Electronic System Design for Continuous Bladder Volume Monitoring

by

Emine Bardakcı

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Electrical and Electronics Engineering

KOÇ ÜNİVERSİTESİ

August 10, 2023

Electronic System Design for Continuous Bladder Volume Monitoring

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Emine Bardakcı

and have found that it is complete and satisfactory in all respects,

and that any and all revisions required by the final

examining committee have been made.

Committee Members:

____

Assist.Prof. Levent Beker (Advisor)

____

Assist. Prof. Beren Semiz Gürsoy

____

Assist. Prof. Emin İstif

Date: ________________________________

To my dearest husband, your unwavering belief in me, even in the face of failure, transforms my world into a place of boundless happiness.

ABSTRACT

Electronic System Design for Continuous Bladder Volume Monitoring

Emine Bardakcı

Master of Science in Electrical and Electronics Engineering

August 10, 2023

Bladder volume is an important parameter assessed by healthcare professionals for the early diagnosis and treatment of lower urinary tract dysfunctions. While bladder catheterization remains the gold standard method, its invasive nature often deters patients, and it can potentially lead to various complications. To address this issue, non-invasive bladder volume measurements are conducted using ultrasound. However, this measurement method still requires a clinical setting, and patients are asked to keep a bladder diary. Unfortunately, this approach is not sufficient and efficient enough for continuous monitoring. Moreover, despite significant advancements in wearable ultrasound studies, which have shown great promise, integrating with electronic devices still requires a separate desktop electronic setup. However, our solution involves offering an ultrasonic bladder volume monitoring (UBVM) device, achieved by adding miniaturized electronic hardware to ultrasonic transducers embedded in a flexible patch. We have tested the accuracy of this device with healthy volunteers. Through a non-invasive measurement method, the flexible ultrasonic patch placed on the lower abdomen, transmits signals of bladder volume measurements to a cloud system via an electronic board. By utilizing a smartphone application connected to the cloud system, both users and healthcare professionals can continuously monitor bladder volume values. This setup ensures constant and autonomous monitoring, providing convenience and ease for both users and healthcare professionals. We present an example of the application of the Internet of Things (IoT) in the healthcare domain through the utilization of ultrasonic patch and electronic systems within our current setup. This thesis also paves the way for various wearable ultrasound projects that can incorporate IoT applications.

Keywords: *bladder, volume, flexible, ultrasonic, monitoring, continuous, IoT*

ÖZETÇE

Sürekli Mesane Hacmi Takibi için Elektronik Sistem Tasarımı

Emine Bardakcı

Elektrik ve Elektronik Mühendisliği, Yüksek Lisans

10 Ağustos 2023

Mesane hacmi, alt üriner sistem disfonksiyonlarının erken teşhisi ve tedavisi için sağlık profesyonelleri tarafından değerlendirilen önemli bir parametredir. Mesane kateterizasyonu hala altın standart yöntem olarak kullanılsa da invaziv doğası hastaları çoğu zaman caydırır ve çeşitli komplikasyonlara neden olabilir. Bu sorunu çözmek için invaziv olmayan bir yöntem olarak ultrason kullanılarak mesane hacmi ölçümleri gerçekleştirilmektedir. Ancak bu ölçüm yöntemi hala klinik bir ortamda gerçekleştirilmesini gerektirir ve hastalardan mesane günlüğü tutmaları istenir. Ne yazık ki, bu yaklaşım sürekli izleme için yeterli ve verimli olmamaktadır. Ayrıca, giyilebilir ultrason çalışmalarında önemli ilerlemeler olmuş olsa da elektronik cihazlarla entegrasyonu için hala ayrı bir masaüstü elektronik kurulum gerektirir. Fakat sunduğumuz çözüm, esnek bir yama içine yerleştirilen ultrasonik transdüserlere küçültülmüş elektronik donanım ekleyerek elde edilen ultrasonik mesane hacmi izleme (UMHI) cihazını içerir. Bu cihazın doğruluğunu sağlıklı gönüllü insanlarla test ettik. Non-invaziv bir ölçüm yöntemi kullanarak, alt karın bölgesine yerleştirilen esnek ultrasonik yama, mesane hacmi ölçümlerinin sinyallerini bir bulut sisteme elektronik bir kart üzerinden iletmektedir. Bulut sistemine bağlı bir akıllı telefon uygulamasını kullanarak hem kullanıcılar hem de sağlık profesyonelleri sürekli olarak mesane hacmi değerlerini izleyebilirler. Bu kurulum, kullanıcılar ve sağlık profesyonelleri için sürekli ve otomatik izleme sağlayarak kullanım kolaylığı ve rahatlığı sunar. Mevcut sistemimiz içinde, Nesnelerin İnterneti (IoT) uygulamasının sağlık alanındaki örneğini ultrasonik yama ve elektronik sistemlerin kullanımıyla sunuyoruz. Bu tez aynı zamanda, IoT uygulamalarını içerebilecek çeşitli giyilebilir ultrason projelerinin yolunu da açmaktadır.

