Toronto Metropolitan University

Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering & Architectural Science

The Automated Tourniquet

Sayeed Hasan[1*]

Hassan Khan[1*]

Haashim Shahzada[1*]

Diana Temelkos[1*]

[1]Department of Electrical, Computer, & Biomedical Engineering, Faculty of Engineering & Architectural Science, Toronto Metropolitan University

[*]These authors contributed equally

Biomedical Engineering Capstone Design Project

Toronto Metropolitan University, 2024

Acknowledgements

We would like to acknowledge the contributions of Dr. Ali Tavallaei for his invaluable support and guidance in the design of the automated tourniquet. We would also like to thank Josh Richmond and Yi Yue for their invaluable feedback and advice throughout the project.

Certification of Authorship

We, the undersigned, hereby certify that we are the authors of this document and that any assistance we received in its preparation is fully acknowledged and disclosed in the document. We have also cited all sources from which we obtained data, ideas, or words that are copied directly or paraphrased in the document in IEEE format. Sources are properly credited according to accepted standards for professional publications. We also certify that this paper was prepared by us for this purpose.

Date of Submission: April 30th, 2024

Purpose and Title of Submission: Capstone Design Project Report - The Automated Tourniquet

FLC Details:

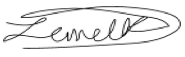
Name: Ali Tavallaei

Email: ali.tavallaei@torontomu.ca

Student A Details:

Name: Diana Temelkos

Email: dtemelkos@torontomu.ca

Signature: 

Student B Details:

Name: Hassan Khan

Email: hassan.a.khan@torontomu.ca

Signature: 

Student C Details:

Name: Haashim Shahzada

Email: hshahzada@torontomu.ca

Signature: 

Student D Details:

Name: Sayeed Hasan

Email: sayeed.hasan@torontomu.ca

Signature: 

Table of Contents

Abstract

Uncontrolled bleeding in emergency and military settings presents grave dangers, frequently resulting in fatalities. Although traditional manual tourniquets are effective, they carry inherent risks such as potential arterial damage. This project is dedicated to the development of an automated tourniquet system, aiming to mitigate these challenges and enhance practitioners' precision in achieving limb occlusion for surgeries or vein engorgement for venipuncture procedures. The automated system dynamically adjusts pressure based on user input and medical guidelines, ensuring precise and timed application. The research underscores the incorporation of sensors into the conventional tourniquet design to provide a smoother, less variable, and safer alternative to the manual method, ultimately advancing patient care and outcomes in critical situations.

Keywords: automation, tourniquet, occlusion, venipuncture, hemorrhage, closed-loop, vein engorgement

1.0 Introduction & Background

1.1 Tourniquet Usage

Uncontrolled bleeding presents a critical challenge across diverse settings, where its management is paramount for saving lives. Military combat medical care has identified it as the leading cause of death [1], while non-military medical care ranks it as the second-most significant cause of traumatic death [2]. In response to this urgent need, tourniquets have emerged as a vital tool for stemming uncontrolled blood flow, particularly in cases of extremity injuries.

Traditionally crafted from rubber, tourniquets are strategically applied proximal to the point of injury to halt blood flow, often utilizing a combination of mechanical pressure, straps, and Velcro mechanisms for tightening. These simple yet effective devices serve dual purposes in medical practice. Firstly, they induce vein engorgement, facilitating procedures like venipuncture for blood testing [3]. Secondly, they enable full-limb occlusion for specific surgical interventions by completely arresting arterial blood flow [4].

The versatility of tourniquets extends beyond their immediate application in emergency situations. They play a crucial role in both military and civilian healthcare settings, providing frontline responders and medical professionals with a rapid and effective means of hemorrhage control. As advancements continue in materials and design, tourniquets are evolving to offer even greater reliability and ease of use, further enhancing their life-saving potential in the face of traumatic injury.

1.2 Gaps in Current Designs

Tourniquets stand as indispensable tools in patient care; however, the current manual venipuncture tourniquet system grapples with two critical challenges: lack of smoothness and high-pressure variability. The absence of smooth operation can stem from potential misoperation, particularly in high-stress scenarios such as combat medical care, where hurried actions may lead to jerking motions and sudden pressure on vital tissues, ultimately risking patient mortality [5]. These difficulties are compounded in situations involving multiple patients, where time constraints and environmental stressors further impede smooth application [6].

Moreover, variability in pressure application poses a significant concern, as it introduces inconsistencies in the tightness of the tourniquet. This variability arises from differences among individuals tightening the tourniquet, influenced by factors such as stress levels, physical strength, training, and experience [7]. Such lack of standardized control over pressure can result in unforeseen adverse effects, ranging from complications due to excessive tightening to inadequate compression leading to patient hemorrhage or even death [8]. The incorrect application or improper tightening of a manual tourniquet significantly complicates the patient's recovery process or, worse, diminishes the likelihood of a favorable outcome altogether.

Addressing these challenges is imperative for enhancing patient safety and optimizing the effectiveness of tourniquet use in medical practice. Innovative solutions aimed at improving the smoothness of operation and standardizing pressure application are essential to mitigate risks and ensure the reliable performance of tourniquets in critical care scenarios.

1.3 Needs Analysis

1.3.1 User Needs

The automation of the manual tourniquet will advance patient care by providing healthcare professionals with an intuitively operated, versatile device that prioritizes patient safety, comfort, and adjustability for both venipuncture and full occlusion procedures. Furthermore, it should offer a portable and durable solution, enhancing efficiency in clinical settings.

1.3.2 Market Needs

Current automated tourniquet devices primarily focus on full occlusion, as seen in products like the *Zimmer Biomet A.T.S.* [9] and the *Stryker SmartPump Tourniquet System* [10]. However, a gap in the market remains for automated tourniquets designed specifically for venipuncture procedures.

1.3.3 Basic Design Needs

The design of an automated tourniquet device must incorporate several key elements to meet the diverse needs of users and regulatory standards. These include precise pressure control mechanisms, the use of skin-friendly and durable materials, compact and lightweight form factors, flexible power sources, a comprehensive array of safety features, user-friendly interfaces, adjustability for different arm sizes, and seamless integration capabilities. Moreover, it must ensure ease of cleaning and maintenance.

1.4 Problem Statement

The development of an alternative to the manual tourniquet device is essential to minimize or eliminate potential dangers to patients, particularly arterial damage and pressure variability, associated with manual operation.

1.5 Significance and Potential Impact

Anticipated outcomes of the device development include addressing inherent issues found in manual tourniquets, catering to individuals in clinical and military settings irrespective of experience or stress levels. Moreover, it is expected to lead to a reduction in patient mortality and a decrease in the frequency of recovery complications attributed to the use of the developed device.

2.0 Objective

The objective of this project is to design and develop an innovative automated tourniquet system incorporating a closed-loop control system with sensory input, thereby enabling dynamic pressure adjustments. This system is specifically designed to address and overcome the challenges associated with the manual method, with a focus on ensuring aesthetics, intuitive usage, retention of manual functionality, and cost-effectiveness. The overarching goal of this endeavor is to create a novel system that not only enhances patient care but also ensures accessibility to primary medical interventions for patients in need of reliable healthcare services.

3.0 Theory and Design Overview

3.1 Hardware Design

The hardware design encompasses six key components: the microcontroller, power supply, battery charger, pressure sensor, screen interface, and motor driver. Each component fulfills a crucial and interconnected role in the overall implementation of the circuitry. All components used in this project were sourced from Digikey, while the Printed Circuit Boards (PCBs) were procured from JLCPCB. This section will comprehensively review each design choice along with its respective justification to ensure the creation of a functional and effective tourniquet device.

3.1.1 Microcontroller

The design problem involved selecting a microcontroller that met specific criteria for the automated tourniquet. A detailed list of requirements, including low cost, low battery consumption, communication interfaces, and form factor considerations, was prepared. The selected ESP32 met all criteria and offered a balance between performance and ease of integration [11]. Other options briefly considered included the Atmega328, PIC24 and STM32 microcontrollers manufactured by MicroChip and STMicroelectronics. They were quickly discarded due to lack of connectivity, compatibility or high cost. The NRF7001 offered lower power consumption; it was not chosen due to difficulties in working with its SDK and driver incompatibilities [12]. The approach involved a systematic evaluation of microcontroller options based on the established criteria. Eventually, after literature and documentation reviews, Espressif System's ESP32 microcontroller was chosen.

To optimize the design and reduce costs, the ESP32-S3-WROOM module, equipped with a bootloader and essential hardware like a crystal oscillator (48kHz), was chosen over the ESP32-WROOM-32 UE microcontroller. The ESP32 operates at an input supply of 3 - 3.6V (typically 3.3V) and requires around 500 mA to run without brownouts, allowing approximately 250 mA for distribution among connected peripherals [11]. The documentation, availability, and compatibility with the project's requirements were the main factors that led to its selection. The ESP32-S3-WROOM module was chosen for its pre-existing hardware features, reducing the need for additional components and complexity while maintaining the ability to create a custom embedded system for the device [11]. Tools used to conduct research included datasheets, and the aforementioned evaluation criteria. The chosen microcontroller was incorporated into the design using schematic/PCB design software, considering factors such as power requirements, pin compatibility, and available communication interfaces.

3.1.1.1 Serial Communication

The ESP32-S3-WROOM microcontroller offers two primary methods for programming: through the UART pins or via the built-in USB JTAG connector. In this design, to reduce costs, and minimize potential sources of error, the UART pins are broken out with a pin header and used to communicate with the device through any external USB to UART bridge. The USB receptacle can then be plugged into any device to provide power to the circuit. To ensure stable power, the USB C receptacle is connected to two 5.1 kΩ resistors on the CC1 and CC2 pins in accordance with the standard [11][13].

3.1.1.2 Analog to Digital Converter

The ESP32-S3 microcontroller has a 12 bit ADC used to convert any analog signal ranging from 0 to 1.1V to a digital signal that the microcontroller can quantize and assign values to. The ADC in the ESP32-S3 microcontroller has 10 channels and a median reference voltage of 1.1V used to convert signals. The ADC can also convert signals with larger voltages through built-in attenuation options. The ESP32-S3 comes with 2 built-in ADCs with these parameters but in this case only ADC1 is used because ADC2 is unavailable whenever Wi-Fi is turned on. The ADC values are calculated using the equation $V_{out} = D_{out} * \frac{V_{max}}{D_{max}}$, where Vout is the digital output result, Dout is the ADC raw digital reading, Vmax is the maximum measurable input analog voltage (1.1V in this case) and Dmax is the maximum of the output ADC raw digital reading result [11].

Since the ADC is being used for battery consumption measurements, the 12 bit resolution capable of 4096 analog levels ($calculated\ using\ 2^n$). With an attenuation of 11 dB would allow for battery measurements from 0-2900 mV at a respective step of 0.71 mV ($\frac{2900}{4096} = 0.71$). This far exceeds the resolution required for the battery monitoring system as a 1% decrease in the battery occurs every 53 mV ((8600-3300)*0.01 = 53). To get within the range of the 0-2900 mV ADC voltage input range, a voltage divider of 10kΩ and 20kΩ resistors is used to bring the voltage down to 2.86V which is near the top end of the readable voltage range of the ADC [11].

3.1.1.3 MCU Operation

The microcontroller reset design aimed to provide a fail-safe mechanism for system resets in case of a crash or hang. A reset button was added to toggle the EN pin of the ESP32. Alternative reset mechanisms, such as software-based resets, were considered but a physical reset button was chosen for its simplicity and reliability. The approach involved adding a reset button connected to the EN pin, allowing users to manually reset the microcontroller when needed [11].

3.1.2 Power Supply and Voltage Regulation

As shown in Figure 1 below, the tourniquet system is powered by two main voltage lines, VBUS (from a plugged-in USB connector) or VBAT (from an included 7.4V 2200 mAh LiPo battery). To ensure the circuit operates seamlessly on either battery or USB power, both power lines pass through a protection circuit. This circuit enables the source with the highest voltage to power the system while preventing reverse current and voltage spikes from damaging the battery or USB power source. The first part of the circuit includes a Schottkey diode in series with the VBUS power line and a solid state relay (SSR) in series with the VBAT power line. While the Schottky diode acts as a reverse voltage and current blocker, it introduces a high voltage drop. To mitigate this, a load-sharing circuit is implemented using a SSR and a 100kΩ resistor. This ensures the SSR remains off when there's no voltage on the USB power line but turns on as soon as the battery power line has voltage, avoiding the voltage drop effect of the diode on the battery line. Following this, a 10 uF decoupling capacitor is placed in parallel to the voltage line to remove noise and smoothen the signal [11][13].

The signal progresses into a 3.3V LDO regulator (LM1117-3.3V), which stabilizes the incoming voltage from the VIN pin to a consistent 3.3V. The voltage signal leaving the regulator through VOUT is met by four different decoupling capacitors ranging from 0.1 uF to 100 uF in

parallel to eliminate additional noise at multiple corner frequencies before delivering a clean 3.3V signal to the rest of the system [14].

The choice of the LM1117-3.3V LDO was influenced by its compact size, simplicity, wide availability, cost-effectiveness, and straightforward operation. Notably, the LDO exhibits a low current consumption at 10 mA when not in operation, while guaranteeing an output current of 800 mA, surpassing the required 500 mA for the ESP32 module. With a maximum input supply voltage of 15 V and a low voltage drop of only 1.2 V, the LDO can efficiently regulate the 7.4 V battery voltage when it is at its maximum potential charge of 8.6V to 3.3V. As an alternative, the AMS1117-3.3V LDO could be considered for its 1A output current, 3.3V output voltage, and 15V input supply voltage tolerance in a different design approach [15].

3.1.3 Pressure Sensing

The primary sensor selected for the pressure sensing design was the piezoresistive sensor by Honeywell, MPRLS0025PA00001A as it checked all the requirements in the criteria. The MPR series is a 5 mm x 5 mm [0.20 in x 0.20 in] piezoresistive silicon pressure sensor that provides a digital output for pressure over the specified full-scale pressure span, calibrated and compensated, and a wide pressure range of 0-1290 mmHg. The digital output is amplified, offers reduced conversion requirements, and a direct interface to microprocessors avoiding additional circuit complexity. The sensor can give a max frequency output of up to 160 Hz and can detect low-frequency signals as well. Additionally, it has a voltage supply of 1.6 - 3.6 V, and has a current consumption of 3 - 211 nA (typically 33.8 nA). The MPR sensor is from a reliable manufacturer and is well known for being used in medical applications such as blood pressure monitoring and breast pumps [16].

The backup pressure sensor selected is the piezoresistive MPX5050DP by NXP USA Inc. It operates with a pressure range of 0-375 mmHg and has an analog output that is conditioned. It has a high response time of 1 ms, a low voltage supply of 4.75V-5.25V, and an accuracy of 2.5% FSS. This pressure sensor is more expensive than the MPR sensor but is also from a reliable manufacturer and has been found to be used in the development of a self-tightening tourniquet system [17].

The Honeywell Piezoresistive sensor's sensing mechanism seamlessly integrates into the system through comprehensive system controls and a PID controller setup. This sensor captures pressure data crucial for the system's operation, particularly within a timer-based algorithm that continually monitors pressure thresholds. Notably, should the sensor detect pressure exceeding the target by 10 mmHg, a solenoid pin promptly engages to release pressure, ensuring over-inflation is averted. Given its pivotal role, the sensor delivers precise feedback, enabling the system to execute accurate pressure adjustments while prioritizing safety protocols. The MPR sensor is interfaced with the ESP32-S3 through the I2C communication protocol. This is done through the Adafruit MPR Sensor library which assigns the I2C address for the sensor and performs the initial calibration for the sensor. This calibration gives the sensor

3.1.4 Actuation

By definition an automated tourniquet needs to automatically apply enough pressure to an individual’s arm to either engorge veins for venipuncture or stop blood flow for procedures. To do this there are a few possible solutions that were analyzed and ultimately rejected due to design and performance constraints. These implementations were found by reviewing various literature from industry and academia and have been compiled in Table 1 along with both their negative and positive traits. Eventually, due to its cost of implementation, availability, precise control,

pressure distribution, and flexibility, a pneumatic actuation system was chosen for this application.

