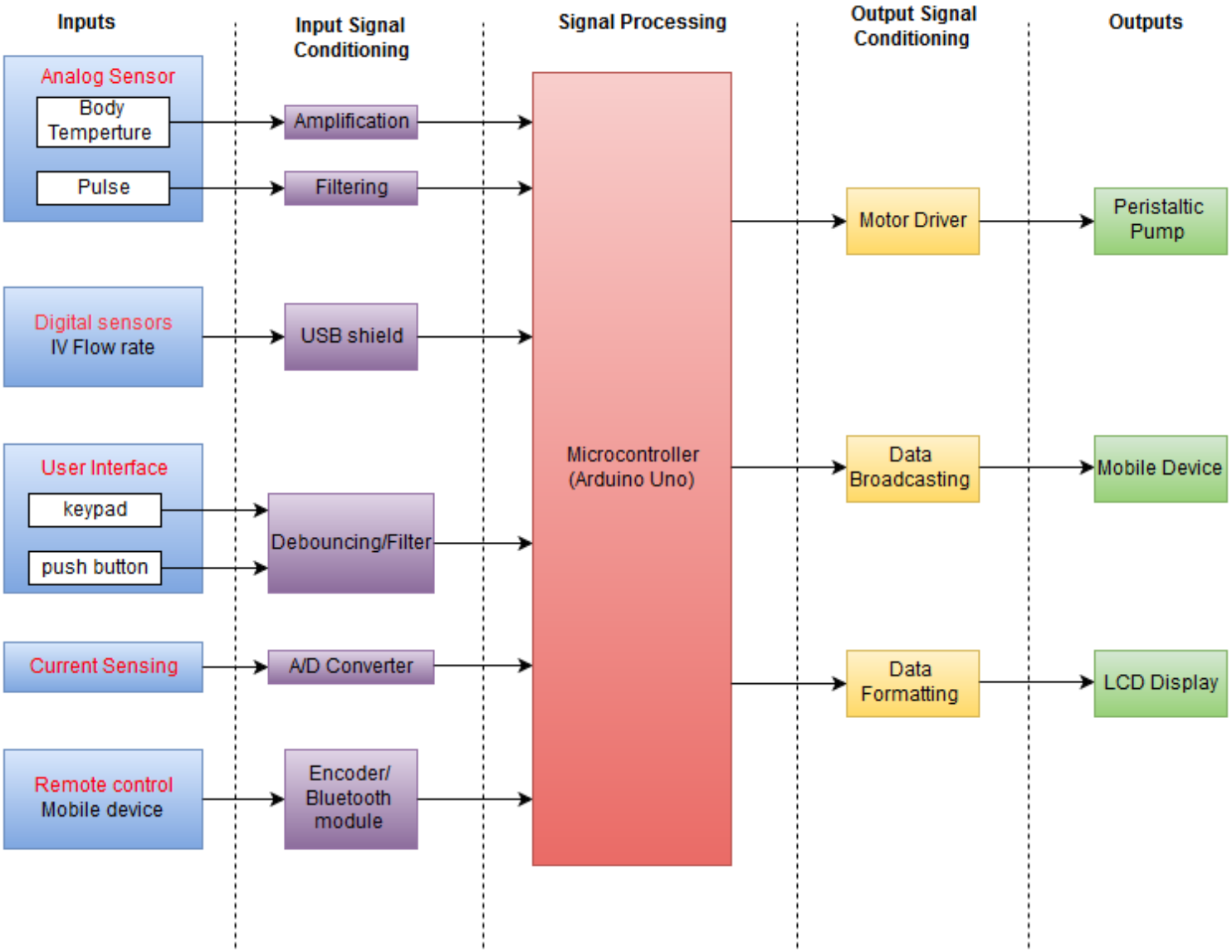


Figure 4: System block diagram.

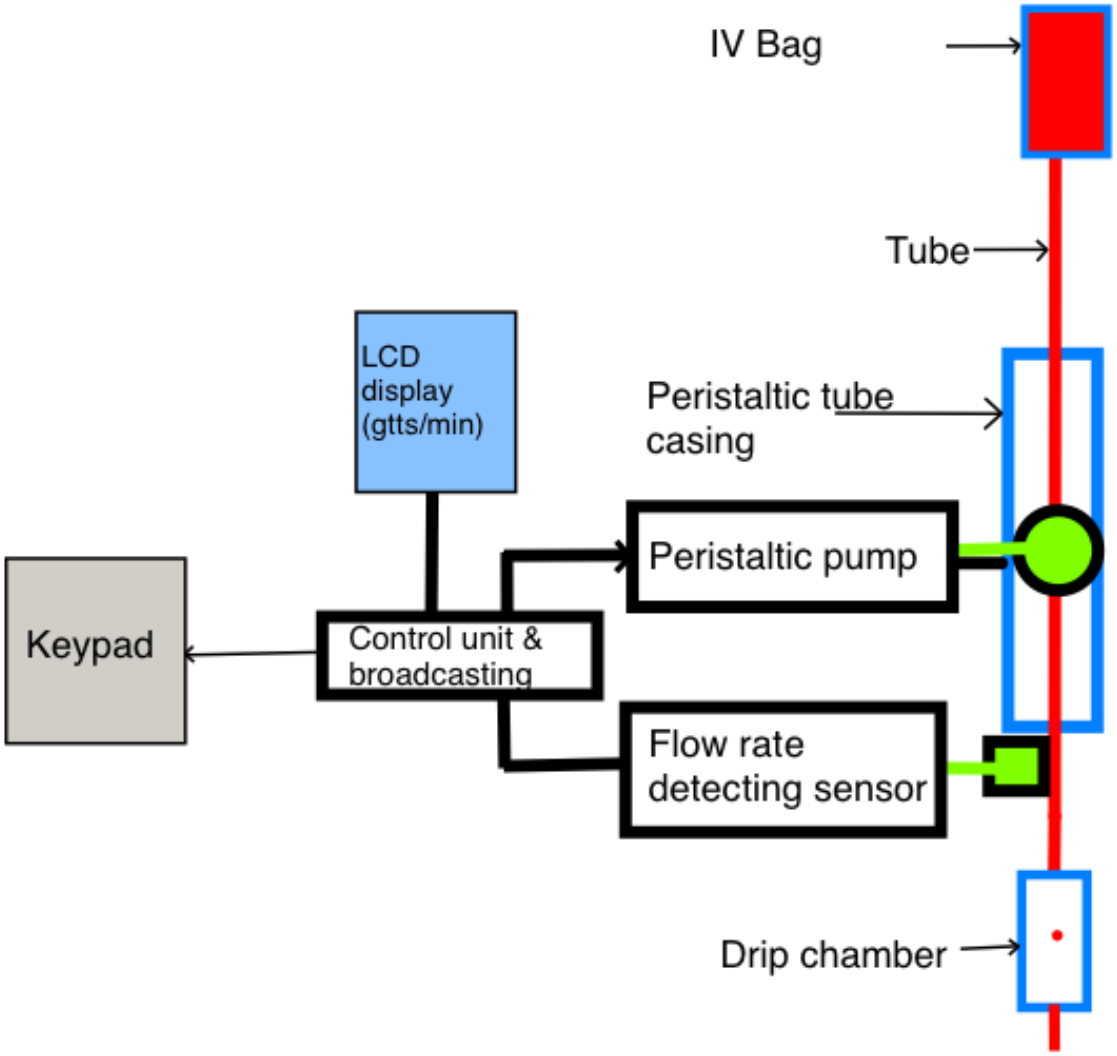


Figure 5: System overview of the Sentipump.

Due to the low output voltage of the temperature sensor LM35, an amplifier will be used to achieve a more precise sensor reading as well as smoother signal processing. For the pulse sensor, a filter will be used to avoid unwanted noise for the pulse reading. Additionally, a debouncing technique will be used to avoid missed readings from the user interfaces (push button and keypad). For the remote control, a

mobile device will be used to send data to the controller and an encoder will be used to translate the mobile data into data that is compatible with the Arduino.