Anahtar Kelimeler: *mesane, hacim, esnek, ultrasonik, izleme, sürekli, IoT*

ACKNOWLEDGEMENTS

I extend my deepest gratitude to my invaluable supervisor, Levent Beker, whose guidance, encouragement, and support have been instrumental in the successful completion of this research project. His expertise, insights, and constructive feedback have been indispensable throughout the thesis journey.

My heartfelt appreciation goes to my colleagues and friends, Alp Timuçin Toymuş and Umut Can Yener, at the Bio-integrated Microdevices Laboratory, whose assistance, collaboration, and genuine friendship have made my academic pursuit not only enjoyable but also etched with unforgettable memories.

To my beloved family, I owe an immense debt of gratitude for their unconditional love, patience, and selfless sacrifices. They have been a constant pillar of support, standing by me during moments of both triumph and tribulation.

I would like to express my sincere thanks to my husband, Ahmet Emin Keskinoğlu, whose unwavering support, encouragement, and understanding have been the driving force behind my determination throughout this journey. He has been more than a life partner; he is my best friend and a constant source of inspiration. I consider myself truly fortunate to have him by my side.

Lastly, I am immensely grateful to Scientific and Technological Research Council of Turkey (TUBITAK) 2210/C programme for their invaluable support in funding this thesis. Furthermore, I extend my sincere appreciation to Koç University Nanofabrication and Nanocharacterization Center (n2STAR) for granting me access to the infrastructure.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

UBVM	Ultrasonic bladder volume monitoring
IoT	Internet of Things
BPH	Benign prostatic hyperplasia
LUTD	Lower urinary tract dysfunction
PCB	Printed circuit board
MCU	Microcontroller Unit
FFC	Flat flexible cable
DMA	Direct memory access
AWS	Amazon web services
WSS	WebSocket Secure
API	Application programming interface
T/R	Transmit/receive
BLE	Bluetooth Low Energy
IPC	Inter-processor communication
PWM	Pulse width modulation
MUX	Multiplexer
GATT	Generic Attribute Profile
VS	Visual studio code

Chapter 1:
INTRODUCTION

The bladder serves as an organ responsible for the storage and controlled voiding of urine in a healthy individual. However, disruptions in its functional integrity due to various pathological conditions, such as benign prostatic hyperplasia (BPH) [1], multiple sclerosis, and spinal cord injuries [2,3], can restrict its normative performance, giving rise to symptoms such as persistent urinary urgency and incomplete emptying following micturition. This condition constellation is collectively termed lower urinary tract dysfunction (LUTD) [4]. The harmful impact of LUTD on patients can significantly compromise their ability to carry out routine activities and, consequently, precipitate a deterioration in their overall quality of life. Notably, LUTD affected a substantial proportion of the global population, encompassing around 1.9 billion individuals, representing approximately 45% of the total population in 2008 [5]. By 2018, this prevalence had escalated by 18.4%, affecting an estimated 2.3 billion people worldwide [6]. It is further projected that the economic burden of this bladder pathology surpassed 73 billion USD in 2019 [7].

In the evaluation of bladder function, parameters such as bladder volume and voiding frequency play a significant role [8]. Currently, the gold standard method for assessing the bladder is catheterization, which involves the invasive placement of a catheter into the bladder to obtain information about urine (such as volume, flow rate, and pH value) [9]. However, this method is not preferred by both clinicians and patients due to its invasive nature. Moreover, it is not favored for routine monitoring as it can lead to certain health complications [10]. Therefore, information about residual urine in the bladder after voiding is often obtained using ultrasound. However, this method also requires professional healthcare personnel and desktop installations, limiting its use to cases requiring hospitalization. As this method does not allow for outpatient monitoring, patients are advised to keep a daily voiding diary [11]. Although some studies suggest the feasibility of monitoring bladder volume through implanted sensors [12-16], this approach is only suitable for advanced-stage patients [17]. To provide a more comfortable means of monitoring bladder volume for patients already facing an unpleasant situation, non-invasive monitoring devices are needed [18].

Through years of ongoing research, numerous devices have been developed for measuring bladder volume. The principle of operation for these devices is based on signals coming from the anterior and posterior walls of the bladder, which provide information about its volume [19,21]. However, these devices only warn the user when the bladder is full, thereby not providing accurate and continuous volume information [22-24]. Additionally, they have rigid and inflexible structures and require the use of ultrasound gel for accurate imaging, which limits users' daily activities [25,26]. Recent studies have yielded the development of high-accuracy devices for measuring bladder volume, but these devices either have a rigid structure or lack electronic components altogether [27,31]. In practical everyday life, such devices are not very efficient for users. Methods such as electrical impedance analysis [32,35] and near-infrared spectroscopy [36,38] have also demonstrated their usability for measuring bladder volume. However, these methods are susceptible to body movements and only provide a rough estimation of bladder volume based on urine-dependent characteristics [39,40]. Consequently, these types of devices are not deemed suitable for accuracy and continuous monitoring due to their restriction of user daily movements and demanding usage.