Table 1: A review of all considered actuation systems

Actuation System	Pros	Cons
Synthetic EAP	• Lightweight • Easy to control using precise PWM pulses of low voltages • Exact fit to the area of interest	• Manufacturing the polymer is hard • Pressure sensing capability lacks resolution • It's hard to put the sheer amount of force it requires to make a tourniquet effective
Pneumatic System	• Cheap • Components are easy to find • Proof of concept already exists • Widely used • Flexible applications • Provides precise pressure control	• Potentially difficult to adjust based on tourniquet site (e.g tourniquets are used on thigh and on arm) • Might cause bruising, injury as seen in doi: 10.1016/j.jcot.2020.09.005
Non-pneumatic System	• Cheap • Components are easy to find • Proof of concept already exists	• Hard to measure and create precise pressure due to belt mechanism used (mathematical representation is not initially clear) • Mostly manual, might be difficult to automate
Silicone Ring System (HemaClear)	• Cheap • Proof of concept already exists • Reduced chances of injury • Components are easy to find	• Hard to measure and create precise pressure due to manual operation • Rides up the limb and covers a lot of surface area • Disposable, can only be used once
Tightening Rod	• Cheap • Components are easy to find • Proof of concept already exists	• Hard to measure and create precise pressure due to belt mechanism used (mathematical representation is not initially clear) • Mostly manual, might be difficult to automate
Suction and Pump	• Cheap • Components are easy to find • Proof of concept already exists	• Hard to measure and create precise pressure due to zip mechanism used (mathematical representation is not initially clear) • Mostly manual, might be difficult to automate • Disposable, can only be used once • Small size, no proof of concept for

		operation on thigh
Motor controlled cuff	• Cheap • Components are easy to find • Easy to control using motor drivers	• Ergonomically unstable, there would have to be a wire in place to correctly control motor function • Pressure would be focused on a specific point instead uniform through the cuff

3.1.4.1 Motor Driver

The actuation process is facilitated by the circuit depicted in Figure 1. Power for the motor driver is supplied by the 7.4V device battery, with pin Vm serving as the connection point. Pin Vcp is also linked to VBUS via a 0.01 uF, 16 V X7R ceramic capacitor, as specified in the datasheet. Vint, the chip's internal bypass supply, is connected to GND through a 2.2 uF capacitor. AISEN and BISEN pins regulate current chopping for the motors driven. However, given that the motor in use has a maximum current of 380 mA under load, these pins are directly connected to GND. The nSleep pin, connected to a GPIO pin on the ESP32, instructs the motor driver to enter sleep mode when set to LOW. Similarly, the nFault pin, also linked to a GPIO pin on the ESP32, indicates a failure state when outputting a logical LOW due to factors such as undervolting or overheating. The AIN and BIN pins manage their respective motors, with the motor driver adjusting the operation of the motors connected to AOUT and BOUT based on the logical HIGH or LOW input into these pins. The truth table outlining these logical inputs is detailed in Table 2. Since both motors are connected to a single driver, the combined voltage drop of the actuation system is approximately 0.138 V. ($V_{drop} = 0.36\,ohms * 0.38\,A$) [18].

The solenoid valve utilized in this circuit has a rated current consumption under load of 240 mA ± 15%, while the DC pump, as mentioned earlier, has a maximum current consumption of 380 mA under load. The DRV8833PW motor driver IC exhibits a current consumption of 1.7-3 mA and can be transitioned into low power sleep mode through the sleep pin when not in use, thereby reducing current consumption to microamps [18].

Table 2. Logical Control of the DRV8833PW Motor Driver

xIN1	xIN2	FUNCTION
PWM	0	Forward PWM, fast decay
1	PWM	Forward PWM, slow decay
0	PWM	Reverse PWM, fast decay
PWM	1	Reverse PWM, slow decay

3.1.4.2 Actuation

The integrated circuit (IC) serves a crucial role in driving both a DC pump and a solenoid valve within the system, responsible for regulating the air pressure. The airflow is directed into a cuff, resembling those used in blood pressure applications. As the cuff inflates, the pressure

exerted on the user's arm gradually increases. Once the pressure reaches the desired value for venipuncture, the cuff ceases inflation and maintains the set pressure. In scenarios where the system aims to halt blood flow in the limb, the cuff continues to inflate until there is no detectable pulse in the individual's arm. Subsequently, the pressure is slowly released through the valve until reaching the precise inflection point between blood flow and no blood flow. This methodology mirrors the technique employed by blood pressure monitors to determine both systolic and diastolic pressure. The setup involves a piping system connecting the solenoid valve to a T-joint, which in turn is linked to piping connecting the motor and pressure sensor together [16]. This configuration enables seamless coordination between the various components, ensuring precise control over air pressure regulation within the cuff.

3.1.6 Screen Interfacing

To provide users with feedback on the status of the board, a 320x280 TFT LCD Display with an ILI9341 display driver has been selected. This screen is connected to the microcontroller via an SPI interface, utilizing the hardware SPI bus for the fastest connection speed.

In this setup, six different data lines are interfaced in the circuit:

1. SCK (SPI Clock): Responsible for synchronizing data transmission between the microcontroller and the display.
2. CS (SPI Chip Select): Used to select the display as the target device for communication.
3. MOSI (SPI Master Out Slave In): Transmits data from the microcontroller to the display.
4. DC (Screen Data Control): Determines whether data sent over SPI is interpreted as command or display data.
5. RESET (Screen Reset): Resets the display to its default state if needed.
6. BL (Screen Backlight) and VCC (3V3 in): Provide power to the screen and control the backlighting.

All these pins are directly connected to the microcontroller, facilitating seamless communication between the two components. The choice of this screen was primarily influenced by its size and dimensions, while its touchscreen capability, although not utilized in this project, adds versatility for future applications [19].

3.1.7 Overall Circuit Design

All of these circuits come together to form the full circuit design for the DH^3 automated tourniquet system. This design was created on KiCad using symbol models from Digikey. There are also buttons and LEDs connected directly to the ESP32-S3 microcontroller to provide options for user input and give user feedback on the power and status of the device. Additionally breakout headers are added for direct communication with the ESP32-S3 through UART as well as an expansion header for PPG sensor input (to detect total limb occlusion). Figure 1 is the detailed schematic.

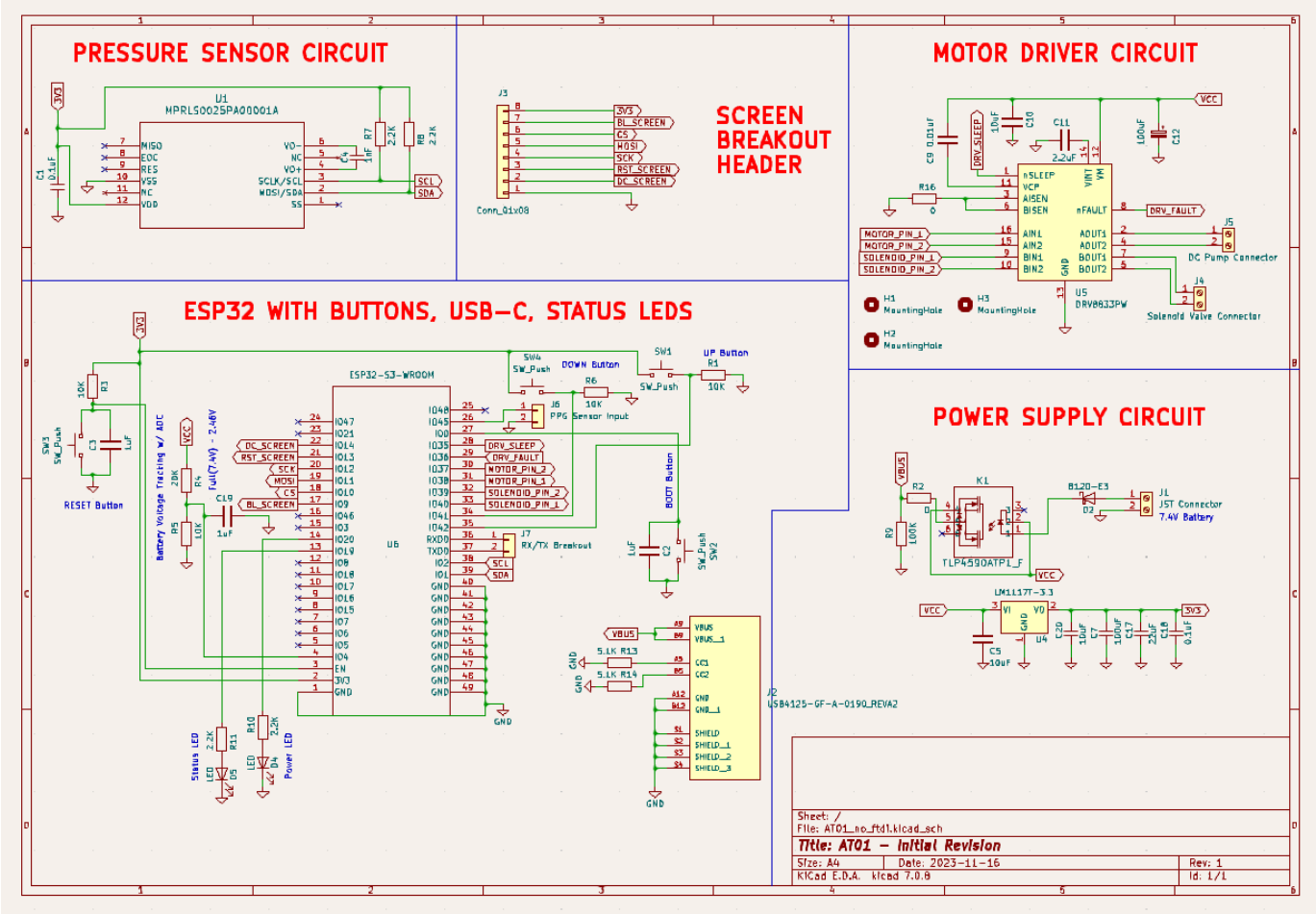


Figure 1. Overall Circuit Design

3.2 Housing Design

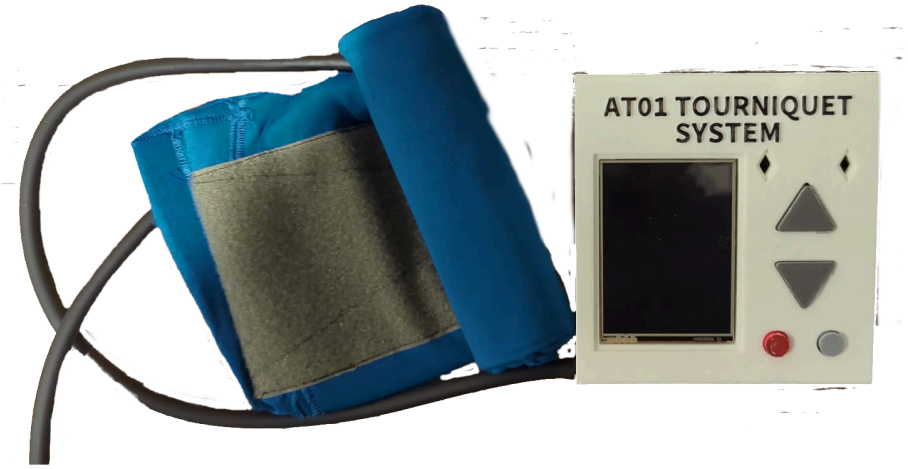


Figure 2. Overall Fabricated Housing Design

The figure above shows the cuff, the housing and the tubing in the final design.

3.2.1 Device Housing Design

In the initial stages of housing design for the device, the primary objective was to achieve stability, initially pursued through a triangular shape. However, upon thorough evaluation, a flatter semi-pyramidal design emerged as the more practical choice. This design offers several advantages, including enhanced resistance against tipping, which is critical for stability. By adopting a flatter semi-pyramidal shape, the device maintains stability while also ensuring a compact and manageable structure. This design evolution ultimately enhances the usability and portability of the device, making it more suitable for various settings and applications.

The design of the display was meticulously integrated with the user interface plan to deliver a seamless and user-friendly experience. A dedicated section for the screen was seamlessly incorporated into the design, complemented by four buttons—power, reset, and up/down selection—ensuring intuitive usability and simplified input for users. Furthermore, a cover was integrated into the design to display most directions on the screen, offering users clear guidance and enhancing usability. To provide users with real-time feedback, a transparent opening was incorporated into the cover, allowing for visibility of error and power indicators represented by red and green lights located in the top right corner of Figure 4. This thoughtful design approach ensures that users can effortlessly monitor the status of the device at a glance, thereby enhancing overall user experience and operational efficiency.

Figure 3. 3D Modeled Design

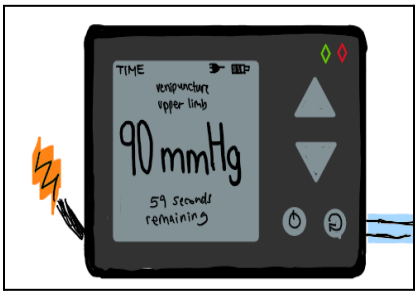


Figure 4. Device Cover Design

Figure 5. 3D Printed Design

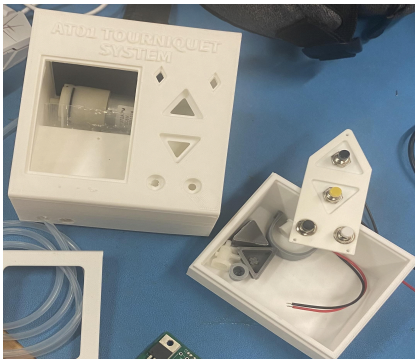


Figure 6. 3D Printed Design with all accompanying pieces/fittings

The design of the casing and cover itself were modified to accommodate for the available screen/display to be used which has a dimension of 4.3x5.7cm as can be seen in the following Figure 7.

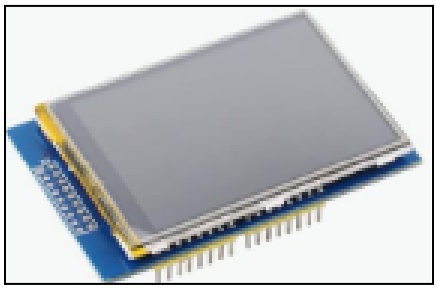


Figure 7. 4.3x5.7 cm Display

In the materials analysis phase of the design process, several options were considered, including High and Low-Density Polyethylene, Polycarbonate (PC), Polylactic Acid (PLA), and Polypropylene (PP). All these materials are FDA-approved 3D-printable plastics, offering ease of sterilization using high-energy radiation tools and chemicals. However, after careful consideration, Polylactic Acid (PLA) emerged as the optimal choice for the housing design. PLA was chosen for its transparency, durability, and widespread use in medical-grade products. Additionally, it's easy accessibility and environmental benefits contributed to its selection [20][21]. In contrast, other materials considered were comparatively softer and weaker, making them more susceptible to corrosion and unsuitable for meeting the device's portability requirement, thus potentially increasing the risk of damage [22].

Moreover, the final device design underwent rigorous testing through the SUS (System Usability Scale) scoring process [40], spanning three different phases. These phases included initial testing with a diverse group of users, followed by iterative refinements based on feedback, and culminating in final end-user testing with healthcare professionals. Each phase of testing provided invaluable insights and feedback, allowing for continuous refinement of the housing design to ensure optimal usability and user satisfaction. Through this iterative process, any potential usability issues were identified and addressed, leading to improvements in the overall design and functionality of the device.

By engaging with users from various backgrounds and expertise levels, including healthcare professionals, the final end-user testing phase ensured that the device met the specific needs and requirements of its intended users. This comprehensive approach to testing and refinement underscores the commitment to delivering a product that not only meets but exceeds user expectations, ultimately enhancing its usability and effectiveness in real-world scenarios.

3.2.2 Inflatable Cuff Design

When designing the cuff for the automated tourniquet, numerous critical factors were meticulously considered, placing particular emphasis on the incorporation of a bladder. Traditionally crafted from impermeable latex, the bladder serves as the conduit for inflation to uphold pressure. Extensive research has highlighted a distinct preference for rubber/latex bladder cuffs, showcasing minimal differences of merely 5 mmHg [23]. Recognizing the paramount importance of uniform pressure distribution for accurate readings and the direct influence of blood pressure measurement on the tourniquet’s efficacy, the decision to include a bladder was imperative. This choice ensures the precise maintenance of venipuncture and full occlusion pressures at 60 mmHg and 250 mmHg, respectively [24][25].

Following the decision to incorporate the bladder, the choice to use a 4-layer fabric covering it was confirmed. Previous studies supported this layered approach to protect patients from direct bladder contact. Among the fabric options considered—broadcloth, nylon, polyester, and vinyl—nylon emerged as the most effective choice, demonstrating minimal creasing and wrinkling, which is crucial for maintaining shape integrity during cuff inflation [25]. Despite nylon's woven structure, which may be less optimal for pressure distribution, a blend of nylon and polyester was selected for the cuff [25], with a 100% nylon inner outward facing lining to decrease instances of bulging and ensure even pressure distribution. These fabric choices and blends meet regulatory standards, are reusable, sterilizable, durable, and non-absorbent—a crucial characteristic in medical environments. Material elasticity is essential for patient safety and injury prevention [26], aligning with the design philosophy observed in other cuffs such as the Phillips blood pressure monitor cuff [27]. Regardless of the cuff material, it's crucial to emphasize the necessity of a protective cover, such as cast padding, to shield patients from pinching effects. Although cuffs with integrated gel padding were considered to address this issue and eliminate the need for cast padding, they were found to have a more adverse impact on the patient's skin [27].

The shape of the cuff itself was determined through the study of various sources, all of which highlighted a direct correlation between accurate blood pressure measurement and cuff width, as illustrated in the following figures.