The system outputs consist of an LCD, a keypad and a mobile device for remote control. These outputs will be conditioned using a motor driver, data formatting and a USB shield. Since the peristaltic pump's maximum current rating is approximately 80 mA and the working voltage range is up to 12V, the Arduino motor shield will be used to accommodate the pump power requirements. The system output data will be displayed on an LCD and broadcast to a mobile device. To acquire proper display, the output data will be formatted into LCD compatible data. To broadcast data, a USB module will be used to communicate with a mobile device.

3.3 Sensor Placement

There are three sensors implemented in this remote controlled IV infusion pump system. For the detection of flow rate, a drop counter built by us will be used in the prototype to detect flow. This sensor will connect to the IV drip chamber, in order to detect the drips.

For sensing the body temperature, an LM35 sensor made by Texas Instruments will be used. It needs to be attached to the patient's fingertip continuously throughout the duration of the IV treatment. It will give a precise temperature measurement and will also be comfortable for the patient. The system will be using a pulse oximeter for heart rate monitoring, which will also attach to the patient's fingertip. The overview of the sensor system is shown below in *Figure 6.*

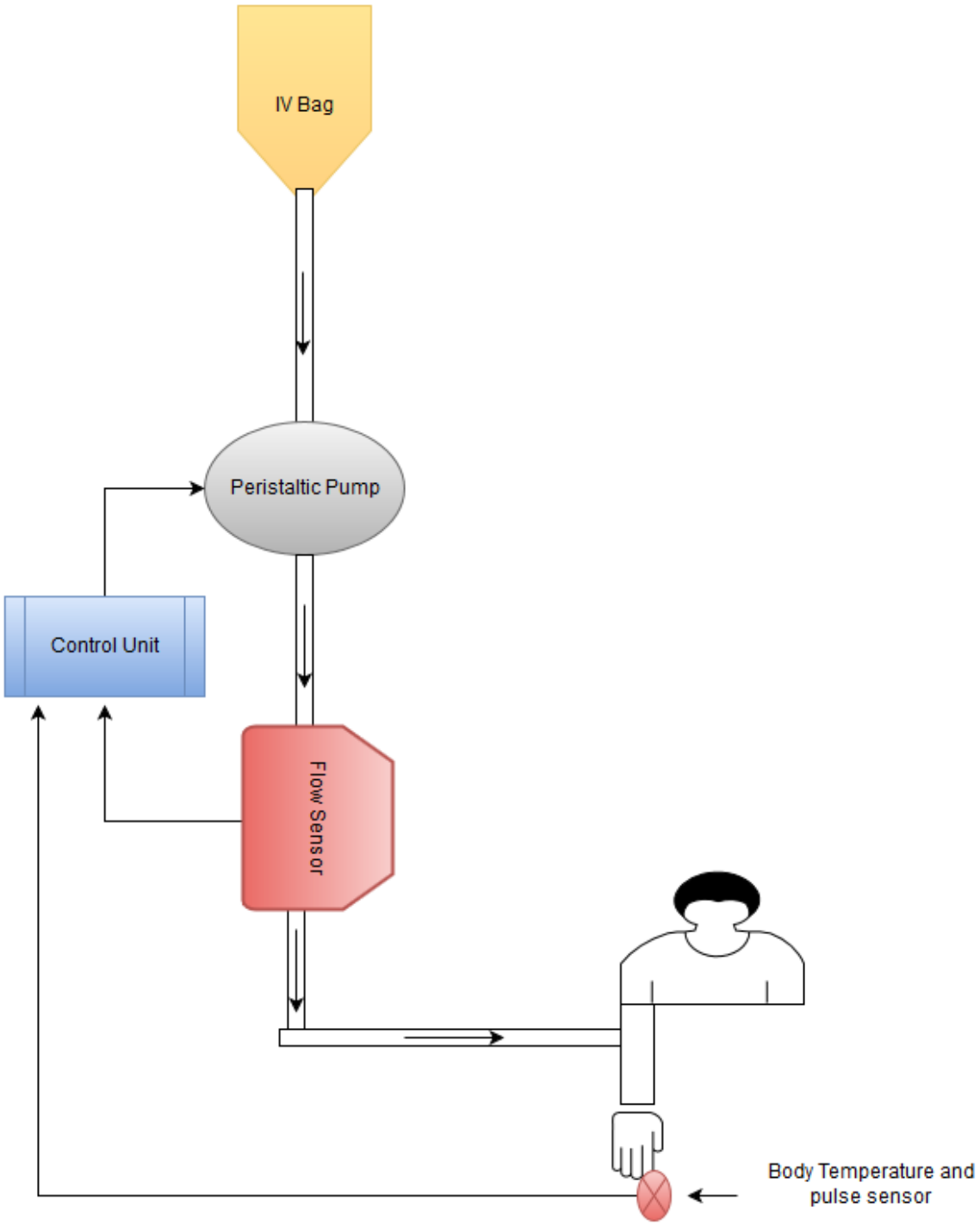


Figure 6: Overview of sensor placement.

3.4 Pump Placement

The pump shall be placed directly after the IV bag, before the drip chamber as a safety precaution in case the pump introduces any air to the tubing system. This would allow the bubbles to vent through the drip chamber, which is designed to prevent air bubbles from traveling through the tubing [3]. The pump should also be placed before any connections to other tubes, to allow each fluid to be pumped individually.

3.5 Electrical System

Most of the components of our device will run on a 5V power supply except the peristaltic pump and the motor shield, which require a 12V power supply. Thus, we need to have two power supply voltages, 5V and 12V.

The motor shield requires a 12V power supply since the motor shield will be operating the peristaltic pump that requires 12V. The motor shield requires a separate power supply. This is a precautionary measure recommended by the manufacturer of the motor shield to avoid damage to the main board [4]. Thus, we need a separate ground plane for the pump and the shield. The connections for the system are shown below in *Figure 7.*