Recent studies have made significant advancements in wearable ultrasonic transducer technologies, offering new possibilities in healthcare services [41-49]. However, many developed devices lack electronic components, rely on desktop electronic hardware, and do not provide continuous monitoring of bladder volume. This thesis presents a novel technology by incorporating a wireless and wearable electronic hardware module with fabricated ultrasonic transducers, enabling continuous monitoring through wireless transmission and cloud connectivity. The transducers' air-backed and flexible patch-based structure allows for highly accurate, non-invasive, and comfortable usage without the need for ultrasound gel, thereby minimizing disruption to users' daily activities. Cloud-based bladder volume estimation is achieved using a simple algorithm (spherical fitting algorithm) to avoid the complexity associated with ultrasound imaging systems and algorithms. This ensures low energy consumption while maintaining high accuracy in tracking bladder volume. The Bluetooth technology employed in the electronic module enables users' bladder volume estimations to be sent to the Amazon cloud system via an Internet connection on their phones and displayed in the application every 30 minutes. The continuous functionality of the system is supported by both in vitro and in vivo testing.

Chapter 2:
GENERAL SYSTEM OPERATION

The complete bladder volume monitoring system comprises piezoelectric ultrasonic transducers and a printed circuit board (PCB) consisting of the electronic system. A patch of four piezoelectric transducers is fabricated as a flexible for a comfortable touch on the human skin. The electronic system comprises of a microcontroller (MCU), two analog multiplexers, a high voltage pulser chip, two opAmps, two filters and a comparator. When the manufactured ultrasonic patch is placed on the area of the bladder (lower abdomen) on the skin, it starts to send the pulse signals generated from the electronic system to the bladder. After triggering the bladder, the ultrasonic patch receives the echo signals from both the anterior and posterior walls of the bladder, and the echo signals are then transferred to the electronic board via a flat flexible cable (ffc) for further calculating the volume. Each ultrasonic transducer is triggered from the electronic system board, which accepts the reception of echo signals at specific time intervals. Calculating the bladder volume is dependent on the location information number of the timestamps of the echo signals. When a minimum of four location information come from both the anterior and posterior walls, the volume can be calculated using a spherical fitting algorithm. Besides, the volume calculation can be calculated simply using the sphere volume formula.

Each echo signal reached the electronic board and was then transferred to the receiver circuit for filtering and amplifying the signals due to the low amplitude. After that, the signals reach the comparator converting them to digital signals for feeding the MCU capture pin. The comparator converts the analog signal to digital according to a specific threshold value. If the input signal exceeds the threshold value, it is converted to a High (5V) logic value; if not, it is converted to a Low (0V) logic value. The MCU detects this changing state using its built-in edge detection hardware and stores these values in a buffer via direct memory access (DMA). The MCU sends each stored value to the ST BLE SENSOR application on the phone. This application has several features, such as logging data to CSV or cloud logging. In this project, amazon web services (AWS)' AWS IOT Core service is used for logging the data. On the cloud side, the volume calculation is made on the lambda function, passing through several services as a pipeline consisting of AWS IoT Core, AWS Kinesis Firehose, AWS S3 Bucket, AWS Lambda, AWS Appsync, AWS DynamoDB Table, and a custom user application.

AWS IoT Core provides the IoT devices to connect them to other AWS services using MQTT, HTTPS, and the MQTT over WebSocket Secure (WSS) protocols. These protocols enable the devices to publish or subscribe the messages to the IoT Core topic. AWS Kinesis Firehose can inject the data from IoT Core with a specific time interval and convert the type of collected data—json or .csv file for being stored in an S3 Bucket. S3 Bucket is a container allowing users to store and access data through other services according to security permissions. AWS Lambda allows users to create a function to run their code to transmit, transform, or manipulate data. The bladder volume calculation code is written in a lambda function that runs by every file upload in an S3 bucket. Then, through the AWS Appsync service, the volume parameter is stored in a table in the AWS DynamoDB service, which provides connection and access to data from a custom application on the phone via an application programming interface (API). A custom application is written using JavaScript language on the Expo platform allowing the users to create applications on Android or IOS. The custom application on the phone shows the current bladder volume of a person by reaching that data from the DynamoDB table. Finally, users can continuously monitor their bladder volume by opening the app that automatically calculates the volume on their phones.

Chapter 3:
ELECTRONIC SYSTEM DESIGN

The complete electronic system board is shown in Figure 3.1.