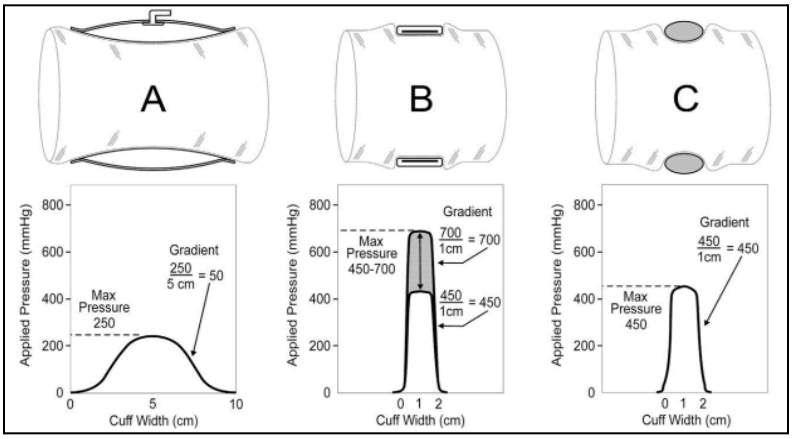


Figure 8. Cuff Width Pressure Gradient

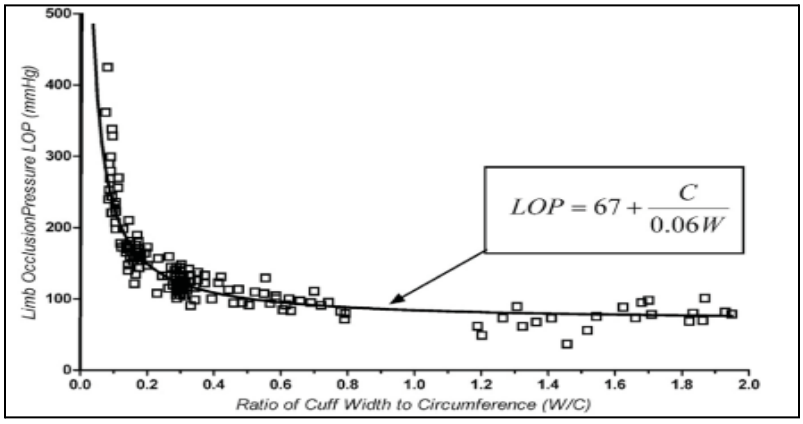


Figure 9. Role of Cuff Width on LOP

The left figure underscores the significance of the pressure gradient across different cuff inflation widths, highlighting the crucial relationship between cuff width and applied pressure. Conversely, the right curve illustrates how increased patient width correlates with decreased applied pressure. This correlation is pivotal for mitigating the risk of patient injury in both surgical and venipuncture tourniquet applications [28]. Studies have corroborated these findings by demonstrating that narrower cuff sizes tend to induce more pain compared to wider cuff widths [29]. Therefore, the initial cuff width was established at 6.5 inches, a determination informed by user studies and research conducted by North Carolina State University on mean human limb lengths [30].

In terms of cuff shape, early designs featured a rudimentary, rectangular shape (Figure 9), proven effective in applying even pressure according to studies. However, future considerations lean towards a curved cuff design (Figure 10), especially suitable for individuals with larger limbs or obesity [30].

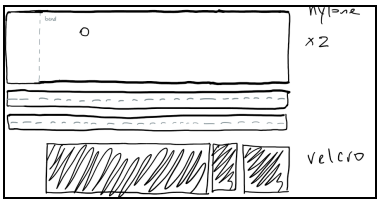


Figure 10. Rough Rectangular Cuff Design

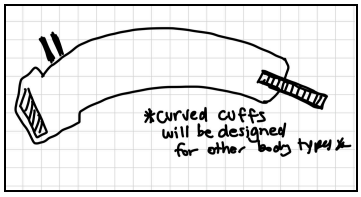


Figure 11. Rough Curved Design

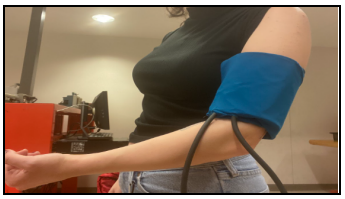


Figure 12. Fabricated Cuff Design

To assess the usability of the design, feedback was gathered from a group of 5 testers during the initial stages of the project. Research suggests that for smaller projects like this one, a smaller focus group is appropriate. However, as physical testing progresses, a larger group becomes more advantageous. Studies indicate that with approximately 5 testers, around 86% of design problems can be identified. This percentage tends to increase with more testers, plateauing at around 20-50 testers [31]. To align with these findings, the number of testers was expanded, with a focus on end users. The total results from all testing phases are reflected in the figure below. The System Usability Scale (SUS) Questions yielded an overall SUS score of 89.67%, indicating a high level of usability for the design. Additionally, color preferences for the cuff and device were explored during this testing phase, further enhancing user feedback and satisfaction. The SUS questions can be more closely observed in the appendix section 11.1.1.

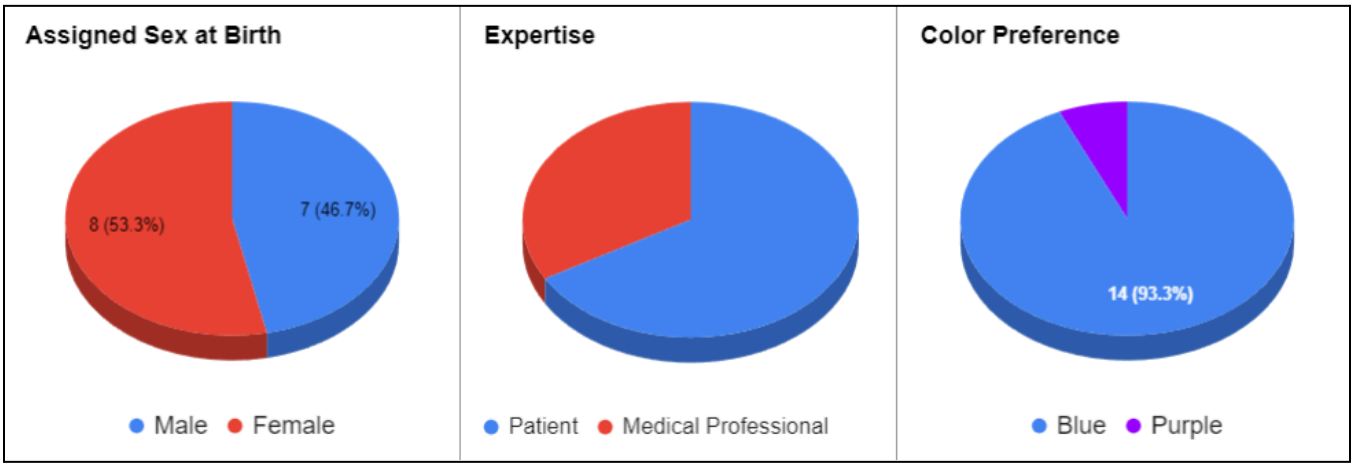


Figure 13a. System Usability Score Rating Outcomes

Age Range	19 - 62 yrs old
Height Range	5'3" - 6'2"
Weight Range	110 lbs - 200 lbs
Arm Circumference Range	25 cm - 40 cm

Figure 13b. Demographic Data

Mean SUS Score (/100)
89.66666667

Figure 13c. Mean SUS Score

End-user testing was conducted to gather feedback on the final device from an active healthcare practitioner. The average SUS score obtained from this testing phase was 95%, indicating a high level of usability and user satisfaction with the design. However, a few issues were observed during testing, particularly regarding the pneumatic ports and screen readability. These issues were addressed for improvement, with text size and icons being increased and pneumatic ports being resized and fitted.

Figure 14. (a-c) Images of End User (Registered Nurse) Testing Venipuncture on Patient

The feedback received from end users provides valuable insights for further refinement and enhancement of the device. By addressing issues such as pneumatic port functionality and screen readability, the overall usability and effectiveness of the device can be optimized, ensuring a seamless user experience in clinical settings.

3.3 Pressure Sensing Design

3.3.1 Technical Criteria

The pressure sensing component plays a critical role in accurately measuring the tourniquet's pressure on the patient's arm to achieve optimal vein engorgement and complete occlusion of arterial blood flow. This necessitates the selection, integration, and calibration of a suitable pressure sensor capable of detecting subtle pressure changes and enabling real-time precise control. To guide the selection process, criteria were established based on the automated tourniquet's intended applications, conditions, and settings.

These criteria include:

1. High sensitivity and accuracy to detect subtle changes in pressure and enable precise real-time control.
2. Easy integration with minimal circuit and design complexity.
3. Wide applicability across various medical scenarios.
4. Pressure range sufficient for both venipuncture and limb occlusion procedures.
5. Cost-effectiveness to facilitate broad adoption.
6. Low supply voltage to ensure compatibility with the battery and efficient power consumption.

Based on the findings from the literature review, it was noted that blood pressure changes typically occur at low frequencies (~0.5Hz). Therefore, the pressure sensor must possess a high response time and a low-frequency response to process these signals without attenuation, ensuring quick and accurate responses [32]. Additionally, it is crucial for the pressure sensor to provide an easily interfaced electrical output for control systems, preferably in digital format to facilitate accurate data processing with the microcontroller. Considering the recommended tourniquet pressures for venipuncture (~60 mmHg) and limb occlusion (~250 mmHg) [24][25], a pressure sensor with a low supply voltage is desirable to ensure compatibility with the battery and promote efficient power consumption. These criteria guide the selection process, ensuring that the chosen pressure sensor meets the specific requirements of the automated tourniquet design.

3.3.2 Pressure Sensor Comparison

Four pressure sensor types were considered: piezoresistive, capacitive, optical, and strain gauge. While capacitive sensors offer high accuracy with low power consumption, they require complex signal conditioning circuits and calibration algorithms [33]. Optical sensors, recognized for accuracy and sensitivity, are costly and demand a light source and detector, potentially complicating integration in an automated tourniquet [34]. Strain gauge sensors are less sensitive and more prone to signal noise [35]. Piezoresistive sensors, known for high sensitivity and accuracy, are compact, flexible, and easy to integrate due to simple circuit design, offering a versatile electrical output for control system interfacing [32]. Additionally, they are cost-effective and available for various pressure ranges. Considering these factors, the piezoresistive sensor stands out as the most suitable choice for the automated tourniquet system.

3.4 User Interface Design

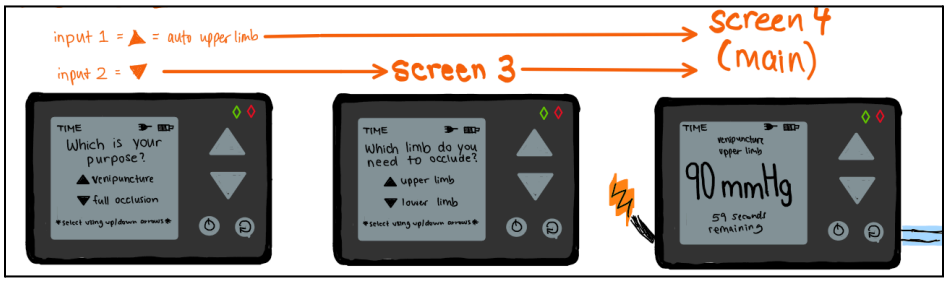


Figure 15. User Interface Logic.

Figure 16 (a-c). Layout of various pages on screen. **a)** Main menu layout. **b) d)** Occlusion/ limb selection page layout. **c)**. Venepuncture pressure and time display page.

The figures above depict the general layout and control flow of the screen.

This section discusses the development of the user interface (UI), specifically highlighting the rationale behind the modality and discussing the functions available to the user.

3.4.1 Purpose

3.4.1.1 Significance of User Interface

The objective of the UI is to enable the user to interact with and control the device in a reliable manner and see the results of usage in real time.

3.4.1.2 Comparison of UI Modalities

Three modalities of user interface were considered, namely: physical screen, web server and mobile application. These are abbreviated as: M (manual, i.e. physical screen), WS (web server), and MA (mobile application). There were 4 main parameters that were chosen to determine the final choice of UI, namely: (1) cost to build and maintain, (2) customizability, (3) security and (4) support for multiple users.

Table 3: A review of all considered user interfaces

User Interface Type	Pros	Cons
Web Server/ Mobile Application	• support for multiple users and roles • customizable via programming	• potential security vulnerabilities • higher cost for maintenance
Manual	• lower cost for maintenance • no potential security vulnerabilities	• lower support for multiple users and roles • lower customization due to hardware

When the parameters were considered for the 3 modalities in the context of our device, the following were the conclusions, which is a summary of Table 3 above.

Parameter Considerations

- Cost to build/ maintain (M > WS/MA)
- Customizability (WS/MA = M)
- Security (M > WS/MA)
- Support for multiple users (WS/MA > M)

As can be seen, the physical screen wins 2 of the 4 criteria, and is equal to the other options in a 3rd, making it the clear choice. As such, for the final device, the physical screen UI method was chosen.

3.4.1.3 Target User Base

In this project, the choice of platform was heavily influenced by the identified target audience, with a primary focus on ensuring ease of use. Recognizing that the target audience comprises nurses and physicians, conducting thorough market analysis became essential to determine the most accessible platform for end-users. The selected target audience for this project falls within the age range of mid-20s to mid-60s. This demographic range was chosen based on several factors. Firstly, individuals in their mid-20s typically begin their nursing career after completing their education. On the other hand, doctors usually start their careers later, typically in their early to mid-30s, due to the longer education and training required. Additionally, the upper limit of the age range, mid-60s, corresponds to the typical retirement age for nurses, doctors, and many other professionals. Considering this age range ensures that the platform caters to both early-career professionals who are accustomed to newer technologies and older professionals who may be more familiar with traditional systems. By aligning the platform choice with the demographics and preferences of the target audience, the project aims to ensure maximum accessibility and usability for nurses and physicians across different career stages. This approach enhances user satisfaction and facilitates seamless integration into existing workflows, ultimately improving the overall effectiveness of the device in clinical settings.

Our user base is comfortable with the usage of a physical screen with buttons for navigation, as per our market research. This is due to the fact that medical professionals use many similar devices, such as patient monitors (monitors vital signs such as heart rate, blood

pressure, oxygen saturation and temperature), ventilators (display respiratory parameters and alarms) and defibrillators (display ECG waveforms).

3.4.2 Frontend Design

3.4.2.1 Microcontroller Setup and Integration

The microcontroller serves as the core component of the system, responsible for controlling the tourniquet mechanism and interfacing with peripheral devices. Through meticulous hardware design and programming, the microcontroller ensures precise and timely execution of treatment protocols. Additionally, to provide the user with engaging, informative visuals, the system UI is built from scratch using the LVGL (Light and Versatile Graphics Library) display library and the TFT eSPI display driver library. These libraries are directly compatible with both the ESP-IDF and Arduino frameworks of the ESP32 and as such can be easily implemented to produce visual output. The library itself uses labels, external button feedback, timer interrupts, active screens and various design elements such as arcs and squares to create user-friendly interfaces that can be operated by the user with a very small learning curve. The LVGL library is written in C and the tourniquet code is written in C++ that is then flashed onto the microcontroller using the Arduino framework. The development library used for this is PlatformIO built on Visual Studio Code due to its professional development interface, easy access to microcontroller communication interfaces (such as the serial monitor) and its compatibility with custom board designs.

To ensure readability the code is split into various functions that depict each of the control screens the user has to go through. The code starts with the homeScreen() function after all input and display drivers are set up. From the home screen depending on which external button (up or down) is pressed, the user can either choose venepuncture or occlusion mode. If the user chooses venepuncture the code immediately goes into the actuationScreen() function where a large dial shows the current pressure with a timer and target pressure at the bottom of the screen. If the user chooses the occlusion setting then another menu asking the user to choose either bicep or thigh occlusion shows up on the screen after which the user can choose either mode and enter the actuation screen. The code developed for this system can be found in Appendix 11.2.2.

3.4.2.2 Peripheral Device Integration

Integration of peripheral devices, such as the TFT screen, is crucial for providing a comprehensive UI. The TFT screen displays vital information and allows users to interact with the system, facilitating intuitive control and monitoring of treatment parameters. The TFT screen is interfaced with the microcontroller using the TFT eSPI display software driver built by Bodmer. The TFT eSPI display driver directly interfaces with the ILI9341 driver by controlling the internal registers. In the code seen in Appendix 11.2.2, the MADCTRL register of the ILI9341 uses default values to initialize the screen in all cases other than for screen rotation which has been edited to physically rotate the screen by 90 degrees. The TFT eSPI library uses an initializer header file to customize fonts being used and hardware connections. In this case the

BL, MOSI, DC, RST, SCK and CS pins are customized from the defaults to use the hardware SPI and utilize the fastest SPI bus speeds rather than interfacing with the slower software SPI bus.

3.4.3 User Manual

3.4.3.1 Operation Instructions

The following gives simple operating instructions for successful usage of the device.

Step 1: Power On: Press the power button to activate the system.

Step 2: Mode/ Limb Selection: Use the navigation buttons (UP/DOWN) to select the desired treatment mode and limb.

Step 3: Start Treatment: The process will be started after selection of mode and limb.

Step 4: Parameter Adjustment: Adjust treatment parameters, namely pressure and duration, using the navigation buttons (UP/DOWN).

Step 5: Monitoring: Monitor treatment progress and adjust settings as necessary during the session.

Step 6: Power Off: Turn off the system once procedure is complete by pressing the power button again, or reset button.

3.4.3.2 Troubleshooting Guide for Hardware Components

Some common hardware issues are given below, along with their suggested solutions.

Screen Display Issues: Ensure proper connection of the TFT screen to the microcontroller and check for any loose connections. The front cover of the device will need to be opened.

Control Button Malfunction: Verify the integrity of control buttons (UP/DOWN/POWER/RESET) and check for any debris or obstruction affecting their operation. Do not use water to clean the device, a dry cloth should be used to ensure no electrical malfunction.

Power Supply Problems: Confirm adequate power supply to the system and check for any issues with power cables or connectors. To access the battery, the device's front cover needs to be opened.