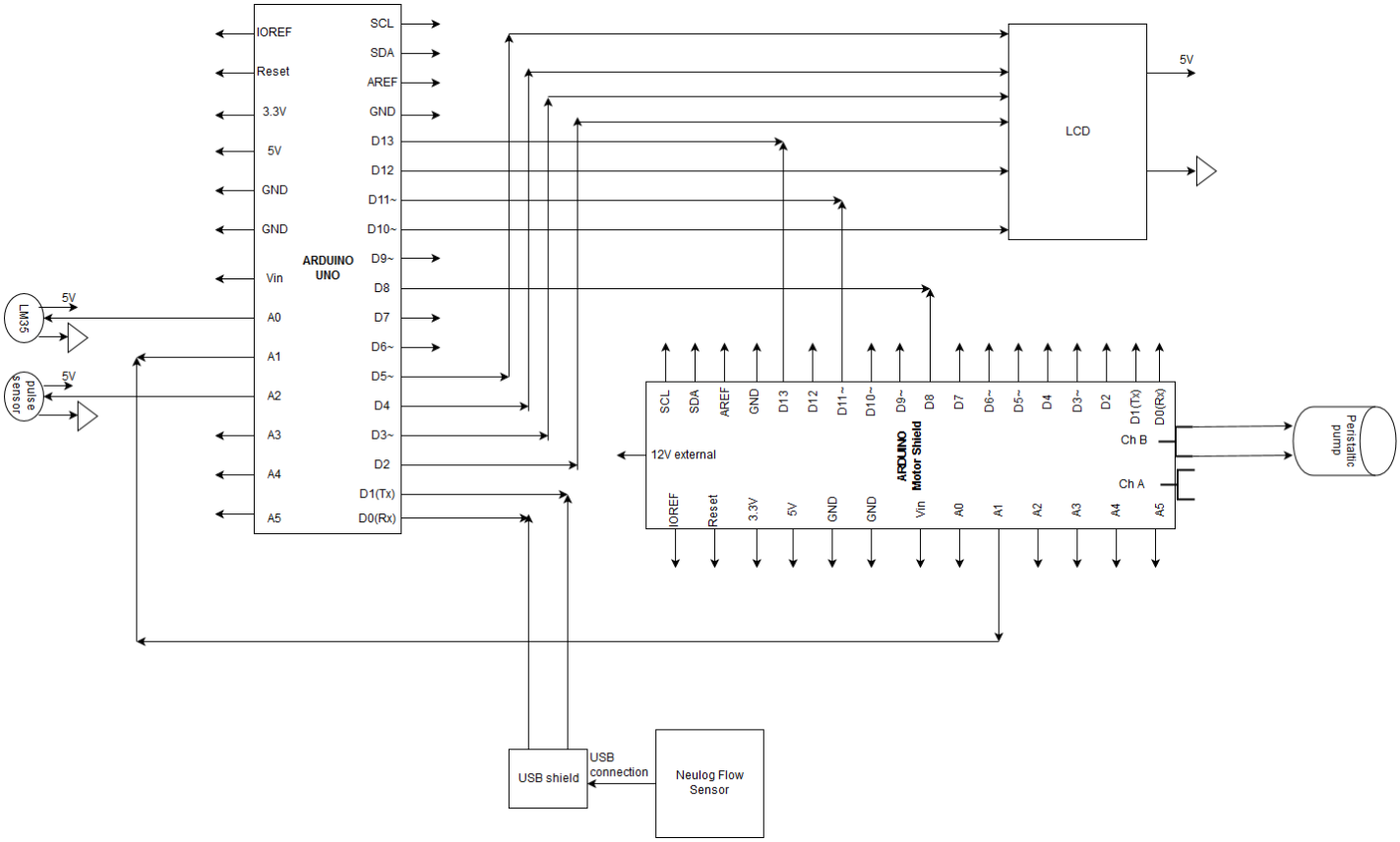


Figure 7: Pin diagram of the system.

3.5.1 Noise Considerations

Since our device is intended for home use, considerable amount of care will be taken to avoid interference to and from other medical devices and electronic equipment, which are a source of noise.

Electromagnetic interference (EMI) will be handled by the use of proper shielding materials. Shielding materials will either reflect or absorb the EM waves generated both by the outside environment and also the internal circuitry of the device. We will not be able to show this in our prototype, but the production version will have the EMI shielding. Even power cables can cause interference with other devices. The cables will either have power filters or use shielded cables that end at the shielding connector.

Even with shielding, one might encounter some disturbances. These can be attenuated using high signal-to-noise ratio (SNR) and avoiding the use of long connector wires. [5]

3.5.2 Safety Considerations

Electrical safety is an important step to prevent damage or harm to the user or to the components of the device. The following precautions are undertaken for electrical safety:

- Proper grounding and insulation of the wires
- Using insulated material for the housing (discussed in the casing section below)
- Current limiting fuses and circuit components in areas of high current
- Proper ventilation for the heat generated by the battery and other electronic components

3.6 Power Supply

The total current required for the device is calculated by summing the currents required for individual components. The power requirements for the components are given below in *Table 1*.

Table 1: Power requirements for prototype components.

Component	Current	Operating Voltage
Peristaltic Pump	80mA [6]	12V
Drip counter	10mA	5V
Microcontroller (Arduino Uno)	50mA [7]	5 - 12V
Motor shield	Same as pump	5 - 12V
Bluetooth module	40mA [8]	3.3 - 5V
LCD Display	20mA[9]	5V
Keypad	30mA [10]	5V
Total	230mA	5-12V

The display, the keypad, and the Bluetooth module will be powered by using the 5V output pins of the microcontroller board.

For implementation of the prototype, we will be using the power supply from the wall outlet. But for the production version, we will also include a back-up battery in case of power failure or if the user wants to move around. The battery should last for at least 4 hours of continuous run time. Thus, a 12V 24000mAh Li-ion rechargeable battery will be used.

3.7 Casing

The outer case for the pump needs to be rigid, impact resistant and able to withstand changes in temperature. Keeping these factors in mind, we use the most

common medical device materials like Polycarbonates (PCs). PCs contain BPA (bisphenol A), which is a health hazard. ABS (Acrylonitrile Butadiene Styrene) was considered for the housing but we decided to go with PBT (Polybutylene terephthalate) due to its higher impact resistance and better insulation properties. [11]

The electronic components, mainly the PCBs, need proper shielding from EMI (Electromagnetic interference). Any metal sheet will work for this. Also, the peristaltic pump needs some shielding as well due to generation of a magnetic field. The thickness of the metal sheet required can be calculated using the equations for skin depth for EM radiations. This is not done for the prototype version so we will not show the equations here.

4. IV Pump

The Sentipump system will use a peristaltic pump as its actuator. A peristaltic pump is a positive displacement pump that works via peristalsis, which is the alternating radially symmetrical compression and relaxation of the tubing [12]. This compression is obtained by compressing tubing with a rotating roller. The liquid inside the tubing stays uncontaminated, since it does not exit the tubing. At its foundation, a peristaltic pump is a DC stepper motor with a rotor attachment and casing for tubing. A DC stepper motor works very simply; for each electrical pulse, the shaft of the motor moves one step [13] As such, the pump can rotate a single step, and therefore pump a very small volume of liquid per minute, making it an appropriate choice for the Sentipump. A rotary peristaltic pump is shown below in *Figure 8.*

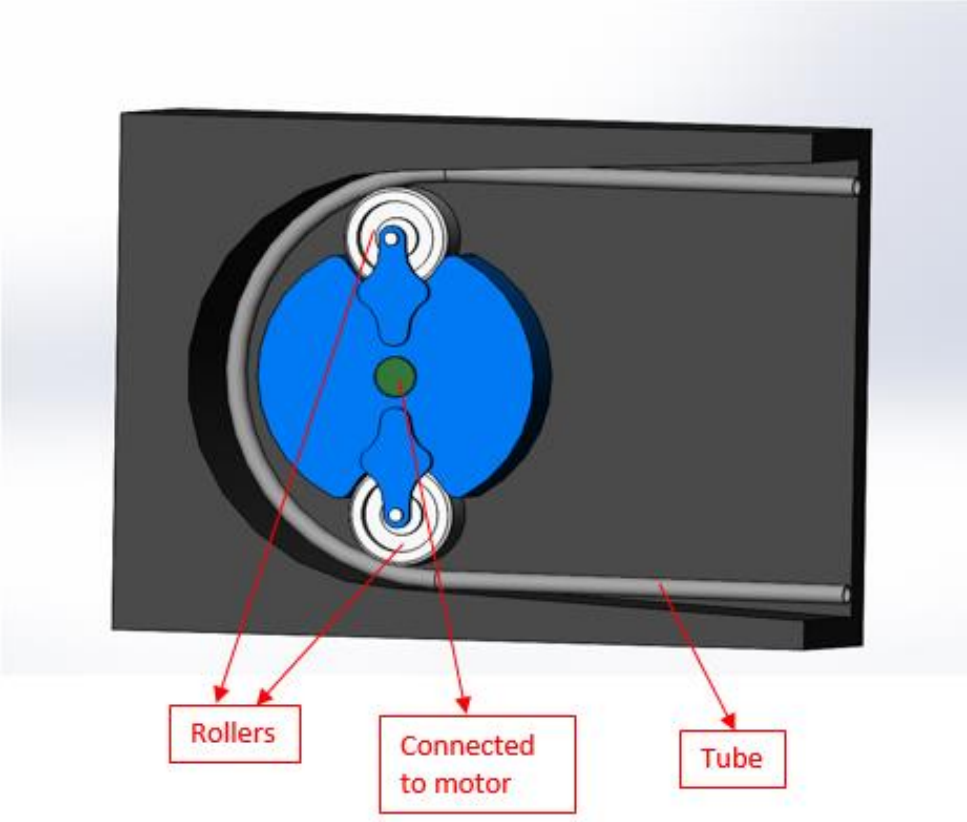


Figure 8: Labelled model of the peristaltic pump.