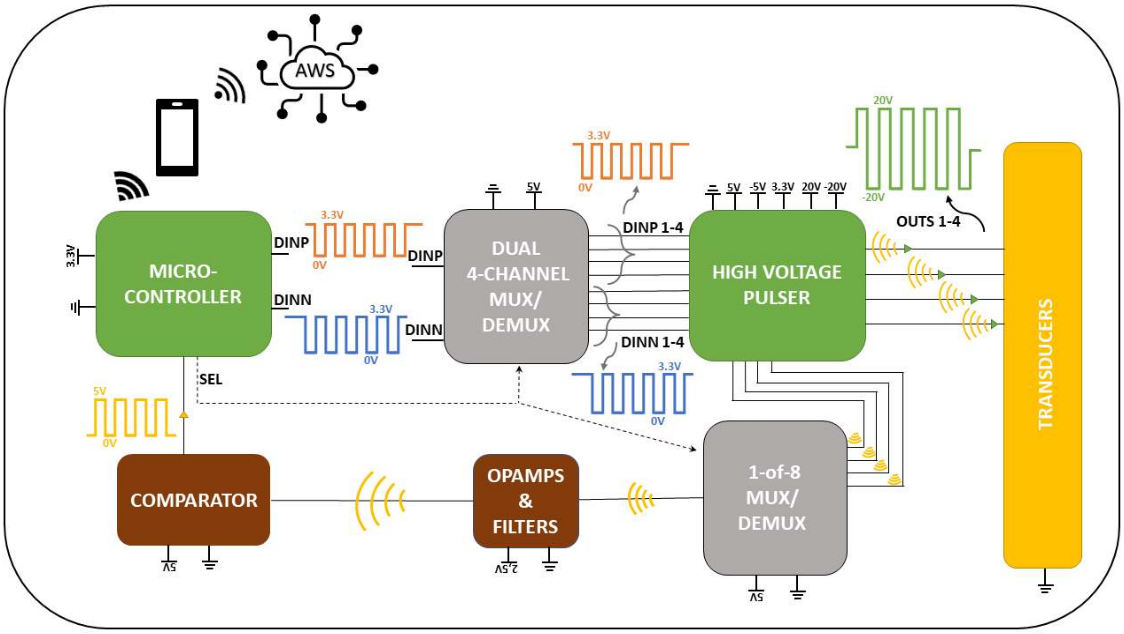


Figure 3.1: The schematic of the complete electronic system

Dual-core MCU (STM32WB55RG, STMicroelectronics) is chosen as the brain of the electronic system for the following properties: Ultra-low power while in transmitting (8.1mA ~ 13.1mA) and receiving (3.8mA ~ 7mA) phases; it has built-in Bluetooth functionality to transfer the data to a phone, tablet, or computer and supports Bluetooth® 5.3 specification, IEEE 802.15.4-2011 PHY and MAC, supporting Thread 1.3 and Zigbee® 3.0. For the case of triggering the four transducers, there is a need for multiplexing the pulse signals (DINP, DINN in the Figure 3.1), so a multiplexer/demultiplexer (DG409, Maxim Integrated) is used between the MCU and the high-voltage pulser. High voltage pulses (OUTS in the Figure 3.1) are needed to drive the piezoelectric transducers effectively, so a high-voltage pulser chip (Max14808, Maxim Integrated) is used. The chip needs five power supplies to work, which are +5V, -5V, +3.3V, and VPPA and VNNA, producing the desired voltage levels up to +100V and -100V. After sending the pulses to the ultrasonic transducers according to arranged voltage levels (VPPA and VNNA are selected +/-20V), transducers get excited and generate a trigger signal of approximately 40V peak-to-peak pulses. These pulses then hit the anterior and posterior of the bladder, and an echo signal containing the bladder's location information is returned to the chip via the transmit/receive (T/R) switch. Each echo signal comes sequentially to the chip and is transferred to the receive circuit through the T/R switch.

In receive phase of the system includes another multiplexer (DG408, Maxim Integrated), two opAmps (OPA357, Texas Instruments) with a gain of 10 each, filters, and a comparator (MAX941, Maxim Integrated). Although the echo signals come sequentially from the ultrasonic transducers, there is the possibility of losing an echo signal because of its very low amplitude when four of the echo signal channels are combined in the Max14808 chip. Another four-channel multiplexer (DG408) is used to separate the signals to get away from the risk of interference. Since the echo signals had a low amplitude at around mV, there is a need for amplification to be detectable from the MCU. Therefore, two opAmps with a gain of 10 each are used to increase the amplitude, and two low-pass filters are used to eliminate the higher-order harmonics in the echo signal caused by the amplifiers' gain-bandwidth product. In addition, an RC low-pass filter is employed with a cut-off frequency of 5.88 MHz, where the signal strength is reduced by 3 decibels. After amplifying and filtering phase, the echo signal is directed to the comparator (MAX941) works according to the specified threshold value. When the echo signal is detected, it is converted to 5V; otherwise, it is converted to 0V. This conversion is essential for reaching the detectable level of the MCU, which senses and stores the timestamps of the coming echo signals in its DMA for further processing on the application.