Tourniquet Mechanism Failure: Inspect the tourniquet mechanism for any mechanical faults or blockages to tubing and address them accordingly. To access the tubing, the device's front cover needs to be opened.

3.5 Control System Design

3.5.1 Need for PID Control

In our context, which is design of an automated tourniquet system which achieves and maintains specific pressure levels using a modified blood pressure cuff, the incorporation of a PID control system is needed for 4 main reasons:

3.5.1.1 Enhanced Precision and Accuracy

Leveraging a PID controller facilitates meticulous and accurate regulation of the pressure applied by the tourniquet cuff. This is particularly crucial in medical scenarios where maintaining precise pressure levels is imperative for both safety and optimal functionality.

3.5.1.2 Dynamic Responsiveness

The PID control algorithm demonstrates agility in responding to fluctuations in pressure requirements, a vital attribute in medical settings where variations in patient physiology or external conditions can impact the needed pressure.

3.5.1.3 Adaptability Over Time

The Integral (I) component of the PID controller plays a pivotal role in mitigating long-term steady-state errors. This feature proves beneficial in medical applications, where external factors or evolving patient conditions may influence the required pressure levels over extended periods.

3.5.1.4 Stability Enhancement

The Derivative (D) component of the PID controller aids in dampening abrupt changes and overshooting, ensuring that the system achieves and sustains the desired pressure without succumbing to oscillations or instability.

3.5.2 Breakdown of PID Controller Components

3.5.2.1 Proportional (P)

This element responds to the current error, effectively addressing discrepancies between the desired and actual pressure. Adjusting the proportional gain allows for varying degrees of corrective action in response to different error magnitudes.

3.5.2.2 Integral (I)

The integral component accumulates past errors and rectifies any persistent steady-state discrepancies. Its role is instrumental in guaranteeing that the system ultimately reaches and maintains the specified pressure level.

3.5.2.3 Derivative (D)

This component anticipates future errors by considering the rate of change in the error. Its function is to prevent overshooting and oscillations, thereby enhancing the overall stability of the system.

In summary, the integration of a PID control system in an automated tourniquet setup facilitates meticulous, stable control over cuff pressure, ensuring both patient safety and the efficacy of the medical procedure.

3.5.3 Explanation of Control System Code

The control system code has been provided in the appendix (11.1.1), a brief explanation is given below. For further details, refer to the comments in the code itself.

The PID library for Arduino, often referred to as "PID Library" by Brett Beauregard, is a widely used software resource that simplifies the implementation of a PID (Proportional-Integral-Derivative) controller in Arduino projects. This library is particularly useful for managing tasks that require precise control, such as adjusting motor speeds as was the case with the automated tourniquet.

3.5.3.1 Key Variables

Several variables and constants for the operation of the PID control were defined. The “Input” to the PID controller was the current measurement used by the PID to compute the error. This was the actual value we were trying to control, which is current pressure as measured by a sensor. The “Output” was set to represent the value that the PID would calculate based on the error between the “Setpoint” and the “Input”. This Output was the control variable to adjust motor speed control for maintaining pressure to reach the desired Setpoint. The “Setpoint” was the desired pressure value that we wanted the system to maintain. Kp, Ki, and Kd were the tuning parameters for the PID controller, and each played a unique role in how the PID adjusted the “Output” as mentioned above in 3.6.2.

3.5.3.2 Setup Function

The PID controller was initialized and configured within our setup() function. The mode was set to automatic to turn the PID on and define the output limits based on our application. The myPID.SetOutputLimits(min, max) function in the PID library is used to define the range of possible output values that the PID controller can produce. We set min, the lowest value that the PID controller was allowed to output as 0. We set the max, the highest value the PID controller could output as 255 which is the typical standard max value for PWM-controlled devices. myPID.Compute() computes the new PID output value which is then applied to the control variable which writes to the motor and adjusts its speed.

3.5.3.3: Loop Function

Within the loop() function, the lvgl_timer_handler() function which runs all of the timers and consistently checks for the occurrence of external button interrupts is continuously called. This handler makes sure the screens are being correctly updated when certain events occur and the control system flow is correctly followed..

3.5.4 Determination of PID gain constants

To determine the PID gain constants, there are many methods such as the Cohen-Coon method, manual trial and error and the Tyreus-Luyben methods. The Ziegler-Nichols (ZN) method is used for this system, as it is widely used in industry and academia. In order to employ this method, a few assumptions are critical. Firstly, the system is in a closed loop control configuration. Secondly, there is a negative feedback loop. There are 5 main steps of the ZN method, which are outlined below:

Step 1: Make Ki and Kd 0 such that the controller is only a proportional controller (Kp ≠ 0).

Step 2: Apply unit step input.

Step 3: Change the values of Kp until it reaches Ku (critical point of Kp at which they are stable. oscillations of output, similar to a sine wave).

Step 4: Measure Pu (average period over several periods at Ku).

Step 5: Substitute into following equations (Figure 14) for the gain values in different control systems.

Controller	K_p	K_i	k_d
P	$K_u/2$	-	-
PI	$K_u/2.2$	$P_u/1.2$	-
PID	$K_u/1.7$	$P_u/2$	$P_u/8$

Figure 15. Ziegler-Nichols settings for PID control tuning [28]

The PID gains were determined in the above method and incorporated into the code, which is provided in the appendix (11.2.1 Control System Testing Code).

3.6 System Integration

The hardware of the design was implemented onto a 75mm x 55mm PCB. This PCB was designed in KiCad and was manufactured as a 2 layer board. All trace widths were chosen using the standard trace calculation equation ($W = \frac{A}{t*1.378}$ where $A = (\frac{I}{k*Trise^{b}})^{\frac{1}{c}}$) [14]. All trace lengths are a minimum of 0.3 mm wide and are capable of carrying 500 mA of current. Any power traces are a minimum of 0.52 mm wide and are capable of carrying 750 mA of current. All 3V3 or 5V power line traces are a minimum of 1 wide and are capable of safely carrying up to 1.2 A of current. The 3V3 line being generated by the voltage regulator has decoupling capacitors around both the regulator and the near the inputs of all 3V3 input power pads to prevent an oscillating power signal from interrupting the operation of the ICs. The ground plane has been poured on both layers of the PCB for thermal relief and there are ground vias all around the board to improve the signal conditioning and remove interference. Additionally, all footprints are chosen with thermal conductivity in mind. Specifically, the LM1117-3.3V LDO uses a TL-220 footprint to improve the thermal performance of the IC and keep it within the recommended operating conditions of the IC (maximum of 125 C) [14]. Additionally, the total current consumption of the board is measured at approximately 700 mA peak and trace widths are designed with that number in mind. All high current carrying traces are wider than standard widths and the PCB was designed with zero DRC violations and with larger than the minimum constraints to ensure there are no problems during manufacturing.

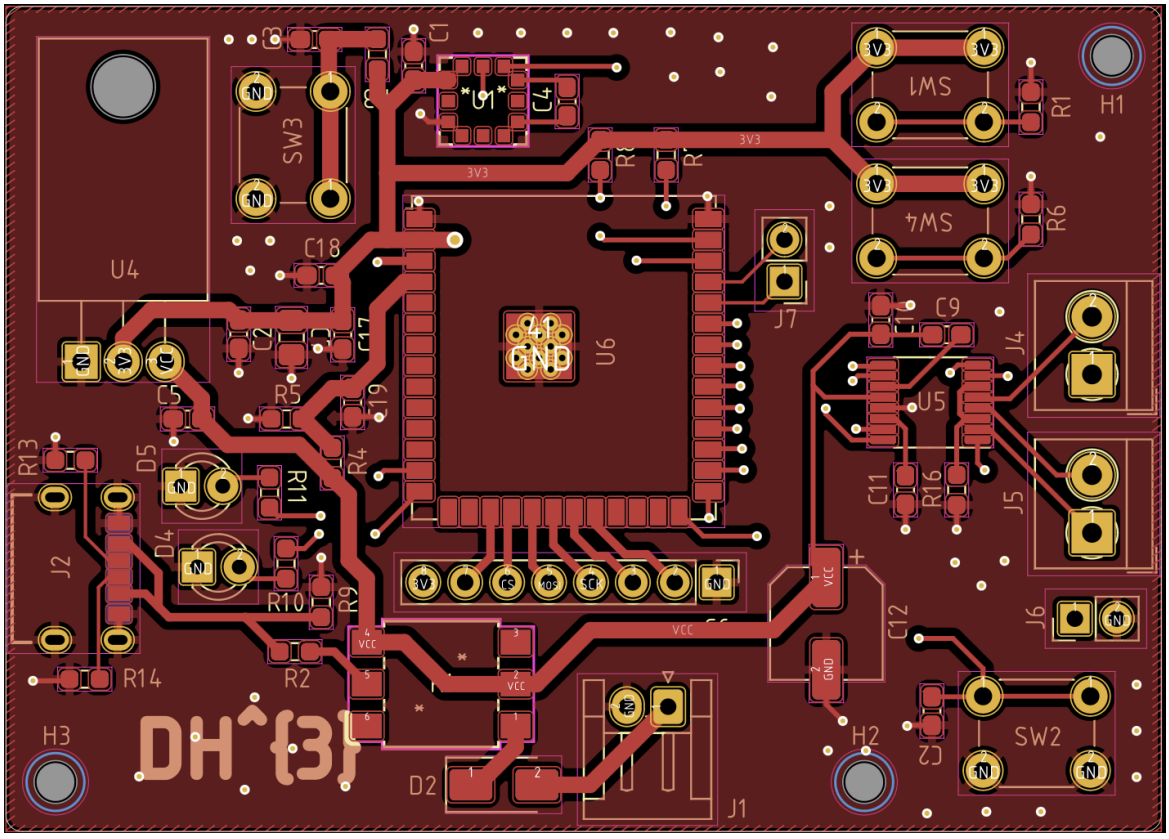


Figure 17: PCB Design of the DH^3 Automated Tourniquet

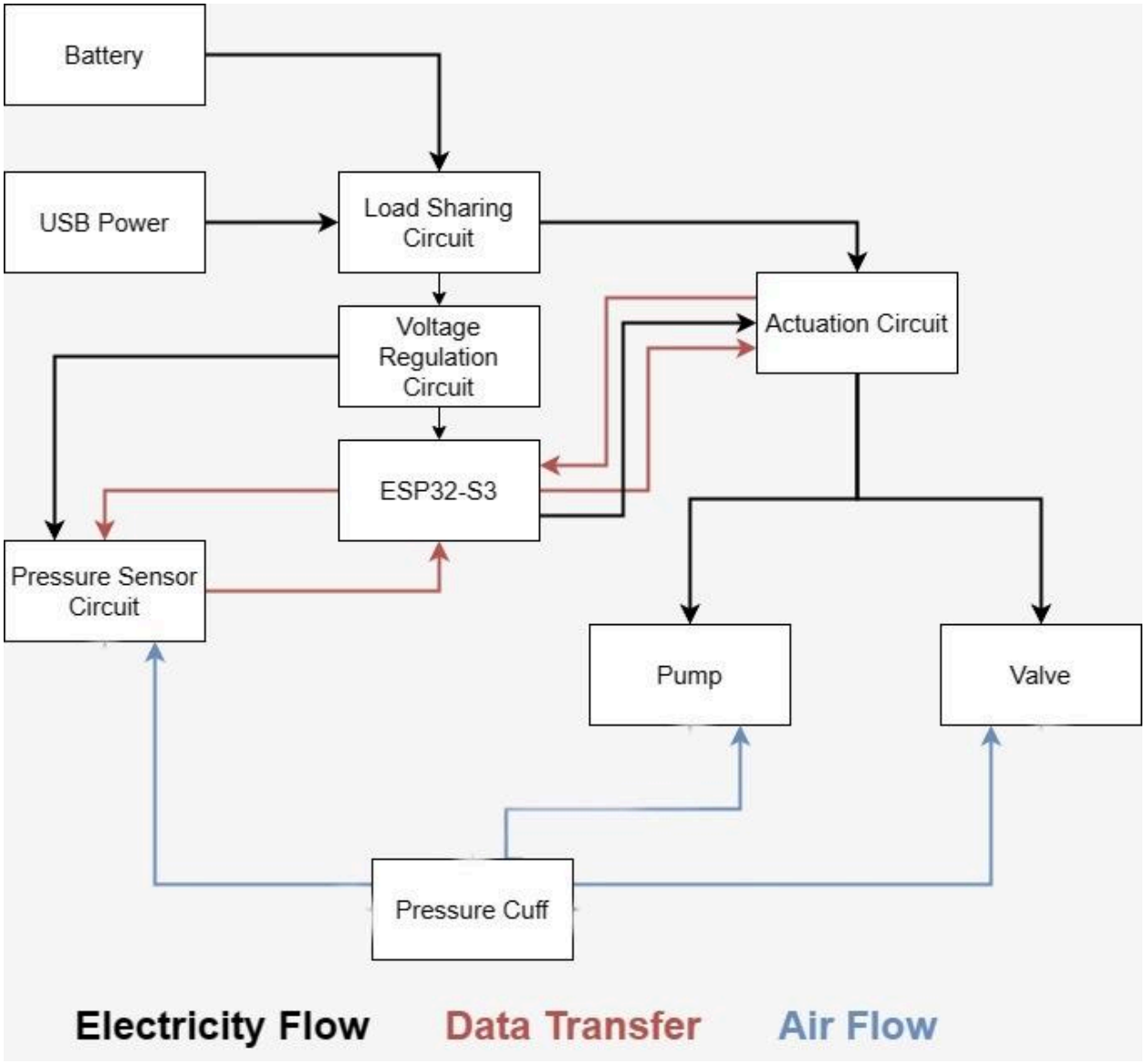


Figure 18: Full System Integration Diagram of the DH[3] Automated Tourniquet

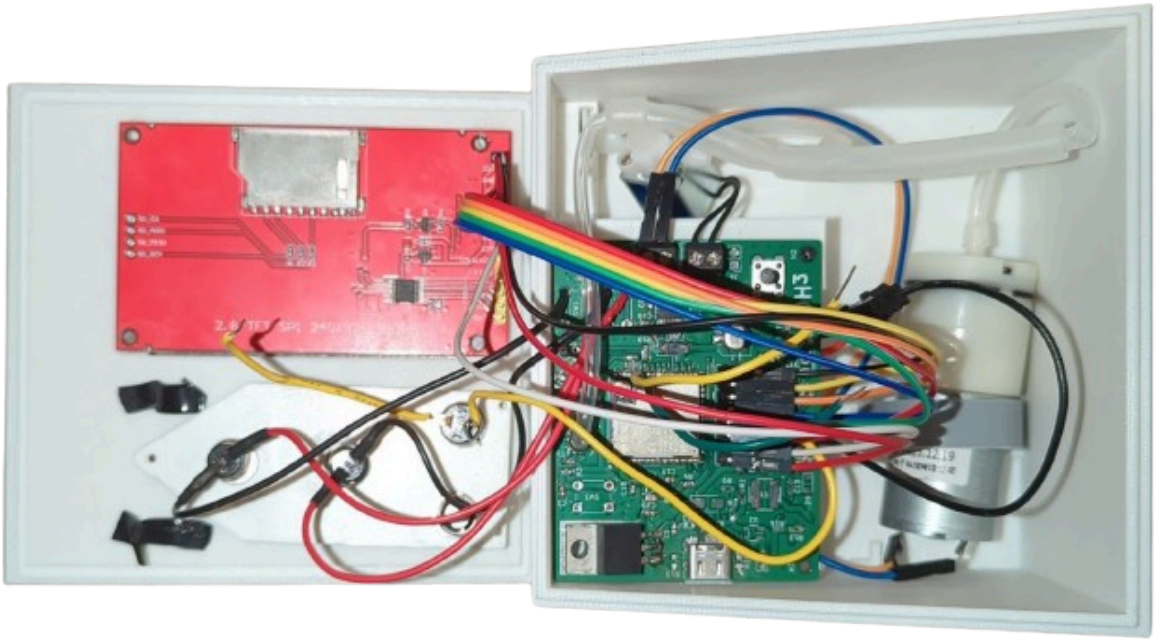


Figure 19: Full DH[3] Automated Tourniquet System

4.0 Alternative Designs

Alternative design solutions were considered at every aspect of the design and a comparative analysis was performed between them, sometimes qualitatively and sometimes quantitatively.The details were mentioned in the design section. The following directs the reader to the appropriate sections.

4.1 Microcontroller

Alternative microcontrollers, such as the Nordic Semiconductor-based NRF7001, were considered. Refer to Section “3.1.1 Microcontroller” for further details.

4.2 Microcontroller Reset Mechanism

Alternative reset mechanisms, such as software-based resets, were considered. Refer to Section “3.1.1.3 MCU Operation” for further details.

4.3 Power Supply and Voltage Regulatory Mechanisms

As an alternative, the AMS1117-3.3V LDO was considered as a power supply and voltage regulatory mechanism. Refer to Section “3.1.2 Power Supply and Voltage Regulation” for further details.

4.4 Casing

For the materials analysis in this design area, various options were considered including High and Low-Density Polyethylene and Polypropylene (PP). Refer to Section “3.2.1 Device Housing Design” for further details.

4.5 Inflatable Cuff

Cuffs with integrated gel padding were considered. Refer to Section “3.2.2 Inflatable Cuff Design” for further details.