4.1 Physical and Mechanical Design

Peristaltic pumps are commonly used in applications where liquid must be moved at a slow, precise rate, without contamination. The peristaltic pump we will choose must accommodate standard IV tubing, which has a diameter of approximately 1mm. To minimize wear on the tubing, the rollers should not squeeze too tightly on the tubing, but they must also not squeeze too little, or the liquid may flow backwards when squeezed. [14] The number of rollers will also affect the system; more rollers means a more constant flow with less pulsation and more precise increments, however it also means more wear on the tubing [14]. To eliminate many of these wear issues, the Sentipump

will have the IV tubing threaded through the peristaltic pump. As the IV tubing will be replaced each time a new IV is administered, the tubing will not have enough time to wear through.

4.2 Electrical Design

The peristaltic pump will connect to the Arduino directly, as the speed of a stepper motor can be controlled by a microcontroller using a simple circuit with a transistor. The Arduino will output a number between 0 and 255, and each number will be a different preset speed. The pump will draw 12V at 80mA. For more precise speed control, the pump can be connected to the Arduino through an H-bridge [15]. The prototype version of the Sentipump may directly connect to the Arduino rather than using an H-bridge, if it proves effective enough in the testing stage.

5. Flow Rate Sensor

For the prototype version, it will be more cost effective to use a drip rate counter than a flow meter. The flow meters on the market that fall within our budget are not sufficiently accurate to show proof of concept for our design. In the production version, a flow meter accurate to 15 microlitres per minute will be used. A drip rate counter works by measuring interference between an emitter and a photo detector. When a drop breaks the plane between the emitter and the photo detector, the emitted light does not reach the photo detector, causing a change in voltage.

5.1 Physical and mechanical design

The drip rate counter will be attached to the drip chamber, to detect the drips. The photo detector will have a narrow angle of view, in order to detect the small deflections from the water droplets.

The drip chamber and the tubing of an IV system are manufactured to provide drips of exact individual volume. Common calibrations are 10 drops per milliliter, 20 drops per milliliter, or 60 drops per milliliter. Paired with a drip counter, the accuracy of the flow measurement for the calibrations would then be 0.1 mL/min, 0.05 mL/min or 0.016 mL/min respectively.

5.2 Electrical design

The drip rate counter for the prototype version will function by detecting the drips that fall into the drip chamber. The drops will cause the emitter light to deflect, and not reach the photo detector. The change in voltage will be observed by the Arduino, which will count the drops and calculate the flow rate from the known volume. The photo detector will detect visible light, as this will be easier to work with in the prototype phase. A circuit diagram for the drip counter is shown below in Figure 8.

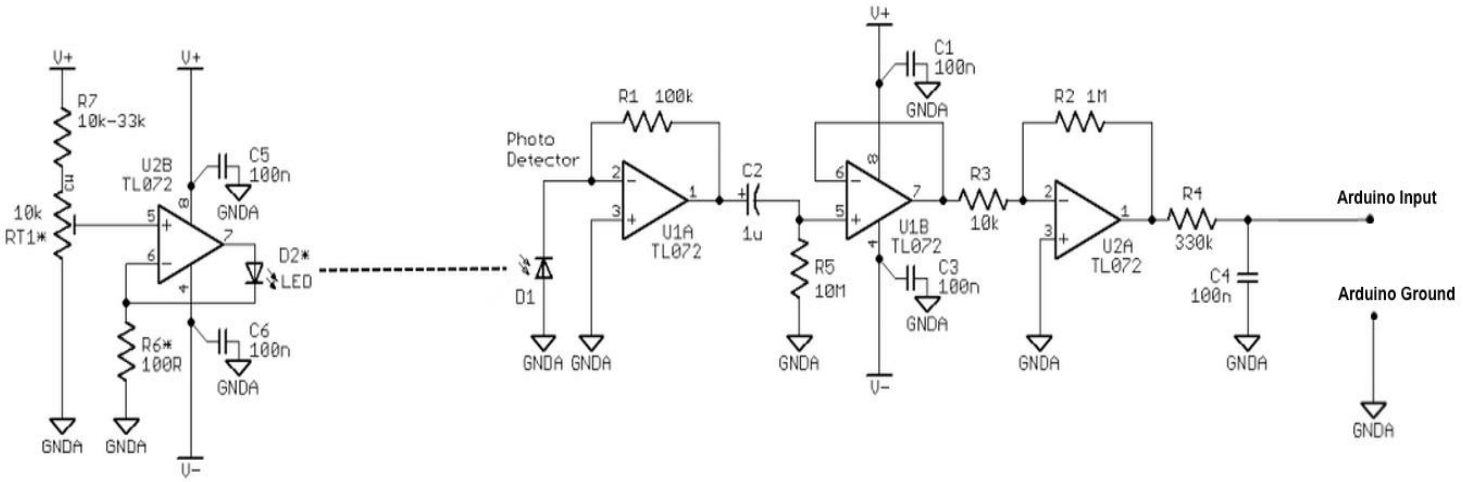


Figure 9: Circuit diagram for the drip rate counter.

6. Control Unit

The control unit will be implemented using either Arduino Uno or Arduino Pro depending on the power consumption of the stepper motor. There are 3 major reasons that an Arduino board has been chosen for this product implementation: it is cost effective, it provides high precision for the stepper motor, and it yields a high efficiency from the sensors. Furthermore, reference materials for the Arduino are abundant. Since Arduino I/O pins can provide very precise output voltage, calibrating with the stepper motor and the flow rate sensor will yield the desired outcome. In addition, an Arduino board is eco-friendly because the operating voltage of the board is 3.3V - 5 V.

6.1 Control Software

The controller will be programmed to do the following functions:

- Calibrating the pump with the flow sensor,
- Receiving the input flow rate from the mobile device,
- Analog sensor data translation,
- LCD formatting,
- Broadcasting patient's stats to mobile device.

6.2 Control Hardware

6.2.1 Flow sensor and peristaltic pump

The flow sensor and the pump will work in conjunction, and the flow sensor will act as a feedback control. Initially, the pump will be at rest, break or a rate of 0. When the desired flow rate is set through keypad, the controller will slowly calibrate the peristaltic pump until the preset value and flow sensor value become the same.

6.2.2 Mobile device and pump control

The control process for the mobile device is very similar to that of the flow sensor and the peristaltic pump. The only difference is that the preset value will be set through mobile device instead of the keypad on the control unit. The preset value can be set through a mobile device using a custom mobile application and the input value will be sent to the controller via Bluetooth connectivity. The calibration process will then be the same as that of the flow sensor and pump control.

6.2.3 Analog sensors control

A body temperature sensor and a pulse sensor will be connected to the analog port of the Arduino Uno, and the incoming data will be translated into the desired output form in order to be displayed.

6.2.4 LCD control

A 16x2 LCD will be directly connected to the Arduino Uno microcontroller via the digital port. The LCD will be powered through the 5V port of the microcontroller. The controller will translate the values for flow rate, body temperature, and heart rate, and will be format these values into LCD compatible data for display.

7. User Interface

The user interface will allow the user/caregiver to view and modify the patient's statistics, and also the rate of flow of the IV fluids. Sentipump will not only allow modifications on the device itself, but also through the mobile application. The mobile application details and interface are explained in Section 8.