3.1 Transmit Phase

3.1.1 Generating pulses from the MCU

The system is controlled by an Stm32wb55rgv6 MCU, which sends pulses, receives echo timestamps, and controls (via SEL in the Figure 3.1) the analog muxes. Since the MCU has a built-in Bluetooth® Low Energy, it sends the received echo timestamps to an Android phone for further processing on Amazon Web Services (AWS). Built-in two cores in the MCU named Arm Cortex-M4 and Arm Cortex-M0+ are used with an operating frequency of 32 Mhz. The chip incorporates a dual-core design, with the Cortex-M4 core functioning at speeds of up to 64 MHz to handle complex tasks and the Cortex-M0+ core operating at speeds of up to 32 MHz to handle more straightforward tasks that require less processing power. This dual-core setup enables the chip to balance performance and power consumption, making it ideal for high-performance processing and low-power usage applications. For instance, the Cortex-M4 core can handle tasks like data processing for Bluetooth Low Energy (BLE) communication and ultrasonic distance measurement, while the Cortex-M0+ core can manage tasks like collecting and processing sensor data. The two cores can effectively communicate with each other through an inter-processor communication (IPC) mechanism, ensuring efficient coordination and seamless operation between them. Before using the MCU chip directly, the tests are made using the NUCLEO-WB55RG board, which is a development kit (Figure 3.2).

Figure 3.2 : STM32 Nucleo-64 development board with STM32WB55RG MCU

The system operation starts when the MCU sends two sets of five 2 MHz pulses amplitude of 3.3V every 2.5 seconds. The two sets (Figure 3.3) have a 180° phase difference between them according to a specification provided by the Max14808 datasheet.

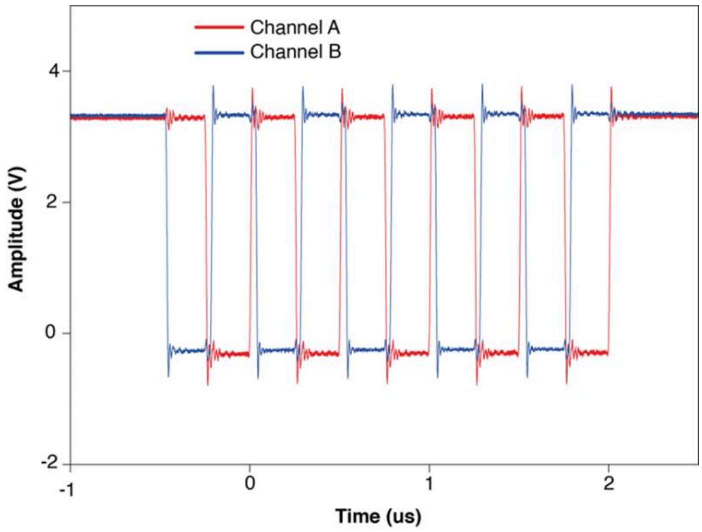


Figure 3.3: The blue and red signals with a 180° phase difference

The custom code is written using C language on Stm32 Cube Ide 1.10.0 version to generate these signals. The custom code implementation is made on an example project named "p2pserver," which is provided by ST Microelectronics. The output compare and one pulse mode of the MCU is used to create two sets of five 2 MHz pulses with a 180° phase difference. Output compare mode generates a signal according to specified timestamps in memory without interrupting the processor. For instance, let the processor's clock as 40 MHz, and light each light every 250, 500,750, and 1000 ms. First, a timer channel needed to be selected as the "output compare channel"; then, the clock should be divided using prescalar because every clock cycle (tick) should be 1 ms. So, a prescalar should be 40000, and the counter period would be selected as 1000; it refers to 1 second. If the mode of this channel is set as "toggle on match," the light would start at the light specific time intervals mentioned earlier. However, this method generates a continuous signal at specified time intervals. To generate a specific number of pulses at a time, another method of output compare mode, called "one pulse mode," should be used. In this method, a pulse stays at high for a certain time, then it stops. In addition to the previously specified configurations, by entering the number of repetitions in the timer channel to 9, five pulses will be produced, but the prescalar value should be updated considering that the frequency will be halved due to toggling. Finally, for the start of the pulses at a high level, the polarity of the channels was selected as low.

Due to the one pulse mode working principle, it stops after generating a pulse, but the signal must be produced at certain time intervals. Therefore, two other pins of the timer channel were used, one generating a pulse width modulation (PWM) signal and the other sensing the rising edges of these signals. Thus, pulse generation was triggered thanks to the edges detected at certain time intervals. To generate a PWM signal, in the timer configuration tab, the channel is configured as PWM generation while the other channel of the timer is configured as input capture direct mode selecting rising edge mode in the polarity section of the channel. At every capturing rising edge, an interrupt would occur, and so pulse generation. The example schema is shown in Figure 3.4:

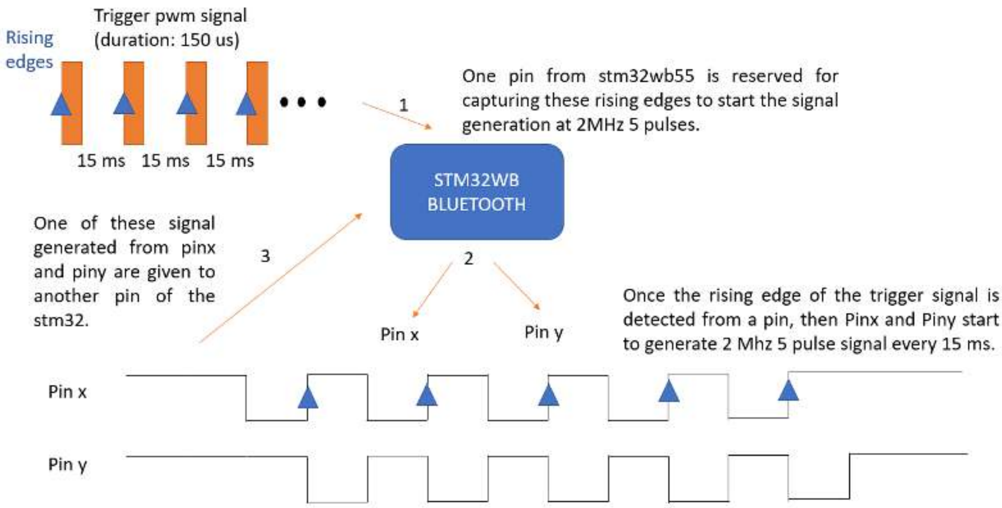
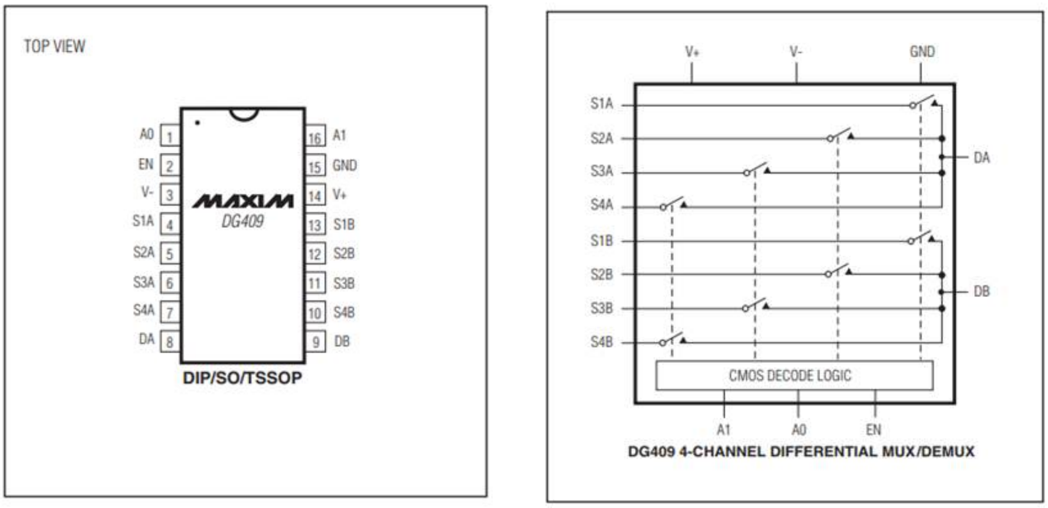


Figure 3.4: Mechanism of the generating pulses

3.1.2 Multiplexing the pulse signals

The piezo patch consists of four transducers; for each of these, corresponding pulse signals are needed to drive them. Two sets that have a 180° phase difference in between are required to generate high voltage pulses with +/- 20V (peak-to-peak 40V) pulses specified Max14808 datasheet. One of the sets is called DINN, and the other is called DINP. However, for the four transducers, there is a need for eight channels to transfer the pulse signals to the Max14808 high voltage pulser chip. Therefore, DG409 multiplexer/demultiplexer is used between the MCU and high voltage pulser chip. In this project, DG409 is used as a multiplexer (mux), two inputs which are 180° phase difference in between, are given from the MCU, and DG409 conveys these signals according to the configuration on the truth table. The MCU controls the muxes at specific time intervals by using its selection and enable pins. Selection pins can enable the changes of switches via open/close, and enable pin provides the current switch is on or not. Pin Configurations, functional diagrams, and truth tables are shown in Figure 3.5.



(a) (b)

A1	A0	EN	ON SWITCH
X	X	0	None
0	0	1	1
0	1	1	2
1	0	1	3
1	1	1	4

DG409

LOGIC "0" $V_{AL} \le 0.8V$, LOGIC "1" $V_{AH} \ge 2.4V$

(c)

Figure 3.5: Diagrams belonging to DG409: (a) pin configuration, (b) functional diagram, (c) truth table.