4.6 Pressure Sensor

Three alternative pressure sensor types were considered: capacitive, optical, and strain gauge. Refer to Section “3.3.2 Pressure Sensor Comparison” for further details.

4.7 User Interface

Two alternative user interface modalities were considered: web server and mobile application. Refer to Section “3.5.1.2 Comparison of UI Modalities” for further details.

4.8 Control System

Five alternative control system structures were considered: P, I, D, PI, and PD. Refer to Section “3.6.1 Need for PID Control” and “3.6.2 Breakdown of PID Controller Components” for further details.

4.9 Display

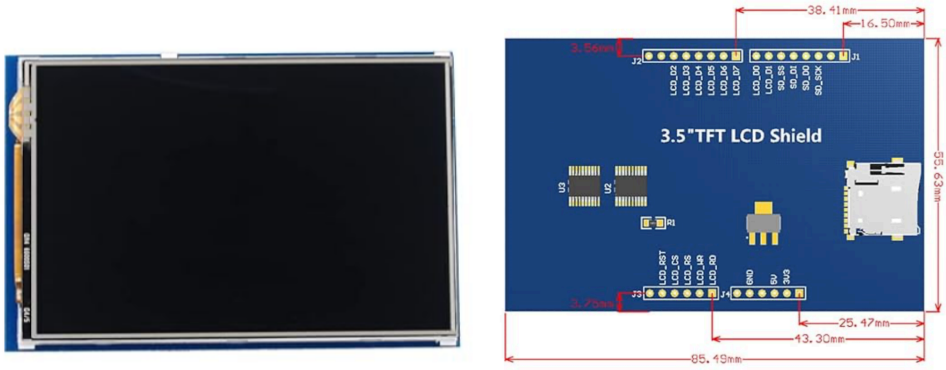


Figure 20: a) Physical Alternative Screen. **b)** Wiring Diagram for Alternative Screen

This alternative screen “3.5" TFT LCD Display Module ILI9486 Resistive Touchscreen 5V/3.3V 480x320 8-bit Parallel Interface”, as shown above in Figure 18 was also considered. Refer to “3.1.6 Screen Interfacing” for further details.

5.0 Material/Component List

5.1 Supplementary Figures, Diagrams & Tables

Table 4: List of components, their technical and pricing details and number required

Part	Vendor	Qty	Total Cost
Securing Bracket	printed	1	$0.00

Air Bladder	Med Warehouse	1	$20.52
Silicon Tubing	Amazon	1	$7.23
Battery	Canada Robotix	1	$61.99
Hosyond 2.8 Inches TFT	Amazon	1	$25.99
0.1uF - Capacitor_SMD:C_0603_1608Metric_Pad1.08x0.95mm_HandSolder	Digikey	1	$2.27
1uF Ceramic - Capacitor_SMD:C_0603_1608Metric_Pad1.08x0.95mm_HandSolder	Digikey	3	$2.27
1nF Ceramic - Capacitor_SMD:C_0603_1608Metric_Pad1.08x0.95mm_HandSolder	Digikey	1	$2.27
10uF Ceramic - Capacitor_SMD:C_0603_1608Metric_Pad1.08x0.95mm_HandSolder	Digikey	4	$2.27
100uF - Capacitor_SMD:C_0603_1608Metric_Pad1.08x0.95mm_HandSolder	Digikey	1	$2.27
4.7uF Ceramic - Capacitor_SMD:C_0603_1608Metric_Pad1.08x0.95mm_HandSolder	Digikey	1	$2.27
0.01uF Ceramic - Capacitor_SMD:C_0603_1608Metric_Pad1.08x0.95mm_HandSolder	Digikey	1	$2.27
2.2uF Ceramic - Capacitor_SMD:C_0603_1608Metric_Pad1.08x0.95mm_HandSolder	Digikey	1	$2.27
100uF Electrolytic - Capacitor_SMD:EEE-FN1A101UR	Digikey	1	$2.27
10uF Ceramic - Capacitor_SMD:C_0603_1608Metric_Pad1.08x0.95mm_HandSolder	Digikey	2	$2.27
22uF - Capacitor_SMD:C_0603_1608Metric_Pad1.08x0.95mm_HandSolder	Digikey	1	$2.27
B120-E3 - Diode_SMD:D_SMA	Digikey	2	$2.27

ESD9101P2T5G - Diode_SMD:D_SOD-923	Digikey	1	$2.27
PESD3V3S1UB,115 - Diode_SMD:D_SOD-523	Digikey	2	$2.27
MBRM120LT3G - Diode_SMD:D_SOD-123F	Digikey	2	$2.27
JST Connector - Connector_JST:JST_EH_S2B-EH_1x02_P2.50mm_Horizontal	Digikey	1	$2.27
USB_C_Receptacle_USB2.0 - Connector_USB:USB_C_Receptacle_HRO_TYPE-C-31-M-12	Digikey	1	$2.27
Solenoid Valve Connector - TERM BLK 2POS SIDE ENT 3.5MM PCB	Digikey	2	$2.27
DC Pump Connector - TERM BLK 2POS SIDE ENT 3.5MM PCB (Molex)	Digikey	1	$2.27
TLP4590ATP1_F - Library:11-7A801S_TOS	Digikey	1	$2.27
2.7uH - Library:IND_R2NC_SUM	Digikey	1	$2.27
FDS6375 - Library:SOIC127P600X175-8N	Digikey	1	$2.27
10K - Resistor_SMD:R_0603_1608Metric_Pad0.98x0.95mm_HandSolder	Digikey	5	$2.27
20K - Resistor_SMD:R_0603_1608Metric_Pad0.98x0.95mm_HandSolder	Digikey	1	$2.27
2.2K - Resistor_SMD:R_0603_1608Metric_Pad0.98x0.95mm_HandSolder	Digikey	3	$2.27
100K - Resistor_SMD:R_0603_1608Metric_Pad0.98x0.95mm_HandSolder	Digikey	2	$2.27
5.1K - Resistor_SMD:R_0603_1608Metric_Pad0.98x0.95mm_HandSolder	Digikey	2	$2.27
61.2K - Resistor_SMD:R_0402_1005Metric_Pad0.72x0.64mm_HandSolder	Digikey	1	$2.27
0 - Resistor_SMD:R_0603_1608Metric_Pad0.98x0.95mm_HandSolder	Digikey	2	$2.27
220m - Resistor_SMD:R_0603_1608Metric_Pad0.98x0.95mm_HandSo	Digikey	1	$2.27

lder			
MPRLS0025PA00001A - MPRLS_HNW	Digikey	1	$2.27
LT1935ES5-TRPBF - S_5_ADI	Digikey	1	$2.27
MCP73844T-840I/MS - MSOP8_MC_MCH	Digikey	1	$2.27
LM1117T-3.3 - Package_TO_SOT_THT:TO-220-3_Horizontal_TabDown	Digikey	1	$2.27
DRV8833PW - Package_SO:TSSOP-16_4.4x5mm_P0.65mm	Digikey	1	$2.27
ESP32-WROOM-32 - RF_Module:ESP32-S3-WROOM-1U-N4	Digikey	1	$2.27
DMG2301L - Package_TO_SOT_SMD:SOT-23	Digikey	1	$2.27
LED - LED_THT:LED_D3.0mm	Amazon	3	$5.27
SW_Push - Button_Switch_THT:SW_PUSH_6mm	Amazon	4	$5.27

Therefore, the total cost to develop to build this device (discluding other costs, such as engineering hours) is: $253.84. Factoring in other associated costs is important, and is accounted for, in a rough sense, in “11.3 Appendix C: Budget Restrictions”.

6.0 Measurement & Testing Procedures

6.1 Hardware

6.1.1 Breadboard Functionality Testing

The circuit's functionality on the breadboard was verified to ensure that the components operated as intended. This process facilitated the identification and resolution of any issues before integrating the circuit into the PCB. Through functionality testing, it became evident which components needed replacement or optimization due to malfunctioning or suboptimal performance. Breadboard testing also aided in addressing power supply issues. Moreover, the effectiveness of active components was assessed through breadboard testing to determine ideal PWM values for the device’s function. Once the circuit passed comprehensive breadboard testing, it was translated into the PCB design for further development.

6.1.2 PCB Functionality Testing

PCB testing was carried out to verify that the components retained their functionality during the transition from the breadboard to the PCB. This process involved confirming the correct routing of all traces and rectifying any soldered connection issues as needed. Additionally, it included checking the functionality of all components once they were soldered onto the board and ensuring that peripheral components connected via JST connectors (a type of electrical connector manufactured by J.S.T. Mfg. Co., Ltd., Japan Solderless Terminal) maintained their functionality. Once it was confirmed that the PCB was fully operational, it was integrated into the housing. Tests such as circuit continuity test using multimeter, Voltage

operating point test on regulator, USB Power and Signal Integrity Analysis, Data Integrity Test through UART and full power analysis were conducted.

6.1.3 Pressure Sensor Testing

The test setup was done by connecting the pressure sensor to the microcontroller with the sensor's output fed directly to the control system which controlled the DC motor. The motor was controlled based on the output from the PID controller. The entire system was secured and enclosed to make sure that there weren't any external environmental effects that could impact the pressure measurements.

Before testing, we calibrated the pressure sensor against a prebuilt reference test in Arduino to ensure accuracy. The calibration was done by seeing how the sensor provided an output pressure under various conditions. These values were matched against the reference gauge under already known pressure conditions. Measurements were taken at various levels. Additionally, pressure sensors were tested by comparing performance between the HX710 differential pressure sensor, the MPRLS sensor and a health canada approved blood pressure machine to ensure the values aligned.

6.2 Software

6.2.1 Control System Testing

The gain parameters were tested by implementing a Matlab script which was written to read pressure measurement data from the serial monitor. The values were displayed on a plot which was updated in real time. As we attempted to reach optimal values for the gain parameters we were able to visualize the dynamic changes in pressure values as we slowly increased the values. We were able to compare the behavior of the system through the plots and were able to confirm the most optimal gain parameters once we reached stability in pressure without intense fluctuations and overshoot.

6.2.2 User Interface Testing

The UI was tested to ensure that all of the possible paths (Bicep venipuncture, Thigh venipuncture and Occlusion) are functioning properly in a visually pleasing manner on the screen. To ensure that the device is accessible, AODA (Accessibility for Ontarians with Disabilities Act) guidelines were adhered to.

6.3 Housing

6.3.1 Hardware Integration Testing

Testing the hardware functionality within the device focused on ensuring that the pressure components could effectively deliver pressure and be perceptible to the user across different modes of usage. Through rigorous testing, adjustments were made to enhance the effectiveness of these components, ensuring optimal performance and user experience. During the testing phase, various scenarios were simulated to assess the device's functionality under different conditions. This included testing the pressure components in both venipuncture and limb occlusion modes to ensure accurate pressure delivery and user feedback. Feedback from users was also solicited during the testing process to gather insights into their experience with the device's pressure components. This feedback was invaluable in identifying areas for improvement and refining the design to better meet user needs and expectations.

Based on the test results and user feedback, adjustments were made to optimize the performance of the pressure components. This may have involved fine-tuning pressure settings, modifying component configurations, or implementing additional features to enhance usability and effectiveness. Overall, the testing of hardware functionality played a crucial role in validating the device's performance and ensuring that it meets the requirements and expectations of users. By iteratively testing and refining the hardware components, the device can deliver reliable and effective pressure control in various clinical settings.

7.0 Performance Measurement Results

7.1 Hardware

The device reaches venipuncture pressure in approximately 35.3 seconds based on 10 separate trials conducted and holds for 60 seconds. Meanwhile, the device takes approximately 1.37 minutes to reach bicep occlusion and 1.43 minutes to reach thigh occlusion. The battery life of the device lasts for approximately 2 hours depending on how frequently the device is being inflated and deflated and if the pressure is held at a certain level, the device can operate for around 5 hours. Through the conducted FMEA (Failure Mode and Effects Analysis), certain aspects of the device were also designed with safety in mind. One main aspect taken into account was, what would happen if the power was suddenly cut to the device. With this in mind, the solenoid valve used is normally closed, meaning that without power, the valve will not open and let air release. Additionally, there are features built into the device to warn the user if they are approaching low battery levels and the user may switch between USB power and battery power depending on their approach with load sharing built into the device.

7.2 Software

The software features several timers built into the device to track how long the person is under pressure and warn the user if safe venipuncture or occlusion time has been exceeded. The software instantaneously boots up and presents the user with a menu allowing them to choose either venipuncture or occlusion, after this, they may then either choose bicep or thigh occlusion. Stats then appear on the device screen informing the user of current pressure, target pressure, timing and target timing.

8.0 Analysis of Performance

8.1 Usability Testing of Overall System Integration

Testing for usability involved assessing whether the pressure components maintained their effectiveness when integrated into the device. During testing, adjustments were made, including adjusting tubing and screw sizes for optimizing component placement to house the sensors, actuators and wiring effectively. Details regarding testing can be found in section 3.2.2 and in the appendix section 11.2.1.

8.2 Control System Testing

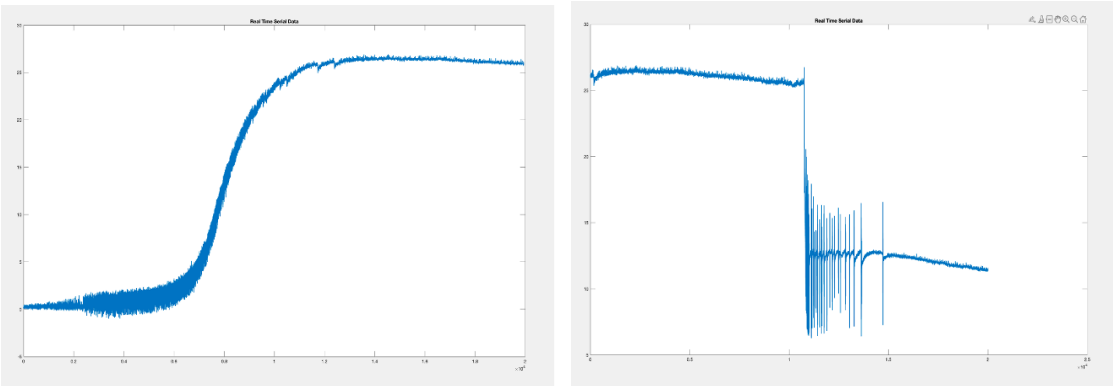


Figure 21: a) Initial pressure readings after PID gain constants applied. **b)** Testing maintenance of pressure after an input to reduce pressure.

Initially, prior to PID integration, the pressure was unable to be maintained at a set value. There were a lot of oscillations and overshoot which resulted in a lack of accuracy when trying to control the pressure. After implementing PID control, the system attained stability and behaved in a more accurate manner. This can be observed in Fig. 21 (a) which displays the pressure stabilization as it reaches the set point of 63 mmHg with minimal oscillation. The ability to stabilize pressure metrics when the cuff is tampered with and external force is applied can be observed in Fig. 21 (b) where the pressure readings rapidly oscillate during the force, only to stabilize as intended following the interruption. When the system detects a change in pressure through a user input or external force, the oscillations represent a transient response. When the new set target pressure is initially reached, the system may overshoot or oscillate around the new setpoint as it adjusts to the sudden change in the control signal. Generally, the system responded well to tuning changes, and its robustness against disturbances indicated a well-designed control strategy. The initial higher overshoot and longer settling times underscored the importance of precise tuning of the PID parameters. The adjustments to Kp, Ki, and Kd improved both transient and steady-state performance, as demonstrated from the figures above. The final gain constants were: Kp = 3, Ki = 0.05 and Kd = 1.

Increasing the proportional gain constant amplified the system's response to errors which lead to faster adjustments to reach the desired pressure. However, it was observed that if the gain value was too high, it was prone to overshooting or too many oscillations around the setpoint, resulting in unstable pressure control. For example, when Kp was set to ~7.5-8, the system experienced more aggressive behavior in terms of correction, leading to larger oscillations and lack of accuracy Fig. 22.

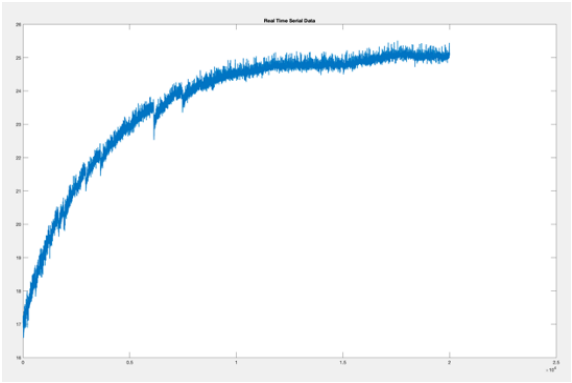


Figure 22: System response to high Kp value (~7.5-8).

The integral term accounted for accumulated errors over time. It helped eliminate steady-state errors by continuously adjusting the output based on the historical error. Increasing the integral gain improved our system's ability to eliminate steady-state errors. The integral term can also introduce instability if it is set too high, leading to oscillatory instability. When we tried to use higher values for Ki such as ~0.5-1, we noticed that once the system approached the setting pressure, it was unable to adjust the output accurately and kept overshooting causing oscillatory instability Fig. 23.