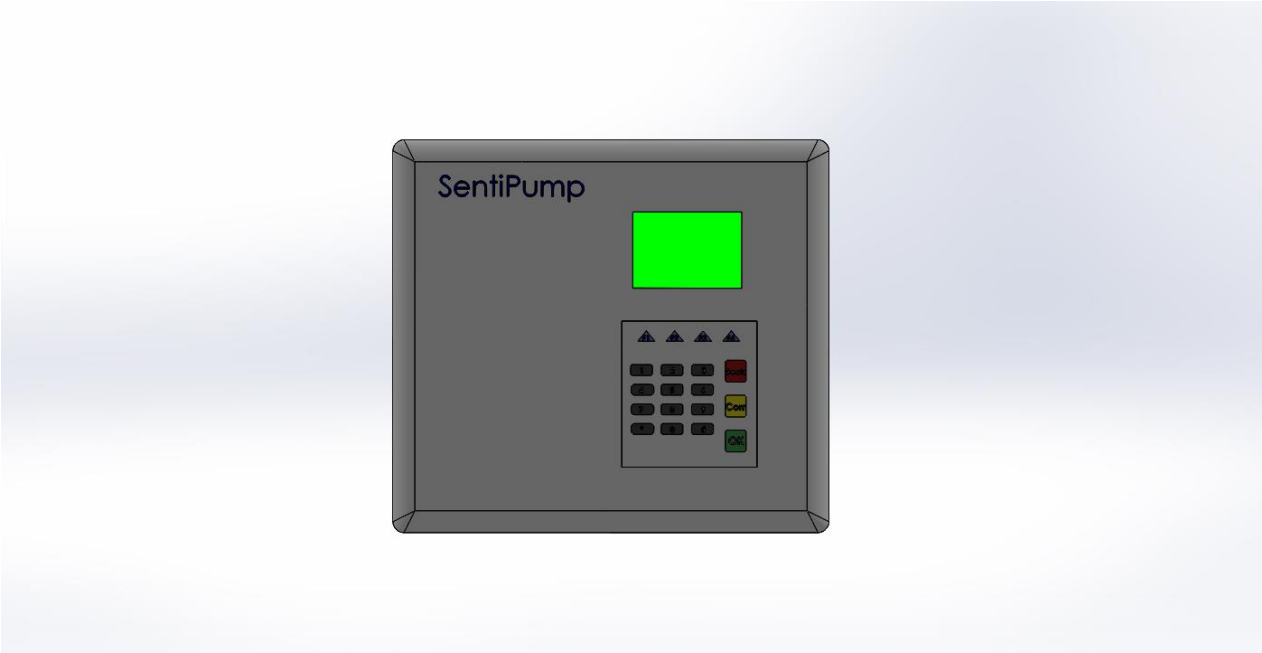


Figure 9: Conceptual drawing of the user interface.

7.1 Hardware interface

The hardware interface on Sentipump consists of an LCD and a keypad.

7.1.1 LCD

For our prototype, we will use a 16x2 LCD with a single line display (double lines display) in order to meet our goal of making our device compact and portable, and also to maximize efficiency of power usage. The LCD will display which part is currently being modified, and it will also display patient statistics like blood pressure, heart rate and body temperature.

For the production version, we will be using a bigger LCD, as shown in *Figure 10*. This will allow us to view multiple statistics at the same time. This will help save time, as the caregiver would not have to browse through the menu in order to monitor any particular health statistic.

Figure 10: Example of LCD screen for production version. [16]

7.1.2 Keypad

The keypad that will be used in the prototype consists of a power button, a numeric portion with the numbers 0-9, two scroll buttons, one select button, and one back button. Furthermore, there is a red push button to immediately stop the pump in case of emergency, a green button to start it up again, and also a yellow button to make any corrections to the values entered.

The production version of the keypad will include buttons along both sides of the LCD, as shown in *Figure 11*, making it easier to select a particular setting by pressing the button located next to it on the screen.

7.2 Software interface

The software interface will be used to register the values inputted by the user, and display the appropriate setting on the LCD. The changes will be implemented to the system within 500ms of the input from the user.

7.2.1 Registration Screen

The registration screen is the first screen the user will come across when using the Sentipump. For purposes of safety and uniformity, the user/caregiver will have to login to the Sentipump using a registration ID and passcode in order to make any changes to the pump settings. Any changes to login credentials will be made using the mobile app.

7.2.2 Main Screen

The main screen will display the patient's heart rate, blood pressure, current rate of infusion of the liquid, and the rate of completion of the current IV bag. For the prototype, the scroll buttons on the keypad will enable the user to browse through these settings, and change the required settings using the numeric keypad. In the production version, we hope to implement a system with a bigger screen and keypad, which will allow us to change settings directly from the main screen instead of browsing to find the right menu option. If there is an incorrect value entered by the user, the LCD will display an error message.

7.3 User Interface Verification

The user interface will be tested using the following methods:

1) Pushing all the buttons to make sure they perform the desired functions, and checking if the corresponding changes are displayed on the LCD,

2) Deliberately entering an incorrect value to see if an error message is shown on the LCD.

8. Mobile App Design

We plan to allow monitoring of the patient and the infusion status remotely, by communicating with our HC-06 Bluetooth module through a smartphone and/or desktop apps. The Arduino microcontroller we plan to use can communicate via a serial channel. Therefore, any other device with serial capabilities will be able to communicate with our system, and it does not matter which program/programming language drives the other device.[17] The design goal of our app is to be cost efficient, support as many device platforms as possible, while running basic functionality without having performance issues. The functionality we plan to implement is to receive data from the Arduino microcontroller to process and display, and send data to the microcontroller to control the infusion pump. Due to time constraints, auxiliary features will be developed in the final versions of Sentipump. These features are discussed in section 8.2, Mobile App.

We plan to develop the app using the cross-platform development kit Corona SDK. Of the available cross-platform development tools, we chose Corona SDK because it is the fastest solution for us to develop our app for the most number of device platforms, due to its complete coverage of devices supported. Instead of developing an app for one device at a time, Corona SDK will allow us to build our app for usage on all major device platforms, such as Windows, Mac, Android, iOS, Windows Phone. As a result, while our app can be used on many devices, more testing is needed to ensure that the app performs properly on all devices. For our prototype, we only plan to test the performance of the app on Android phones, but for future versions, ensuring that the app runs properly on other devices should not take much more work. The functionality we plan to develop in our app, such as sending, retrieving, syncing data, and displaying information, are basic enough that Corona SDK can support these functions while running seamlessly on all major device platforms. [18]

The following chart compares 3 essential criteria to consider for different methods of building an app: cost, hardware & OS access, and performance. Since the design goal of our app is to be cost efficient while supporting as many devices as possible, we want to use a cross-platform development method, since as shown in the graph, cross-platform design methods are the most cost-efficient by far when designing for multiple device platforms. In exchange for lower costs, hardware and OS access, as well as performance is decreased. This is not a problem, however. Since our app uses only basic functionality, we will have access to all the functionality we need. Furthermore, even with a small loss in performance, basic functionality such as sending, retrieving, syncing data, and displaying information do not take much processing power, and are simple enough for there to not be any performance issues.