The test of the DG409 multiplexer

The multiplexer is connected to the single supply (+5V), and the negative power line is connected to the ground as specified in the datasheet of the product. The two sets of pulse signals coming from the stm32wb55 discovery board are connected to the inputs of the multiplexer, and one channel's output of the MCU and multiplexer is shown in Figure

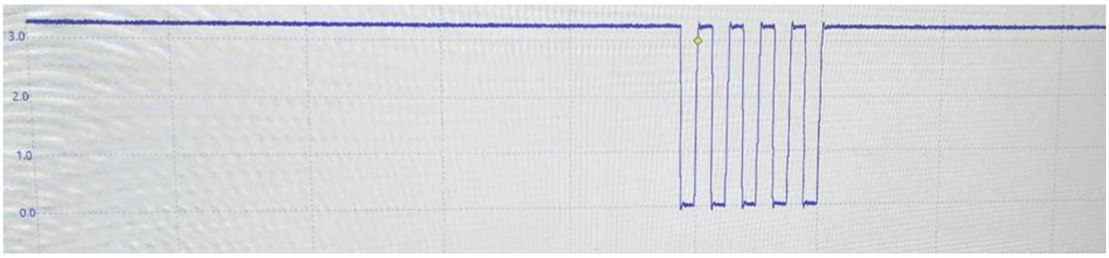


Figure 3.6: Microcontroller's output

3.6 and Figure 3.7, respectively (since every channel shows the same output). During the test, the MCU is supplied from the laptop's USB connection (Figure 3.8).

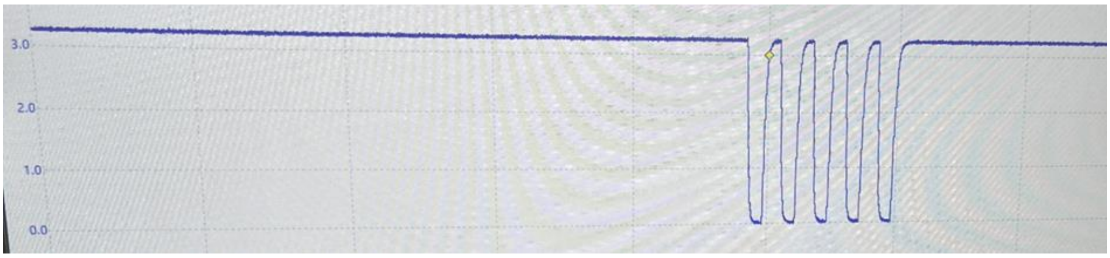


Figure 3.7: DG409 multiplexer's output

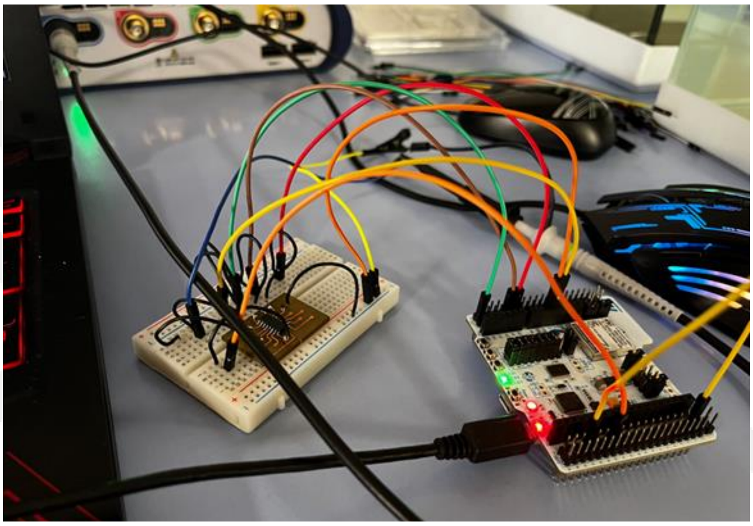
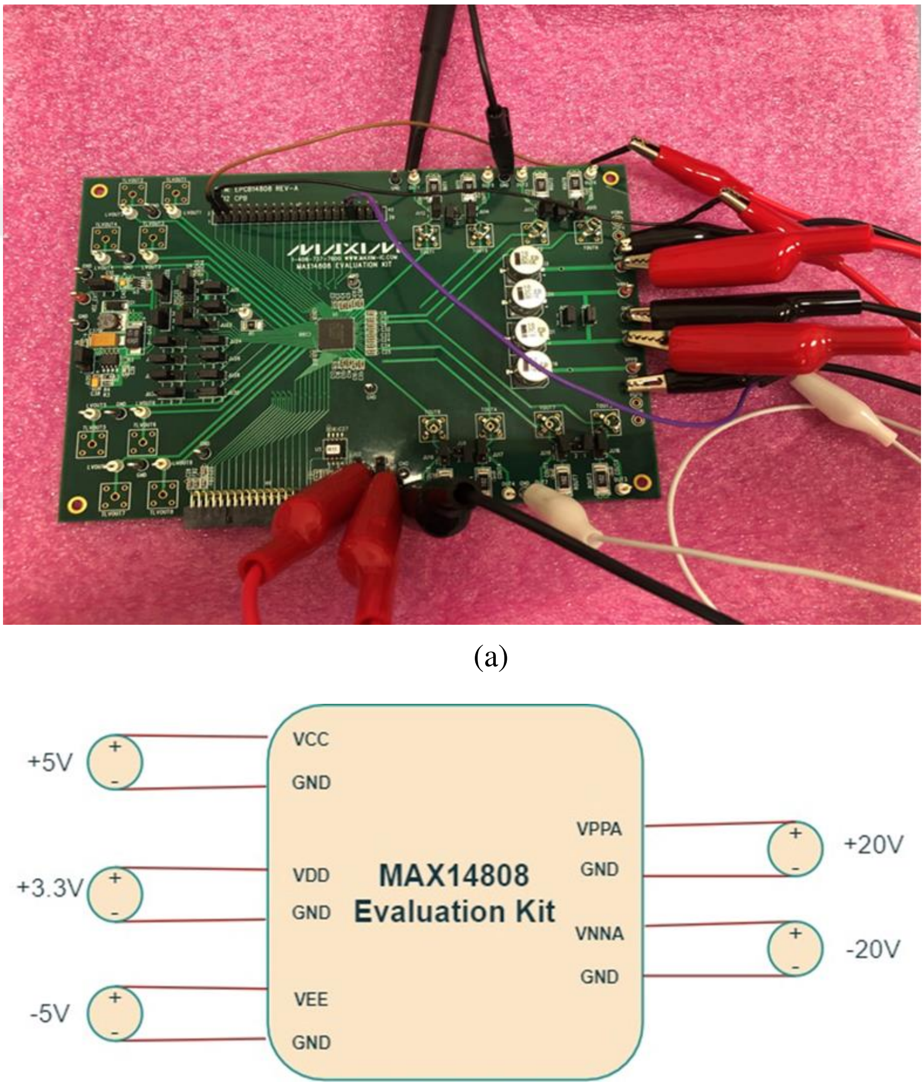