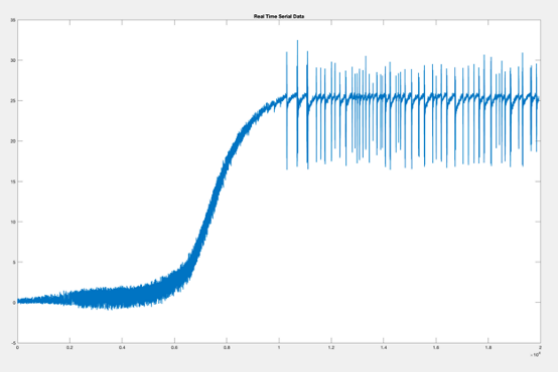


Figure 23: System response to high Ki value (~0.5-1)

Tuning the derivative term helped dampen the system's response to rapid changes, reducing overshoot and oscillations. The derivative gain made the system more stable by dampening oscillations and reducing settling time, but a very high derivative gain was shown to lead to a slow response to changes and even cause instability.

8.3 FMEA

Table 5. Failure Mode and Effects Analysis and Final Implemented Solutions

Process Function	Potential Failure Mode	Potential Effect(s) of Failure	Potential Cause(s)/ Mechanism(s) of Failure	Recommended Action(s)	Final Solution
Regulation of pressure	Malfunctioning pressure regulator	Inaccurate pressure readings, inaccurate times of compression/inflation Underinflation or overinflation of the tourniquet cuff, leading to ineffective arterial occlusion or the risk of vascular and nerve damage.	Faulty solenoid, damaged tubing, issue with communication between pump and controls	Calibrate regularly and have a maintenance plan in place	Regular Calibration, exact tubing and PID actuation to prevent overshoot

Pressure appearing on the display	Malfunctioning pressure display	Inaccurate pressure readings = over or under inflation = harm to patient	Faulty display, wiring/connections broken, problem with code/communication between controls	Ensure display material is durable, have maintenance/troubleshooting guide, periodically assess code on maintenance plan	Serial output of pressure data, AC, battery connections to prevent power loss, strong durable screen and wire connections to prevent breakage
Actuation connections from power to the cuff	Bladder or connecting tubing leaks	Gradual deflation of bladder; feel or hear leak through or around cuff, computerized system indicates low pressure or leak alarm Loss of pressure during the procedure, which can compromise the surgical field or lead to patient discomfort.	Tubing is broken, Bladder is torn	Tighten all connectors; replace cuffs or tubing if necessary. Maintain a maintenance plan.	Maintain maintenance plan, use heat shrink tubing where necessary to strengthen actuation connections
Actuation tubing remains straight to ensure proper flow	Kinking of tubing	Inability to inflate the cuff	bladder or tubing is broken/twisted, signal/communication is not functioning, issue with code	Turn connecting device to unkink tubing. Have set-up guidelines in place.	Exact tubing measurements to ensure no kinking
Power connected to the cuff	Failure to operate	Loss of electrical power receptacle.	Electrical supply failure, battery depleted/burnt out	Make sure plug is securely in. Have set-up guidelines in place.	Connections are soldered directly to the switch and battery with heat shrink tubing covering them to prevent damage. There is also power load sharing available for both battery and AC power.
Regulation of pressure	Pressure Ramping Failure	The system does not provide a gradual pressure increase or release Rapid changes in pressure, potentially causing patient discomfort, pain, or complications	Faulty valve, power surge causing pump/regulator malfunction	Use a voltage regulator and motor driver to avoid power surges, maintain smooth operation through PWM pulses	PID control is added into the code to regulate oscillations, an LDO provides consistent power and the motor drive is powered separately. The device also turns off when the battery is low and AC is not connected but maintains pressure through the normally closed solenoid valve.

Cuff Structure	Cuff rips	Inability to inflate the cuff	Faulty seams, rip in material, material faulty, cuff not stitched correctly, pump over inflated cuff	Ensure the cuff and material is strong enough to handle multiple cycles of actuation	Cuff is durable and has gone through multiple iterations of testing
Regulation of pressure	Overinflation	The actuation system over inflates the tourniquet cuff, exceeding safe pressure limits. Risk of vascular and nerve damage, pain, or other complications.	Misinterpretation of data from system, incorrect data from sensors, malfunction in the code	Ensure there is a stop gap in place within the human tolerances of pressure	Software features a timer and stops inflation when a certain pressure is reached, additionally the user can reset or turn off the device via the power switch which cuts power to the entire circuit
Regulation of pressure	Rapid loss of inflation	The actuation system decreases the pressure too fast causing rapid "reperfusion" of blood, which can potentially cause complications	Battery depletion, incorrect setup	Ensure there is a safeguard in place to avoid rapid disinflation of the cuff	Solenoid valve requires power to open and let air escape so a dead battery will not let pressure fall. Additionally, there are safeguards in the software setup to determine when to open the valve.

8.4 Overall Performance Parameters

Table 6. Performance Parameter and Evaluation Criteria

Performance Parameter	**Evaluation**
Buttons allow for increase and decrease of pressure in the cuff	PASS
Screen font/image sizes and direction are easy to understand and legible	PASS
LEDs indicate status of the device	PASS
Cuff Inflates upon device/button use	PASS
Cuff inflates to venipuncture pressure (60 mmHg)	PASS
Cuff inflates to full occlusion pressure (200 mmHg)	PASS
Cuff worn without affecting comfort of user	PASS
Cuff was easy to apply by healthcare workers	PASS

No exposed wires/electrical components that could cause danger to an individual	PASS
User is able to easily navigate the menu of the system and understand commands/prompts	PASS
User is able to read the battery level of the device	PASS

9.0 Conclusions

In conclusion, the development of the automated tourniquet device represents a significant advancement in the treatment of arthritis and related conditions. By integrating innovative hardware components such as the microcontroller and TFT SPI screen, we have successfully created a user-friendly and efficient system for administering tourniquet therapy. The physical UI, resembling a blood pressure monitor display, offers intuitive control and monitoring capabilities, enhancing the overall treatment experience for both patients and healthcare providers.

The automated tourniquet device addresses the need for a reliable and accessible solution for arthritis treatment, allowing patients to undergo therapy with greater comfort and convenience. Furthermore, the system's versatility and adaptability make it suitable for various clinical settings, from outpatient clinics to home care environments. Overall, the efforts in designing and developing this device have culminated in a valuable tool that has the potential to significantly improve the quality of life for individuals managing arthritis and similar conditions.

9.1 Future Development

While the current version of the automated tourniquet device represents a significant milestone, there are several opportunities for further development and enhancement, the details of which are given below. By pursuing these avenues for further development, we can continue to advance the automated tourniquet device, making it an indispensable tool for arthritis management and contributing to the improvement of patient care in rheumatology and orthopedics.

9.1.1 Integration of Sensor Technology

Incorporating sensor technology for real-time monitoring of limb circumference and pressure levels can provide valuable feedback to optimize treatment parameters and ensure patient safety.

9.1.2 Remote Monitoring and Connectivity

Adding connectivity features such as Bluetooth or Wi-Fi enables remote monitoring by healthcare providers and facilitates data transmission for analysis and tracking of treatment outcomes.

9.1.3 Enhanced User Interface

Continuously improving the user interface by refining the layout, adding graphical elements, and implementing touchscreen functionality can further enhance usability and user experience.

9.1.4 Customization and Personalization

Introducing features for personalized treatment protocols and user preferences will allow for tailored therapy plans based on individual needs and medical requirements.

9.1.5 Clinical Validation and Regulatory Compliance

Conducting clinical studies to validate the effectiveness and safety of the device in clinical settings is essential for gaining regulatory approval and widespread adoption.

9.1.6 Collaboration with Healthcare Professionals

Engaging healthcare professionals in the development process through collaboration and feedback will ensure that the device meets the needs and expectations of both clinicians and patients.

10.0 References

[1] R. F. Bellamy, "The causes of death in conventional land warfare: implications for combat casualty care research," Mil. Med., vol. 149, no. 2, pp. 55–62, Feb. 1984.

[2] A. Sauaia et al., "Epidemiology of trauma deaths: a reassessment," J. Trauma, vol. 38, no. 2, pp. 185–193, Feb. 1995.

[3] K. R. Harbert, "Chapter 5 - Venipuncture," in Essential Clinical Procedures (Second Edition), R. W. Dehn and D. P. Asprey, Eds., Philadelphia: W.B. Saunders, 2007, pp. 47–61.

[4] Z. W. Bell et al., "Limb Occlusion Pressure: A Method to Assess Changes in Systolic Blood Pressure," Int. J. Exerc. Sci., vol. 13, no. 2, pp. 366–373, Feb. 2020.

[5] S. Noordin, J. A. McEwen, C. J. F. Kragh, A. Eisen, and B. A. Masri, "Surgical Tourniquets in Orthopaedics," The Journal of Bone and Joint Surgery-American Volume, vol. 91, no. 12, pp. 2958–2967, Dec. 2009.

[6] R. Schreckengaust, L. Littlejohn, and G. J. Zarow, "Effects of training and simulated combat stress on leg tourniquet application accuracy, time, and effectiveness," Mil. Med., vol. 179, no. 2, pp. 114–120, Feb. 2014.

[7] E. N. Baruch et al., "Confidence-Competence Mismatch and Reasons for Failure of Non-Medical Tourniquet Users," Prehosp. Emerg. Care, vol. 21, no. 1, pp. 39–45, Jan-Feb 2017.

[8] A. Dennis et al., "Missing expectations: Windlass tourniquet use without formal training yields poor results," J. Trauma Acute Care Surg., vol. 87, no. 5, pp. 1096–1103, Nov. 2019.

[9] "A.T.S.® 5000 & A.T.S.® 3200 tourniquet systems," A.T.S.® 5000 and A.T.S.® 3200 Tourniquet Systems, https://www.zimmerbiomet.com/en/products-and-solutions/specialties/surgical/ats-5000-tourniquet.html

[10] "SmartPump tourniquet system," Stryker, https://www.stryker.com/us/en/orthopaedic-instruments/products/smartpump-tourniquet.html

[11] "ESP32WROOM32E & ESP32WROOM32UE Datasheet." Available: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf

[12] "nRF7001 - Objective Product Specification," 2023. Available: https://components101.com/sites/default/files/2023-06/nRF7001_OPS_v0.9.pdf

[13] "Introduction to USB Type-C." Available: https://cdn.sparkfun.com/assets/e/b/4/f/7/USB-C_Datasheet.pdf

[14] "LM1117 800-mA, Low-Dropout Linear Regulator," 2017. Available: https://www.ti.com/lit/ds/symlink/lm1117.pdf?ts=1707461131109&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FLM1117

[15] "AMS1117-3.3 Voltage Regulator Low Drop Out +3.3V - 1A SOT-223 5 or 10 pack," Tempero Systems Shopping. https://temperosystems.com.au/products/ams1117-3-3-voltage-regulator-low-drop-out/ (accessed Apr. 12, 2024).

[16] "MicroPressure Board Mount Pressure Sensors." Available: https://cdn.sparkfun.com/assets/b/1/7/f/a/MPRLS0025PA00001A.pdf

[17] "MPX5050 Integrated silicon pressure sensor, on-chip signal conditioned, temperature compensated and calibrated," 2023. Accessed: Dec. 04, 2023. [Online]. Available: https://www.mouser.ca/datasheet/2/302/MPX5050-3138730.pdf

[18] "DRV8833 Dual H-Bridge Motor Driver." Available: https://www.ti.com/lit/ds/symlink/drv8833.pdf?ts=1701707360583

[19] "1.69inch LCD Module - Waveshare Wiki," WaveShare, https://www.waveshare.com/wiki/1.69inch_LCD_Module

[20] V. DeStefano, S. Khan, and A. Tabada, "Applications of PLA in modern medicine," Engineered Regeneration, vol. 1, pp. 76–87, 2020. doi:10.1016/j.engreg.2020.08.002

[21] M. S. Singhvi, S. S. Zinjarde, and D. V. Gokhale, "Polylactic acid: Synthesis and biomedical applications," Journal of Applied Microbiology, vol. 127, no. 6, pp. 1612–1626, Jun. 2019. doi:10.1111/jam.14290

[22] Gordon, C. Claire et al., "Anthropometric Data" The Ergonomics Center. North Carolina State University, Apr. 2006. https://multisite.eos.ncsu.edu/www-ergocenter-ncsu-edu/wp-content/uploads/sites/18/2016/06/Anthropometric-Detailed-Data-Tables.pdf

[23] S. Naqvi, P. Potluri, P. Mandal, and P. Lewis, "Effect of different cuff types on blood pressure measurement: Variation in BP values for different cuff types," Journal of Industrial Textiles, vol. 47, no. 7, pp. 1478–1495, Feb. 2017. doi:10.1177/1528083717695838

[24] S. Sasaki, N. Murakami, Y. Matsumura, M. Ichimura, and M. Mori, "Relationship between Tourniquet Pressure and a Cross-Section Area of Superficial Vein of Forearm," ACTA MEDICA OKAYAMA, vol. 66, no. 1, pp. 67–71, 2012.

[25] S. Naqvi, M. D. Husain, P. Potluri, P. Mandal, and P. Lewis, "Pressure distribution under different types of blood pressure measurement cuffs," Journal of Industrial Textiles, vol. 47, no. 1, pp. 89–103, Mar. 2016. doi:10.1177/1528083716637868

[26] J. Mcewen, K.Inkpen, "Tourniquet safety: Preventing skin injuries." The Surgical Technologist. vol 34. pp. 7-15, 2002.

[27] "What is the material of the Cuff?," Philips, https://www.usa.philips.com/c-f/XC000001114/what-is-the-material-of-the-cuff

[28] S. Noordin, J. A. McEwen, C. J. Kragh, A. Eisen, and B. A. Masri, "Surgical Tourniquets in orthopaedics," The Journal of Bone and Joint Surgery-American Volume, vol. 91, no. 12, pp. 2958–2967, Dec. 2009. doi:10.2106/jbjs.i.00634

[29] J. ‐P. Estebe, A. Le Naoures, L. Chemaly, and C. Ecoffey, “Tourniquet pain in a volunteer study: Effect of changes in cuff width and pressure,” Anaesthesia, vol. 55, no. 1, pp. 21–26, Jan. 2000. doi:10.1046/j.1365-2044.2000.01128.x
[30] P. Palatini and R. Asmar, “Cuff challenges in blood pressure measurement,” The Journal of Clinical Hypertension, vol. 20, no. 7, pp. 1100–1103, Jul. 2018. doi:10.1111/jch.13301
[31] R. Alroobaea and P. J. Mayhew, “How many participants are really enough for usability studies?,” 2014 Science and Information Conference, Aug. 2014. doi:10.1109/sai.2014.6918171
[32] Xu S;Xu Z;Li D;Cui T;Li X;Yang Y;Liu H;Ren T;, “Recent advances in flexible piezoresistive arrays: Materials, design, and applications,” Polymers, https://pubmed.ncbi.nlm.nih.gov/37376345/ (accessed Dec. 4, 2023).
[33] “The differences between capacitive & piezoresistive pressure sensors,” The Differences Between Capacitive & Piezoresistive Pressure Sensors - Pressure Sensors - Mass Flow Sensors - Humidity Sensors - Micropumps, https://www.servoflo.com/company/general-information/news/the-differences-between-capacitive-piezoresistive-pressure-sensors (accessed Dec. 4, 2023).
[34] “RF Wireless World,” Advantages of optical sensor | Disadvantages of optical sensor, https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-Optical-Sensor.html#:~:text=,%E2%9E%A8Susceptible%20to%20physical%20damage (accessed Dec. 4, 2023).
[35] “Strain gauges vs. piezoelectric sensors,” Tacuna Systems, https://tacunasystems.com/knowledge-base/comparing-strain-gauges-to-piezoelectric-sensors/ (accessed Dec. 4, 2023).
[36] B. Tuncali, H. Boya, Z. Kayhan, and S. Arac, “Tourniquet pressure settings based on limb occlusion pressure determination or arterial occlusion pressure estimation in total knee arthroplasty? A prospective, randomized, double blind trial,” *Acta Orthopaedica et Traumatologica Turcica*, vol. 52, no. 4, pp. 256–260, 2018. doi:10.1016/j.aott.2018.04.001
[37] G. Parati, J. P. Saul, M. Di Rienzo, and G. Mancia, “Spectral analysis of blood pressure and heart rate variability in evaluating cardiovascular regulation,” *Hypertension*, vol. 25, no. 6, pp. 1276–1286, 1995. doi:10.1161/01.hyp.25.6.1276
[38] “Velocity Analysis of a DC Brushed Encoder Motor using Ziegler-Nichols Algorithm: A Case of an Automated Guided Vehicle.” https://www.researchgate.net/publication/309455948_Velocity_Analysis_of_a_DC_Brushed_Encoder_Motor_using_Ziegler-Nichols_Algorithm_A_Case_of_an_Automated_Guided_Vehicle?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6Il9kaXJlY3QiLCJwYWdlIjoiX2RpcmVjdCJ9fQ (accessed Feb. 09, 2024).
[38] B. J. Briscoe, E. Pelillo, and S. K. Sinha, “Scratch hardness and deformation maps for polycarbonate and polyethylene,” Polymer Engineering & Science, vol. 36, no. 24, pp. 2996–3005, Dec. 1996. doi:10.1002/pen.10702
[39] “Is there a typical state machine implementation pattern?,” Stack Overflow. https://stackoverflow.com/questions/133214/is-there-a-typical-state-machine-implementation-pattern
[40] R. A. Grier, A. Bangor, P. Kortum, and S. C. Peres, “The system usability scale,” Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 57, no. 1, pp. 187–191, Sep. 2013. doi:10.1177/1541931213571042