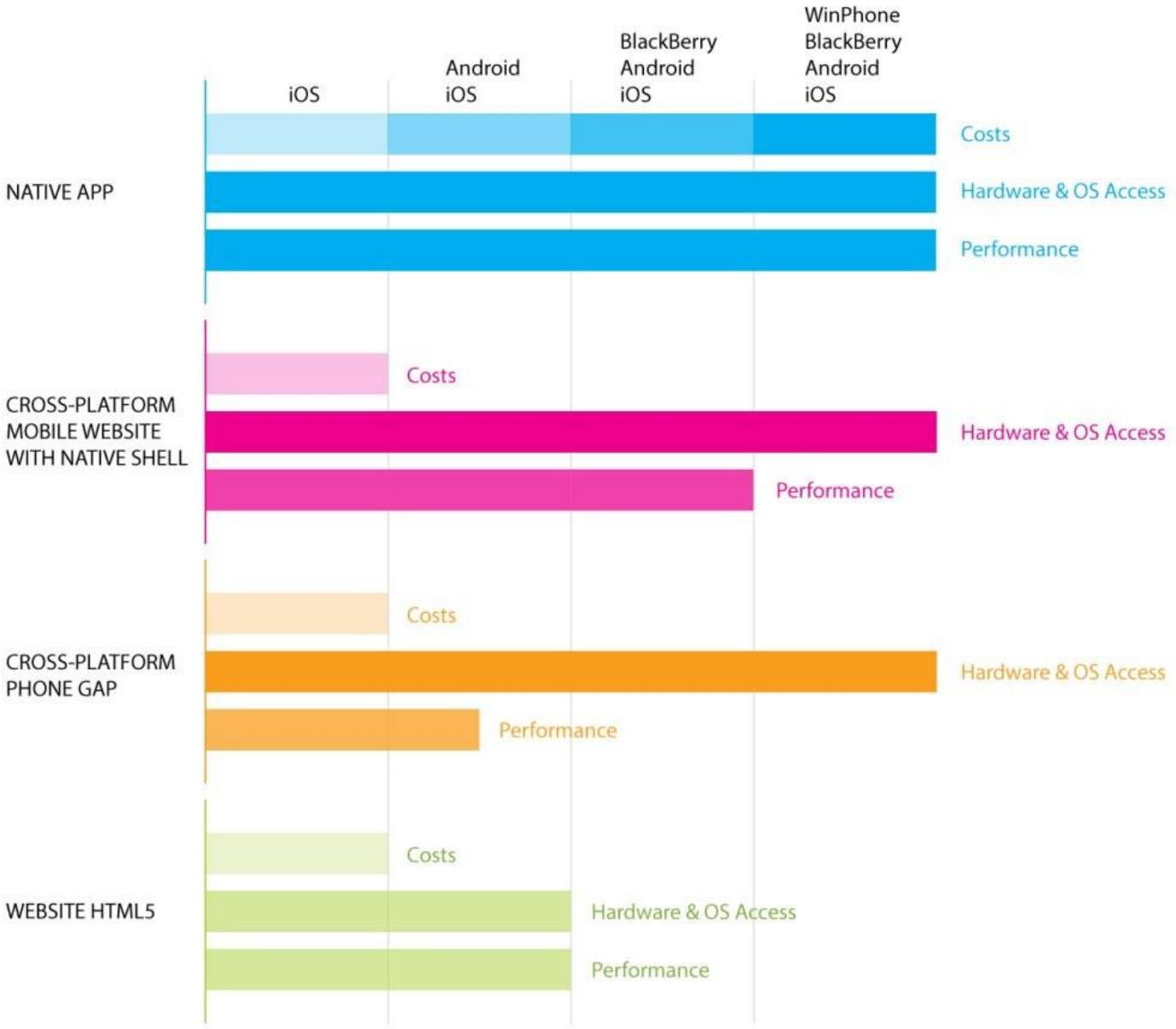


Figure 12: Comparison of essential criteria between different app development methods.[19]

The primary data we want the app to display include infusion duration, infusion rate, delivery volume, dosage concentration, patient and practitioner ID, and data from attachments such as body temperature and heart rate. When values reach provided thresholds, the practitioner can be warned about it immediately. The figure below shows a conceptual drawing of the application's user interface on a mobile device.

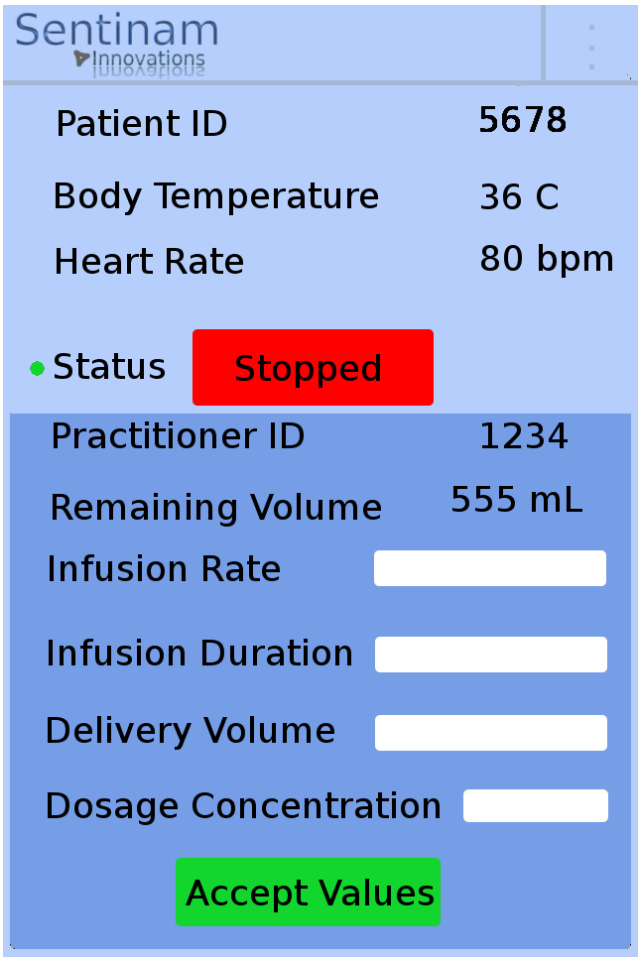


Figure 11: Conceptual drawing of the application's user interface.

8.1 Mobile App Interface:

The colour and style design of the interface will be intuitive, with the buttons for confirmation coloured green, and the buttons for canceling coloured red. Text will be lined up to clearly indicate what information is provided, so practitioners can quickly monitor a patient's health status. The interface will provide a button for practitioners to confirm parameters for the infusion process, and to start and stop the process. There will also be a settings page to change the alarm threshold value and alarm sound.

8.2 Mobile App Features:

An internet connection will be required to access the application. We considered establishing a connection between the microcontroller and user's device using the standard Arduino WiFi Shield or Ethernet Shield, by creating a server which a user's device can connect to. However, in the case of the WiFi shield, the code libraries provided by Arduino are not able to be used to flexibly connect to any network the user wants, because the network accessed has to be hard-coded. In addition to this lack of flexibility, the library does not inherently support the username/password authorization login method on SFU campus.

For the case of the Arduino Ethernet shield, using this requires an Ethernet cable connection to a router, which is an inconvenient restriction in placement since our microcontroller will be attached to our infusion pump, and it this requirement is unlikely to be fulfilled both for our demo and in a production environment.

Instead, we decided to establish connection using Bluetooth. In addition to being more cost-efficient than the WiFi and Ethernet Shields, since the caregiver should be reasonably close to the patient when providing care, the 9 meter connection range offered by our choice of bluetooth, HC-06, is suitable for the purpose of our product. Also, the bluetooth module uses fewer pins than the Wifi and Ethernet Shield, which

allows us to save the few remaining available pins on our Arduino Uno for other attachments. [20]

After we pair up the microcontroller with a user's device, the microcontroller and device can then communicate by transferring data packets which our system will decipher by parsing each byte of the packet [21], [22].

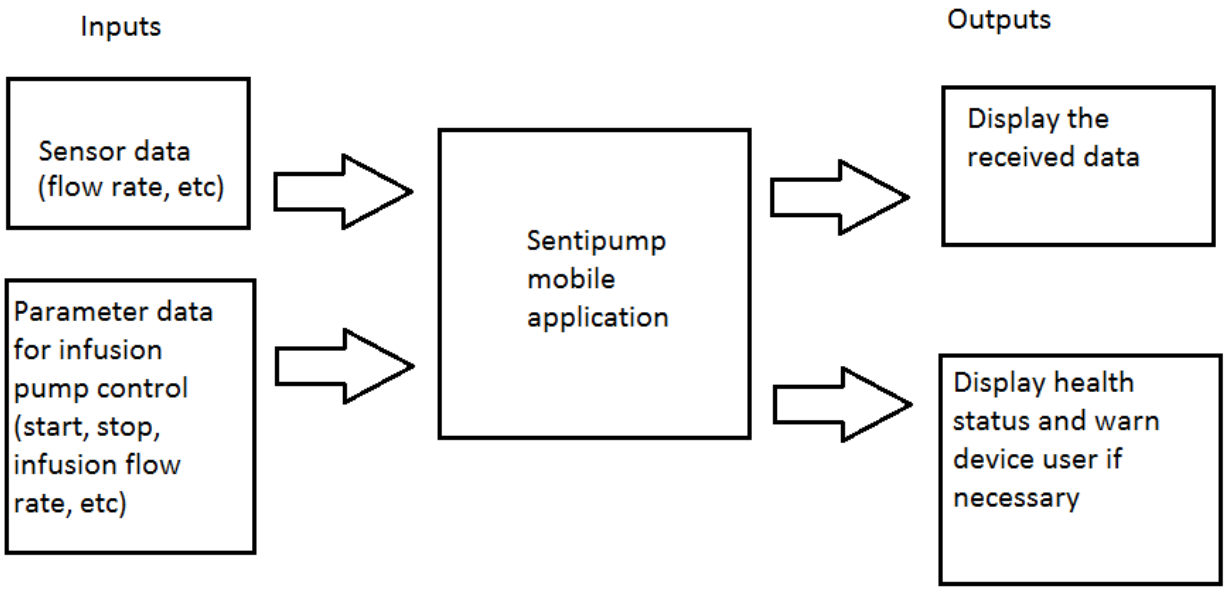


Figure 12: Input and output chart for the prototype version of the mobile app.

Through the app, the user will be able to modify values used to control the system IV infusion flow, monitor the patient's health status, and start and stop the infusion process. An alarm will sound when the health status of the patient reaches a preset threshold value. Also, there will be a login page to enter patient and practitioner ID, to prevent unauthorized access.

For our prototype, we plan to include the primary features in our mobile application to demonstrate its remote monitoring, connection, and control capabilities.

Due to time constraints, auxiliary features are implemented in the final versions of Sentipump, including authorization of practitioner and patient login, and security protocols. In addition, the final versions of Sentipump will provide access control to ensure values can only be changed from one device at a time.

8.3 Bluetooth Module Circuit

The following is the circuit configuration we will use to connect the bluetooth module. The module used is HC-06.

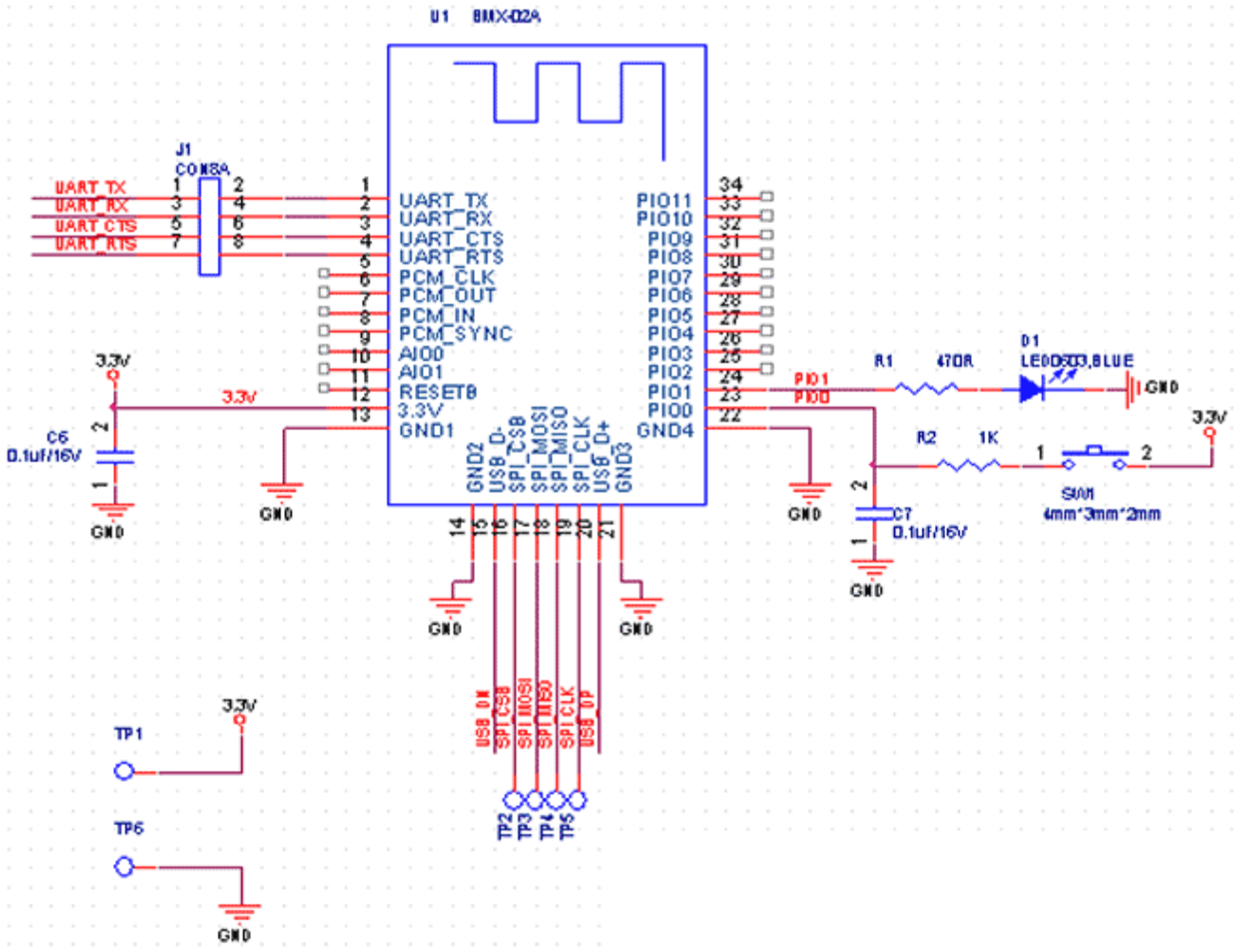


Figure 13: Circuit configuration used to connect the Bluetooth module. [23]

The LED and button will be used to test the Bluetooth connection between the Arduino and user's device. HC-06 module has 4 pins to be connected to arduino:

- RXD, TXD, VCC, GND
- RXD receives data from arduino, TXD sends data to arduino, VCC is the power supply (3.3V 6.6V), and GND is the ground.

After connecting the Bluetooth module according to picture, we connect the pins as follows:

- RX Arduino pin- to TXD HC-06 Bluetooth module,
- TX Arduino pin- to RXD HC-06 Bluetooth module,
- +3.3V Arduino pin- to VCC HC-06 Bluetooth module,
- GND Arduino pin- to GND HC-06 Bluetooth module.

9. Test Plan

Sentinam Innovations intends to perform rigorous unit and integration tests to ensure our system design meets our high functional standards and expectations. Ensuring that the system delivers an accurate infusion rate is the highest priority. After the main feature is tested, we will then test the side features, such as additional sensor attachments and WiFi connection to mobile/desktop applications. As features are modified, connected features will have to be retested as well to ensure new changes do not break the other components.

We will be using bottom-up testing to systematically ensure Sentipump functions properly. Components will first be tested individually with a range of valid and invalid input values, which will be simulating values to be obtained from other connected components. Then we will perform integration testing to make sure the system functions correctly when the components are connected together.

For our prototype, tests will be conducted by pumping water into a container with a known volume size, in place of a person. Measurements will be carried out on the container to ensure the system accurately delivers the correct amount of substances. For the final versions of our product, more realistic substances will be used to simulate the different drugs that can be delivered to patients.

9.1 Unit Test Cases

Components will be tested individually in this stage of testing. The following tests will be performed to ensure each component functions as required.

- **Pump hardware control:** Verify that the system controls are responsive and reliable, and pushing the buttons causes an interrupt to occur with little delay. Verify that the patient health status is indicated by an LED, and the alarm triggers when thresholds for vital health statistics are reached. The power from a wall outlet must be sufficient to drive the system without the system shutting down.

- **Pump software control:** Verify that the software runs smoothly. Verify that the components of the GUI corresponds to the correct functionality, and the display values match the infusion pump values. Verify that the software handles a range of valid and invalid inputs accurately, and values that go out of range are prevented by displaying an error message. Verify that the system triggers the alarm when thresholds for vital values are reached.