11.0 Appendices

11.1 Appendix A: User Testing Criteria

1. Name	2. Age	3. Occupation	Equipment Expertise	4. Assigned Sex at Birth	5. Height	6. Weight	7. Left Arm Circumference	8. Right Arm Circumference	9. What colour would you most like to see on the device?	10. The design is an adequate size.	11. The design looks adjustable to a variety of body types.	12. The project weight of the design is comfortable on the arm.	13. I think that I would like to use this system frequently.	14. I found the system unnecessarily complex.
			Patient	Male	5'10"	200lbs	38cm	37.5cm	Blue	2	5	5	5	1
			Medical Professional	Female	5'3	128lbs	29cm	27.5cm	Blue	5	5	5	5	1
			Patient	Female	5'5"	110lbs	25cm	24.5 cm	Blue	4	5	5	2	1
			Patient	Male	5'9"	155lbs	35cm	36cm	Blue	5	5	5	1	1
			Medical Professional	Female	5'6"	145lbs	30cm	30.5cm	Purple	5	5	5	5	1
			Patient	Female	5'2"	100lbs	24cm	24.5cm	Blue	4	4	5	4	5
			Patient	Female	5'8	135lbs	27cm	26.6cm	Blue	5	5	5	5	1
			Patient	Female	166cm	160lb	33.5cm	31cm	Blue	4	5	5	5	1
			Patient	Female	5'5"	110lbs	25cm	25.6cm	Blue	5	5	5	4	1
			Medical Professional	Female	5'4"	133lbs	28cm	26.7cm	Blue	5	5	5	3	1
			Patient	Male	5'8"	144lbs	32cm	33cm	Blue	5	5	5	4	2
			Medical Professional	Male	5'9"	156lbs	35cm	35.3cm	Blue	4	5	5	4	1
			Patient	Male	6'3"	252lbs	40cm	39.25cm	Blue	4	5	5	4	1
			Patient	Male	5'11"	172lbs	36cm	37cm	Blue	5	5	5	3	1
			Medical Professional	Male	6'0"	192lbs	34cm	34.3cm	Blue	5	4	5	4	1
										4.2	5	5	3.6	1

15. I thought the system was easy to use.	16. I think that I would need the support of a technical person to be able to use this system.	17. I found the various functions in this system were well integrated	18. I thought there was too much inconsistency in this system.	19. I would imagine that most people would learn to use this system very quickly.	20. I found the system very cumbersome to use.	21. I felt very confident using the system.	22. I needed to learn a lot of things before I could get going with this system.
5	1	3	1	5	1	5	1
5	1	5	1	5	1	5	1
5	1	4	1	5	1	5	1
4	1	4	1	5	1	4	1
5	1	4	1	5	1	4	1
5	1	5	1	5	2	4	2
5	2	5	1	5	1	5	1
5	5	5	1	5	1	4	2
5	1	4	1	4	1	4	4
4	2	4	1	5	1	5	1
5	1	4	1	5	1	4	2
4	2	5	1	4	1	4	1
5	1	4	1	5	1	4	1
5	1	4	1	5	1	4	1
5	1	4	1	5	1	4	1
4.8	1	4	1	5	1	4.6	1

Figure 21. a-b) Screenshots of questions asked of participants for user testing procedures
Note: Names, ages and occupations blocked out for confidentiality

11.2 Appendix B: Source Code

Note: For documentation of all code, Github and version control was utilized in order to separate code revisions, review changes to the main branch and avoid merge conflicts. The Github repository can be accessed here.

11.2.1 Control System Testing Code

```
%

close all; clear all;

% serial object

s = serialport("/dev/cu.usbserial-0001",115200);

% open serial port

fopen(s);

% Plotting data
```

```
i = 1;
while(1)
 data(i) = single(str2double(fscanf(s)));

 if i <= 20000
  plot(data,'Linewidth',1.5);
 else
    plot(data(end-20000:end),'Linewidth',1.5);
 end
 title('Real Time Serial Data');
 %pause(0.1);
 i = i+1;
end
```

11.2.2 Device Code

12.2.2.1 arrow_up.c

```
#ifdef __has_include
   #if __has_include("lvgl.h")
      #ifndef LV_LVGL_H_INCLUDE_SIMPLE
         #define LV_LVGL_H_INCLUDE_SIMPLE
      #endif
   #endif
#endif

#if defined(LV_LVGL_H_INCLUDE_SIMPLE)
   #include "lvgl.h"
#else
   #include "lvgl/lvgl.h"
#endif
```

```
#ifndef LV_ATTRIBUTE_MEM_ALIGN

#define LV_ATTRIBUTE_MEM_ALIGN

#endif


#ifndef LV_ATTRIBUTE_IMAGE_ARROW_UP

#define LV_ATTRIBUTE_IMAGE_ARROW_UP

#endif


const LV_ATTRIBUTE_MEM_ALIGN LV_ATTRIBUTE_LARGE_CONST LV_ATTRIBUTE_IMAGE_ARROW_UP uint8_t arrow_up_map[] = {

  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xbb, 0x2c, 0x39, 0x24, 0x39, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x19, 0x24, 0x39, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x39, 0x24, 0x39, 0x24, 0x19, 0x24, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xbb, 0x2c, 0x39, 0x24, 0x39, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x39, 0x24, 0x39, 0x24, 0x19, 0x24, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xbb, 0x2c, 0x1a, 0x1c, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x39, 0x24, 0x1a, 0x1c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x1c, 0xbb, 0x2c, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0xbb, 0x2c, 0x19, 0x1c, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x39, 0x24, 0x39, 0x24, 0x19, 0x24, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0xbb, 0x2c, 0x39, 0x24, 0x39, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x39, 0x24, 0x1a, 0x1c, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x1a, 0x1c, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0xbb, 0x2c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x19, 0x24, 0x19, 0x1c, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x1c,
0x19, 0x24, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xbb,
0x2c, 0x39, 0x24, 0x39, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x39, 0x24,
0x39, 0x24, 0xbb, 0x2c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1a,
0x1c, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0xbb, 0x2c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xbb, 0x2c, 0x19,
0x1c, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x1c, 0x19, 0x24, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xbb, 0x2c, 0x1a, 0x1c, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,

0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x39, 0x24, 0xbb, 0x2c, 0x00,

0x00, 0x1a, 0x1c, 0x19, 0x24, 0x1a, 0x1c, 0xbb, 0x2c, 0x00,

0x00, 0xbb, 0x2c, 0x39, 0x24, 0x19, 0x24, 0x00,

0x00, 0xbb, 0x2c, 0x1a, 0x1c, 0x19, 0x24, 0x39, 0x24, 0xbb, 0x2c, 0x00, 0x0

12.2.2.2 arrow_down.c

```
#ifdef __has_include

   #if __has_include("lvgl.h")

      #ifndef LV_LVGL_H_INCLUDE_SIMPLE

         #define LV_LVGL_H_INCLUDE_SIMPLE

      #endif

   #endif

#endif


#if defined(LV_LVGL_H_INCLUDE_SIMPLE)

   #include "lvgl.h"

#else

   #include "lvgl/lvgl.h"

#endif



#ifndef LV_ATTRIBUTE_MEM_ALIGN

#define LV_ATTRIBUTE_MEM_ALIGN

#endif


#ifndef LV_ATTRIBUTE_IMAGE_ARROW_DOWN

#define LV_ATTRIBUTE_IMAGE_ARROW_DOWN

#endif


const LV_ATTRIBUTE_MEM_ALIGN LV_ATTRIBUTE_LARGE_CONST LV_ATTRIBUTE_IMAGE_ARROW_DOWN uint8_t arrow_down_map[] = {

  0x00, 0x00, 0xbb, 0x2c, 0x39, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
```

0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x1a, 0x1c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

0xbb, 0x2c, 0x1a, 0x1c, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x1c, 0x1a, 0x1c, 0xbb, 0x2c, 0x00, 0x00,

0x39, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x1a, 0x1c, 0x00, 0x00,

0x39, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,

0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x1c, 0x1a, 0x1c,

0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,

0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,

0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,

0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,

0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,

0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,

0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,

0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,

0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,

0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,

0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,

0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x39, 0x24, 0x19, 0x24,

0x1a, 0x1c, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x1a, 0x1c,

0xbb, 0x2c, 0x39, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24,
0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19,
0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0x24, 0x19, 0

12.2.2.3 main.cpp

```
#include <lvgl.h>

#include <TFT_eSPI.h>

#include <Wire.h>

#include <Adafruit_I2CDevice.h>

#include "Adafruit_MPRLS.h"

#include <PID_v1.h>

//#include <examples/lv_examples.h>

//#include <demos/lv_demos.h>


/*Set to your screen resolution and rotation*/

#define TFT_HOR_RES   240

#define TFT_VER_RES   320

#define TFT_ROTATION  LV_DISPLAY_ROTATION_0
```

```
/*LVGL draw into this buffer, 1/10 screen size usually works well. The size is in bytes*/
#define DRAW_BUF_SIZE (TFT_HOR_RES * TFT_VER_RES / 10 * (LV_COLOR_DEPTH / 8))
uint32_t draw_buf[DRAW_BUF_SIZE / 4];

static uint32_t lvgl_refresh_timestamp = 0u;
#define LVGL_REFRESH_TIME 5u

// You dont *need* a reset and EOC pin for most uses, so we set to -1 and don't connect
#define RESET_PIN  -1  // set to any GPIO pin # to hard-reset on begin()
#define EOC_PIN    -1  // set to any GPIO pin to read end-of-conversion by pin
#define UP_BUTTON 42
#define DOWN_BUTTON 41
#define SOLENOID_PIN  37
#define MOTOR_PIN  39
#define LED1_PIN -1
#define LED2_PIN -1
#define SLEEP_PIN 35
#define BL_PIN 9
#define STATUS_LED 19
#define POWER_LED 20

const float VENEPUNCTURE_PRESSURE = 60;
const float BICEP_PRESSURE = 203;
const float THIGH_PRESSURE = 300;
float startPressure;
float currentPressure;
float targetPressure;

const unsigned long VENEPUNCTURE_TIME = 60000;
const unsigned long BICEP_TIME = 5400000;
const unsigned long THIGH_TIME = 7200000;
unsigned long startTime = 0;
```

```
// unsigned long currentTime = 0;
unsigned long targetTime = 0;

Adafruit_MPRLS mpr = Adafruit_MPRLS(RESET_PIN, EOC_PIN);

// PID Setup
double PIDPressure;
double Input, Output;
float calibration_value;
double Kp = 3, Ki = 0.05, Kd = 1; // Example PID tuning values, adjust as needed
PID myPID(&Input, &Output, &PIDPressure, Kp, Ki, Kd, DIRECT);

void actuationScreen(const char *mode);
void occlusionScreen();
void bicepOcclusionScreen();
void thighOcclusionScreen();

LV_IMAGE_DECLARE(arrow_up);
LV_IMAGE_DECLARE(arrow_down);

lv_obj_t *pressure_val;
lv_obj_t *vene_label;
lv_obj_t * bar;
lv_timer_t * bar_timer;
int bar_value;
char test2[6];
bool timerOn = false;
int pressureTime;
bool timeStarted = false;

#if LV_USE_LOG != 0
void my_print( lv_log_level_t level, const char * buf )
```

```
{
  LV_UNUSED(level);
  Serial.println(buf);
  Serial.flush();
}
#endif

/* LVGL calls it when a rendered image needs to copied to the display*/
void my_disp_flush( lv_display_t *disp, const lv_area_t *area, uint8_t * px_map)
{
  /*Call it to tell LVGL you are ready*/
  lv_display_flush_ready(disp);
}

/*use Arduinos millis() as tick source*/
static uint32_t my_tick(void)
{
  return millis();
}

static void occlusion_up_pressed(lv_event_t * event)
{
  //Original Code
  lv_event_code_t code = lv_event_get_code(event);

  if(code == LV_EVENT_PRESSED)
  {
    targetPressure = BICEP_PRESSURE;
    bar_value = (int)BICEP_TIME/1000;
    targetTime = BICEP_TIME;
    actuationScreen("Bicep Occlusion");
  }
```

```
}

static void occlusion_down_pressed(lv_event_t * event)
{
  //Original Code
  lv_event_code_t code = lv_event_get_code(event);

  if(code == LV_EVENT_PRESSED)
  {
    targetPressure = THIGH_PRESSURE;
    bar_value = (int)THIGH_TIME/1000;
    targetTime = THIGH_TIME;
    actuationScreen("Thigh Occlusion");
  }
}

static void up_pressed(lv_event_t * event)
{
  //Original Code
  lv_event_code_t code = lv_event_get_code(event);

  if(code == LV_EVENT_PRESSED)
  {
    targetPressure = VENEPUNCTURE_PRESSURE;
    bar_value = (int)VENEPUNCTURE_TIME/1000;
    targetTime = VENEPUNCTURE_TIME;
    actuationScreen("Venepuncture");
  }
}

static void down_pressed(lv_event_t * event)
{
```

```
  //Original Code
  lv_event_code_t code = lv_event_get_code(event);

  if(code == LV_EVENT_PRESSED)
  {
    occlusionScreen();
  }
}

static void timer_labelupdate(lv_timer_t* timer){
  sprintf(test2, "%d", (int)currentPressure);
  lv_label_set_text(pressure_val, test2);
}

static void timer_barupdate(lv_timer_t* timer){
  bar_value--;
}

static void timer_pressureupdate(lv_timer_t* timer){
  currentPressure = (mpr.readPressure() - calibration_value) * 0.75;

  PIDPressure = targetPressure;
  bool buttonUp = digitalRead(UP_BUTTON);
  bool buttonDown = digitalRead(DOWN_BUTTON);
  if (buttonUp == HIGH) {
    targetPressure = targetPressure + 1;
  } else if (buttonDown == HIGH) {
    targetPressure = targetPressure - 1;
  }

  char pressure_target[21];
  sprintf(pressure_target, "%d", (int)targetPressure);
```

```
String target_pressure_label = (String)"Target Pressure: " + pressure_target + (String)" mmHg";
char arr[target_pressure_label.length() + 1];
        strcpy(arr, target_pressure_label.c_str());
lv_label_set_text(vene_label, arr);

Input = currentPressure;
myPID.Compute();

if (timerOn == false) {
  analogWrite(MOTOR_PIN, Output);
}

if (currentPressure >= targetPressure-3 && timeStarted == false){
  startTime = millis();
  timeStarted = true;
} else if (currentPressure < targetPressure-3){
  startTime = 0;
  timeStarted = false;
}

if (millis()-startTime > 5000 && timeStarted == true){
  lv_timer_resume(bar_timer);
  pressureTime = millis();
  analogWrite(MOTOR_PIN, 0);
  timerOn = true;
}

if (millis()-pressureTime >= targetTime && timerOn){
  lv_timer_pause(bar_timer);
  timerOn = false;
}
```

```
  if (currentPressure > targetPressure+10){
    digitalWrite(SOLENOID_PIN, HIGH);
  } else {
    digitalWrite(SOLENOID_PIN, LOW);
  }

  Serial.println(timerOn);
}

bool buttonPressed(int button) {
  return digitalRead(button) == HIGH && digitalRead(button) == HIGH;
}

bool buttonNotPressed(int button) {
  return digitalRead(button) == LOW && digitalRead(button) == LOW;
}

int my_btn_read(){
  if(buttonPressed(UP_BUTTON) && buttonNotPressed(DOWN_BUTTON)){
    return 0;
  }
  if(buttonPressed(DOWN_BUTTON) && buttonNotPressed(UP_BUTTON)){
    return 1;
  } else {
    return -1;
  }
}

void button_read( lv_indev_t * indev, lv_indev_data_t * data ){
  static uint32_t last_btn = 0;   /*Store the last pressed button*/
  int btn_pr = my_btn_read();     /*Get the ID (0,1,2...) of the pressed button*/
  if(btn_pr >= 0) {               /*Is there a button press? (E.g. -1 indicated no button was pressed)*/
```

```
    last_btn = btn_pr;          /*Save the ID of the pressed button*/
    data->state = LV_INDEV_STATE_PRESSED;  /*Set the pressed state*/
  } else {
    data->state = LV_INDEV_STATE_RELEASED; /*Set the released state*/
  }

  data->btn_id = last_btn;        /*Save the last button*/
}

static void set_arc_angle(void * obj, int32_t v)
{
  lv_arc_set_value((lv_obj_t*) obj, currentPressure);
}

static void set_bar_value(void * bar, int32_t v)
{
  lv_bar_set_value((lv_obj_t*) bar, bar_value, LV_ANIM_OFF);
}

void occlusionScreen(){
  lv_obj_t *screen = lv_obj_create(lv_screen_active());
  lv_obj_set_size(screen, TFT_HOR_RES, TFT_VER_RES);
  lv_obj_center(screen);
  lv_obj_clear_flag(screen, LV_OBJ_FLAG_SCROLLABLE);

  lv_obj_t * up = lv_imagebutton_create(screen);
  lv_imagebutton_set_src(up, LV_IMAGEBUTTON_STATE_RELEASED, NULL, &arrow_up, NULL);
  lv_obj_align(up, LV_ALIGN_CENTER, -50, -30);
  lv_obj_add_event_cb(up, occlusion_up_pressed, LV_EVENT_ALL, NULL);

  lv_obj_t * down = lv_imagebutton_create(screen);
  lv_imagebutton_set_src(down, LV_IMAGEBUTTON_STATE_RELEASED, NULL, &arrow_down, NULL);
```