- **IV tubing and bag:** Verify that the tubes and bags are compatible and the fluid flows inside them seamlessly. Verify that they can be recycled and easily replaced.

- **Peristaltic pump:** Verify that the pump can be controlled to deliver the required infusion flow rate.

- **Mobile app:** Verify that the app communicates with the microcontroller and handles data accurately. Verify that the software runs smoothly. Verify that the components of the GUI correspond to the correct functionality, and display values match the infusion pump values. Verify that the software handles a range of valid and invalid inputs accurately, and values that go out of range are prevented. The app shall enter into error mode if any components are not working correctly. Verify that when thresholds for vital values are reached in error mode, the infusion pump becomes manually adjustable only, the flow of any fluids is halted, and the patient's caregiver is alerted through the device.

9.2 Integration Test Cases

Once each component passes unit tests and is determined to be functioning properly, we move to integration tests. The following tests will be carried out to evaluate how well components of the system work together.

- **Infusion pump control test:** Verify that the motor is capable of delivering values that match the system input parameters, by attaching infusion

pump to a container and measuring the substances delivered to the object.

- **Bluetooth connection test:** Verify that the system can establish a stable connection to the supported devices and desktops, and can function with low connection speeds. Infusion pump primary features should function properly even without the Bluetooth connection. Verify that the Bluetooth can connect with the user's device at a range of up to 9 meters, by slowly moving the device away from the microcontroller.

- **Infusion pump mobile app control test:** Verify that the infusion rate of the infusion pump can be controlled from a mobile device using our app, by attaching infusion pump to an object, and measuring the substances delivered to the object through the app. Verify that the motor is capable of delivering values that match the system input parameters.

- **Stability test:** Verify that the infusion pump must be able to run continuously without maintenance by leaving it running continuously for 2 days.

- **Sensor accuracy test:** Monitor the sensors and verify they are able to send signals to the microcontroller, and the values displayed on the LCD are accurate.

10. Conclusion

The specifications provided in this document serve as design solutions for the functional requirements of Sentipump, a remote monitored infusion pump. The document highlights the design method, description and justification for the parts used, and relevant drawings and schematics of the various components used to develop the Sentipump prototype. This document discusses the features that will be added in the final versions of our system, but does not delve into its implementation details. Prototype features are marked by I and II in our functional specifications document, while the final version features are marked by II and III.

A test plan is also provided in this document, which our team will use to verify the quality of Sentipump. In our test plan, we describe the unit and integration tests we will use to ensure our system design meets our high functional standards and expectations.

Sentipump will go through some changes in design during the course of development of the prototype. However, this document will act as a clear basis for the design and will be modified with necessary changes accordingly, until the end of the development phase.

References

[1] A. Fang, C. Currie, P. Malik, B. Singh, N. Moe, “Functional Specification for a Remote Monitored IV for Home Care,” unpublished.

[2] Verderflex, ‘How do peristaltic dosing pumps work?’, 2008. [Online]. Available: http://www.verderflex.com/en/how-do-peristaltic-pumps-work/. [Accessed: 07- Nov- 2015].

[3] C. Quinn. ‘Intravenous flow control and infusion devices,’ in Intravenous Therapy in Nursing Practice (2nd ed.), L. Dougherty and J. Lamb, Eds. 2009.

[4] Arduino, 'Arduino Motor Shield', 2015. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoMotorShieldR3. [Accessed: 08- Nov- 2015].

[5] Dayton T. Brown Inc., 'A Primer on Electromagnetic Shielding', 2015. [Online]. Available: http://www.dtbtest.com/electromagnetic-shielding-primer.aspx. [Accessed: 04- Nov- 2015].

[6] Amazon, 'ZJchao 12V DC DIY Dosing pump Peristaltic dosing Head with Connector For Aquarium Lab Analytic By GHH', 2015. [Online]. Available: http://www.amazon.ca/ZJchao-Peristaltic-Connector-Aquarium-Analytic/dp/B00F9MXFFQ/ref=sr_1_1?ie=UTF8&qid=1444946264&sr=8-1&keywords=peristaltic+pump. [Accessed: 22- Oct- 2015].

[7] Arduino, ‘Arduino Uno & Genuino Uno’, 2015. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardUno/. [Accessed: 22- Oct- 2015].

[8] HC Serial Bluetooth Products, 'User Instructional Manual ', 2015. [Online]. Available: http://abc-rc.pl/templates/images/files/995/1425483439-hc-06-datasheet.pdf. [Accessed: 05- Nov- 2015].

[9] Xiamen Amotec Display CO., LTD, 'Specifications of LCD Module', 2008. [Online]. Available: https://www.sparkfun.com/datasheets/LCD/ADM1602K-NSW-FBS-3.3v.pdf. [Accessed: 05- Nov- 2015].

[10] Parallax, '4x4 Matrix Membrane Keypad (#27899)', 2011. [Online]. Available: https://www.parallax.com/sites/default/files/downloads/27899-4x4-Matrix-Membrane-Keypad-v1.2.pdf. [Accessed: 05- Nov- 2015].

[11] Techmeres, 'Medical & Healthcare Case Study: Infusion Pump Housing Performance', 2015. [Online]. Available: http://www.techmeres.com/2015/03/04/medical-healthcare-case-study-infusion-pump-housing-performance/. [Accessed: 29- Oct- 2015].

[12] Klespitz, J.; Kovacs, L., 'Peristaltic pumps — A review on working and control possibilities,' in Applied Machine Intelligence and Informatics (SAMI), 2014 IEEE 12th International Symposium on , vol., no., pp.191-194, 23-25 Jan. 2014

[13] De Silva, C. W. (2007). Sensors and actuators : Control systems instrumentation (1st ed.). Boca Raton, FL: CRC Press.

[14] [Blue-White Industries, 'Peristaltic Pump Wear Factors', 2015. [Online]. Available: http://blue-white.com/peristaltic-pump-wear-factors/. [Accessed: 09- Nov- 2015]

[15] Arduino, 'Stepper Speed Control', 2015. [Online]. Available: https://www.arduino.cc/en/Reference/Stepper.[Accessed: 08- Nov- 2015]

[16] Arduino, 'Arduino TFT LCD Screen', 2015. [Online]. Available: https://www.arduino.cc/en/Main/GTFT/. [Accessed: 09- Nov- 2015].

[17] Arduino, 'Interfacing With Other Software', 2015. [Online]. Available: http://playground.arduino.cc/Main/InterfacingWithSoftware. [Accessed: 10- Nov- 2015].

[18] Corona Labs, "Corona SDK", 2015. [Online]. Available: https://coronalabs.com/. [Accessed: 10- Nov- 2015].

[19] A. Goldenberg, 'Pros and Cons of Developing Native vs. Cross-Platform Web-Based Mobile Application', 2013. [Online]. Available: http://www.dbbest.com/blog/pros-and-cons-of-developing-native-vs-cross-platform-web-based-mobile-application/. [Accessed: 05- Nov- 2015].

[20] D. Calin, 'How to pick the right Bluetooth module for your DIY Arduino project', 2015. [Online]. Available: http://www.intorobotics.com/pick-right-bluetooth-module-diy-arduino-project/. [Accessed: 08- Nov- 2015].

[21] ANATOLIY, 'Data Transfer Between Android And Arduino Via Bluetooth', 2012. [Online]. Available: http://solderer.tv/data-transfer-between-android-and-arduino-via-bluetooth/. [Accessed: 06- Nov- 2015].

[22] Deyson, 'Android Bluetooth Control LED. (Part 2)', 2015. [Online]. Available: http://www.instructables.com/id/Android-Bluetooth-Control-LED-Part-2/?ALLSTEPS. [Accessed: 08- Nov- 2015].

[23] ANATOLIY, 'Arduino And Bluetooth', 2012. [Online]. Available: http://solderer.tv/arduino-and-bluetooth/. [Accessed: 06- Nov- 2015].