```
lv_obj_align(down, LV_ALIGN_BOTTOM_MID, -50, -20);

lv_obj_add_event_cb(down, occlusion_down_pressed, LV_EVENT_ALL, NULL);


lv_obj_t *menu_label = lv_label_create( screen );

lv_label_set_text( menu_label, "Occlusion" );

lv_obj_align( menu_label, LV_ALIGN_TOP_MID, -3, 0);


static lv_style_t menu_style;

lv_style_init(&menu_style);

 lv_style_set_text_font(&menu_style, &lv_font_montserrat_20); // <--- you have to enable other font sizes in menuconfig

lv_obj_add_style(menu_label, &menu_style, 0);  // <--- obj is the label


lv_obj_t *choose_label = lv_label_create( screen );

lv_label_set_text( choose_label, "Please choose an");

lv_obj_align( choose_label, LV_ALIGN_TOP_MID, -3, 30);

lv_label_set_long_mode(choose_label, LV_LABEL_LONG_WRAP);


lv_obj_t *device_label = lv_label_create( screen );

lv_label_set_text( device_label, "occlusion mode");

lv_obj_align( device_label, LV_ALIGN_TOP_MID, -3, 45);

lv_label_set_long_mode(device_label, LV_LABEL_LONG_WRAP);


static lv_style_t choose_style;

lv_style_init(&choose_style);

 lv_style_set_text_font(&choose_style, &lv_font_montserrat_16); // <--- you have to enable other font sizes in menuconfig

lv_obj_add_style(choose_label, &choose_style, 0);  // <--- obj is the label

lv_obj_add_style(device_label, &choose_style, 0);  // <--- obj is the label


lv_obj_t *vene_label = lv_label_create( screen );

lv_label_set_text( vene_label, "Bicep Occlusion" );

lv_obj_align( vene_label, LV_ALIGN_CENTER, 50, -35);
```

```
 lv_obj_t *occlusion_label = lv_label_create( screen );
 lv_label_set_text( occlusion_label, "Thigh Occlusion" );
 lv_obj_align( occlusion_label, LV_ALIGN_BOTTOM_MID, 50, -55);

 lv_obj_t *select_label = lv_label_create( screen );
 lv_label_set_text( select_label, "Select with up/down arrows" );
 lv_obj_align( select_label, LV_ALIGN_BOTTOM_MID, 0, 3);

 lv_obj_t *battery_label = lv_label_create( screen );
 lv_label_set_text( battery_label, "73% " LV_SYMBOL_BATTERY_3);
 lv_obj_align( battery_label, LV_ALIGN_TOP_RIGHT, 3, 0);

 lv_obj_t *time_label = lv_label_create( screen );
 lv_label_set_text( time_label, "13:26");
 lv_obj_align( time_label, LV_ALIGN_TOP_LEFT, 0, 0);

 static lv_style_t vene_style;
 lv_style_init(&vene_style);
   lv_style_set_text_font(&vene_style,  &lv_font_montserrat_14); // <--- you have to enable other font sizes in menuconfig
 lv_obj_add_style(vene_label, &vene_style, 0);  // <--- obj is the label
 lv_obj_add_style(occlusion_label, &vene_style, 0);
 lv_obj_add_style(select_label, &vene_style, 0);

 static lv_style_t time_style;
 lv_style_init(&time_style);
   lv_style_set_text_font(&time_style,  &lv_font_montserrat_12); // <--- you have to enable other font sizes in menuconfig
 lv_obj_add_style(battery_label, &time_style, 0);  // <--- obj is the label
 lv_obj_add_style(time_label, &time_style, 0);
}
```

```
void homeScreen(){
  lv_obj_t *screen = lv_obj_create(lv_screen_active());
  lv_obj_set_size(screen, TFT_HOR_RES, TFT_VER_RES);
  lv_obj_center(screen);
  lv_obj_clear_flag(screen, LV_OBJ_FLAG_SCROLLABLE);

  lv_obj_t * up = lv_imagebutton_create(screen);
  lv_imagebutton_set_src(up, LV_IMAGEBUTTON_STATE_RELEASED, NULL, &arrow_up, NULL);
  lv_obj_align(up, LV_ALIGN_CENTER, -50, -30);
  lv_obj_add_event_cb(up, up_pressed, LV_EVENT_ALL, NULL);

  lv_obj_t * down = lv_imagebutton_create(screen);
  lv_imagebutton_set_src(down, LV_IMAGEBUTTON_STATE_RELEASED, NULL, &arrow_down, NULL);
  lv_obj_align(down, LV_ALIGN_BOTTOM_MID, -50, -20);
  lv_obj_add_event_cb(down, down_pressed, LV_EVENT_ALL, NULL);

  lv_obj_t *menu_label = lv_label_create( screen );
  lv_label_set_text( menu_label, "Main Menu" );
  lv_obj_align( menu_label, LV_ALIGN_TOP_MID, -3, 0);

  static lv_style_t menu_style;
  lv_style_init(&menu_style);
   lv_style_set_text_font(&menu_style, &lv_font_montserrat_24); // <--- you have to enable other font sizes in menuconfig
  lv_obj_add_style(menu_label, &menu_style, 0);  // <--- obj is the label

  lv_obj_t *choose_label = lv_label_create( screen );
  lv_label_set_text( choose_label, "Please choose a");
  lv_obj_align( choose_label, LV_ALIGN_TOP_MID, -3, 30);
  lv_label_set_long_mode(choose_label, LV_LABEL_LONG_WRAP);

  lv_obj_t *device_label = lv_label_create( screen );
  lv_label_set_text( device_label, "device mode");
```

```
lv_obj_align( device_label, LV_ALIGN_TOP_MID, -3, 45);

lv_label_set_long_mode(device_label, LV_LABEL_LONG_WRAP);


static lv_style_t choose_style;

lv_style_init(&choose_style);

 lv_style_set_text_font(&choose_style, &lv_font_montserrat_16); // <--- you have to enable other font sizes in menuconfig

lv_obj_add_style(choose_label, &choose_style, 0);  // <--- obj is the label

lv_obj_add_style(device_label, &choose_style, 0);  // <--- obj is the label


lv_obj_t *vene_label = lv_label_create( screen );

lv_label_set_text( vene_label, "Venepuncture" );

lv_obj_align( vene_label, LV_ALIGN_CENTER, 50, -35);


lv_obj_t *occlusion_label = lv_label_create( screen );

lv_label_set_text( occlusion_label, "Occlusion" );

lv_obj_align( occlusion_label, LV_ALIGN_BOTTOM_MID, 40, -70);


lv_obj_t *select_label = lv_label_create( screen );

lv_label_set_text( select_label, "Select with up/down arrows" );

lv_obj_align( select_label, LV_ALIGN_BOTTOM_MID, 0, 3);


lv_obj_t *battery_label = lv_label_create( screen );

lv_label_set_text( battery_label, adc_read_1() LV_SYMBOL_BATTERY_3);

lv_obj_align( battery_label, LV_ALIGN_TOP_RIGHT, 3, 0);


lv_obj_t *time_label = lv_label_create( screen );

lv_label_set_text( time_label, time.currentTime());

lv_obj_align( time_label, LV_ALIGN_TOP_LEFT, 0, 0);


static lv_style_t vene_style;

lv_style_init(&vene_style);
```

```
  lv_style_set_text_font(&vene_style, &lv_font_montserrat_14); // <--- you have to enable other font sizes in menuconfig

  lv_obj_add_style(vene_label, &vene_style, 0);  // <--- obj is the label

  lv_obj_add_style(occlusion_label, &vene_style, 0);

  lv_obj_add_style(select_label, &vene_style, 0);


  static lv_style_t time_style;

  lv_style_init(&time_style);

  lv_style_set_text_font(&time_style, &lv_font_montserrat_12); // <--- you have to enable other font sizes in menuconfig

  lv_obj_add_style(battery_label, &time_style, 0);  // <--- obj is the label

  lv_obj_add_style(time_label, &time_style, 0);

}


void actuationScreen(const char *mode){

  lv_timer_t * pressure_timer = lv_timer_create(timer_pressureupdate, 50, NULL);

  lv_timer_ready(pressure_timer);


  lv_obj_t *screen = lv_obj_create(lv_screen_active());

  lv_obj_set_size(screen, TFT_HOR_RES, TFT_VER_RES);

  lv_obj_center(screen);

  lv_obj_clear_flag(screen, LV_OBJ_FLAG_SCROLLABLE);


  lv_obj_t *menu_label = lv_label_create( screen );

  lv_label_set_text( menu_label, mode );

  lv_obj_align( menu_label, LV_ALIGN_TOP_MID, 0, 0);


  static lv_style_t menu_style;

  lv_style_init(&menu_style);

  lv_style_set_text_font(&menu_style, &lv_font_montserrat_24); // <--- you have to enable other font sizes in menuconfig

  lv_obj_add_style(menu_label, &menu_style, 0);  // <--- obj is the label


  vene_label = lv_label_create( screen );
```

```
lv_label_set_text( vene_label, "");
lv_obj_align( vene_label, LV_ALIGN_CENTER, 0, 120);

lv_obj_t * arc = lv_arc_create(screen);
lv_arc_set_rotation(arc, 270);
lv_arc_set_bg_angles(arc, 0, 360);
lv_arc_set_range(arc, 0, targetPressure);
lv_obj_remove_style(arc, NULL, LV_PART_KNOB);   /*Be sure the knob is not displayed*/
lv_obj_remove_flag(arc, LV_OBJ_FLAG_CLICKABLE);  /*To not allow adjusting by click*/
lv_obj_align(arc, LV_ALIGN_CENTER, 0, -20);

lv_anim_t a;
lv_anim_init(&a);
lv_anim_set_var(&a, arc);
lv_anim_set_exec_cb(&a, set_arc_angle);
lv_anim_set_duration(&a, 1000);
lv_anim_set_repeat_count(&a, LV_ANIM_REPEAT_INFINITE);
lv_anim_set_repeat_delay(&a, 500);
lv_anim_start(&a);

pressure_val = lv_label_create( screen );
lv_label_set_text(pressure_val, "");
lv_obj_align(pressure_val, LV_ALIGN_CENTER, 0, -20);

static lv_style_t pressure_label_style;
lv_style_init(&pressure_label_style);
 lv_style_set_text_font(&pressure_label_style, &lv_font_montserrat_24); // <--- you have to enable other font sizes in menuconfig
lv_obj_add_style(pressure_val, &pressure_label_style, 0);  // <--- obj is the label

bar = lv_bar_create(lv_screen_active());
lv_obj_set_size(bar, 200, 20);
lv_obj_align(bar, LV_ALIGN_BOTTOM_MID, 0, -70);
```

```
  lv_bar_set_value(bar, 100, LV_ANIM_OFF);
  lv_bar_set_range(bar, 0, bar_value);
  // lv_obj_add_event_cb(bar, event_cb, LV_EVENT_DRAW_MAIN_END, NULL);

  lv_anim_t bar_anim;
  lv_anim_init(&bar_anim);
  lv_anim_set_var(&bar_anim, bar);
  lv_anim_set_values(&bar_anim, 0, bar_value);
  lv_anim_set_exec_cb(&bar_anim, set_bar_value);
  lv_anim_set_duration(&bar_anim, 4000);
  lv_anim_set_playback_duration(&bar_anim, 4000);
  lv_anim_set_repeat_count(&bar_anim, LV_ANIM_REPEAT_INFINITE);
  lv_anim_start(&bar_anim);

  lv_timer_t * label_timer = lv_timer_create(timer_labelupdate, 500, NULL);
  lv_timer_ready(label_timer);

  bar_timer = lv_timer_create(timer_barupdate, 1000,  NULL);
  lv_timer_pause(bar_timer);
}

void setup()
{
  String LVGL_Arduino = "Hello Arduino! ";
  LVGL_Arduino += String('V') + lv_version_major() + "." + lv_version_minor() + "." + lv_version_patch();

  pinMode(MOTOR_PIN, OUTPUT);
  pinMode(SOLENOID_PIN, OUTPUT);
  pinMode(BL_PIN, OUTPUT);
  pinMode(UP_BUTTON, INPUT);
  pinMode(DOWN_BUTTON, INPUT);
  pinMode(SLEEP_PIN, OUTPUT);
```

```
pinMode(STATUS_LED, OUTPUT);
pinMode(POWER_LED, OUTPUT);

digitalWrite(SOLENOID_PIN, LOW);
digitalWrite(MOTOR_PIN, LOW);
digitalWrite(BL_PIN, HIGH);
digitalWrite(SLEEP_PIN, HIGH);
digitalWrite(POWER_LED, HIGH);

Serial.begin(115200);
Wire.begin(1,2);
Serial.println(LVGL_Arduino);

if (! mpr.begin()) {
  Serial.println("Failed to communicate with MPRLS sensor, check wiring?");
  while (1) {
     digitalWrite(STATUS_LED, HIGH);
     delay(10);
  }
}

calibration_value = mpr.readPressure();

// Initialize PID controller
myPID.SetMode(AUTOMATIC);
myPID.SetOutputLimits(0, 255); // PWM range for motor control

lv_init();

/*Set a tick source so that LVGL will know how much time elapsed. */
lv_tick_set_cb(my_tick);
```

```
  lv_display_t * disp;
  /*TFT_eSPI can be enabled lv_conf.h to initialize the display in a simple way*/
  disp = lv_tft_espi_create(TFT_HOR_RES, TFT_VER_RES, draw_buf, sizeof(draw_buf));
  lv_display_set_rotation(disp, TFT_ROTATION);

  lv_indev_t * button_indev = lv_indev_create();
  lv_indev_set_type(button_indev, LV_INDEV_TYPE_BUTTON);
  lv_indev_set_read_cb(button_indev, button_read);
  static lv_point_t button_array[2] = { { 70, 130 }, {70, 280} };
  lv_indev_set_button_points(button_indev, button_array);

  Serial.println( "Setup done" );

  lvgl_refresh_timestamp = millis();

  homeScreen();
}

static void LVGL_TaskMng( void )
{
  uint32_t now = millis();
  // LVGL Refresh Timed Task
  if( (now - lvgl_refresh_timestamp) >= LVGL_REFRESH_TIME )
  {
    lvgl_refresh_timestamp = now;
    // let the GUI does work
    lv_timer_handler();
  }
}

void loop()
{
```

```
 LVGL_TaskMng(); /* let the GUI do its work */

}
```

11.3 Appendix C: Budget Restrictions

The development of the automated tourniquet device was conducted within the constraints of specific budgetary considerations. The following details outline the budget restrictions that guided the design and development process.

11.3.1 Material Costs

The budget for material costs encompassed expenses related to procuring electronic components, hardware components, sensors, microcontrollers, TFT ESPI screens, and other necessary materials for device construction. The details of these are provided in "5.0 Material/Component List".

11.3.2 Labor Costs

Labor costs entail salaries for the 4 student engineers involved in the design, development, testing, and implementation phases of the project. The budget allocation for labor was based on hourly rates and anticipated project duration.

The labor cost is estimated to be $44,800. This is based on the calculation below.

Hourly Rate of Student Engineer (**R**, $CAD): 35; Number of Student Engineers (**N**): 4; Time Dedicated Per Student Engineer (**T**, hrs): 320; Total Labor Cost (**LC**, $CAD);

$$LC = R * N * T = 35 * 4 * 320 = 44,800$$

11.3.3 Equipment & Tool Costs

Funds were allocated for the acquisition or rental of specialized equipment and tools required for prototype fabrication, assembly, testing, and quality assurance activities. These are included in the material budget given above.

11.3.4 Testing & Validation Expenses

The budget accounted for expenses associated with testing, validation, and certification processes to ensure compliance with regulatory standards and safety requirements. These costs are included in the labor cost as shown above in "11.3.2 Labor Costs".

11.3.5 Prototyping & Iteration Costs

Funds were set aside for prototyping iterations, including the fabrication of multiple prototypes, iterations based on feedback, and refinement of design and functionality. These are accounted for in the aforementioned budget details.

11.3.6 Consultation & Outsourcing Costs

Budget restrictions included expenses related to consulting fees, outsourcing services, and subcontracting for specialized expertise or assistance required during various stages of the project. All external opinions were performed pro bono by our supervisor Dr. Ali Tavallei and other engineers (online platforms like Reddit), and therefore, are not mentioned in the previous budgeting details.

11.3.7 Travel & Miscellaneous Expenses

The budget also covered travel expenses for team members attending meetings, conferences, or site visits related to the project. These are considered as part of labor cost as shown above in “11.3.2 Labor Costs”. No funds were allocated for miscellaneous expenses such as software licenses, documentation, and administrative costs, as free or previously owned softwares were exclusively used.

It is important to note that adherence to budget restrictions was paramount throughout the project lifecycle to ensure efficient resource utilization and project sustainability. Any deviations from the budget were carefully documented via a Google Sheet and approved through proper channels to maintain financial transparency and accountability. By effectively managing budget restrictions, the development team was able to achieve project objectives while delivering a cost-effective solution that meets the needs of stakeholders and end-users.