Figure 3.8: Setup: DG409 multiplexer (left), STM32 Nucleo-64 development board (right)

3.1.3 Generating high voltage pulses

Signals coming from the multiplexer are needed to be amplified to be able to drive the four transducers. Therefore, Max14808 high voltage pulser evaluation board (Figure 3.9a) is used first for testing purposes. The pulse signals can be amplified up to +/-105V. All the power connections are shown in Figure 3.9b. Max14808 has four operations; shutdown, octal three-level, quad five-level, and transmit disable modes. In shutdown mode, it is not possible to transmit or receive the signals since all the channels are disabled. In the octal three-level mode, two logic input signals (DINN and DINP) are required for controlling all eight channels. In this project, the octal three-level mode is used. In quad five-level mode, three logic input signals are required for signal generation. In transmit disable mode, all eight transmit channels are disabled, and it is not possible to transmit the signals. T/R switches can be opened in this case and pass the low-voltage signals read from LVOUT pins. T/R switches in the pulser chip protect the sensitive receiver circuitry from such high voltage pulses during transmission and forward the received signal to the receiver circuit for converting the analog echo signals to digital. For controlling the current drive capability, the minimum (0.5A) is selected and configured from specified pins. For the testing purpose, two logic inputs with a 180° phase difference are given to Max14808, and the results are shown in Appendix A2 for 10V, 16V, 36V, and 50V peak-to-peak pulse generations.

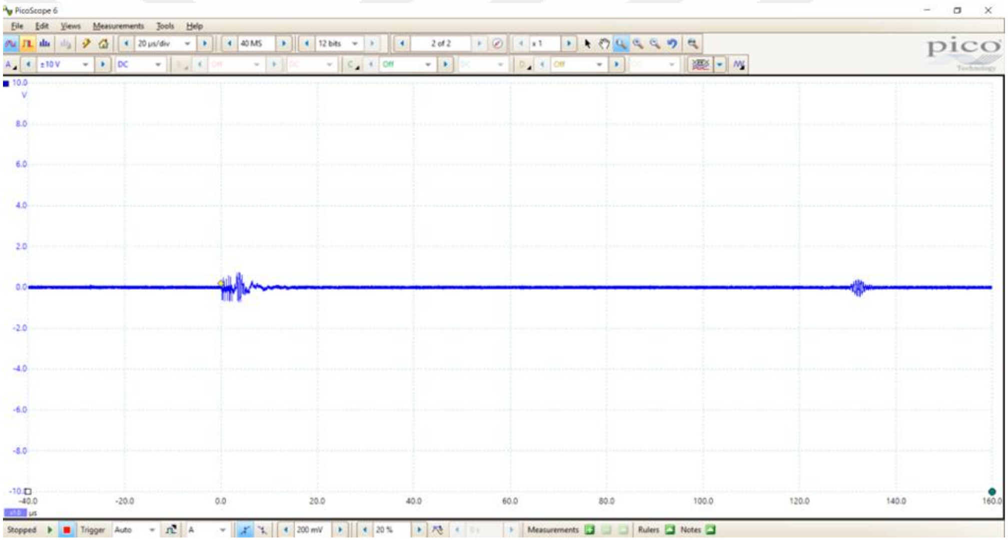


(b)

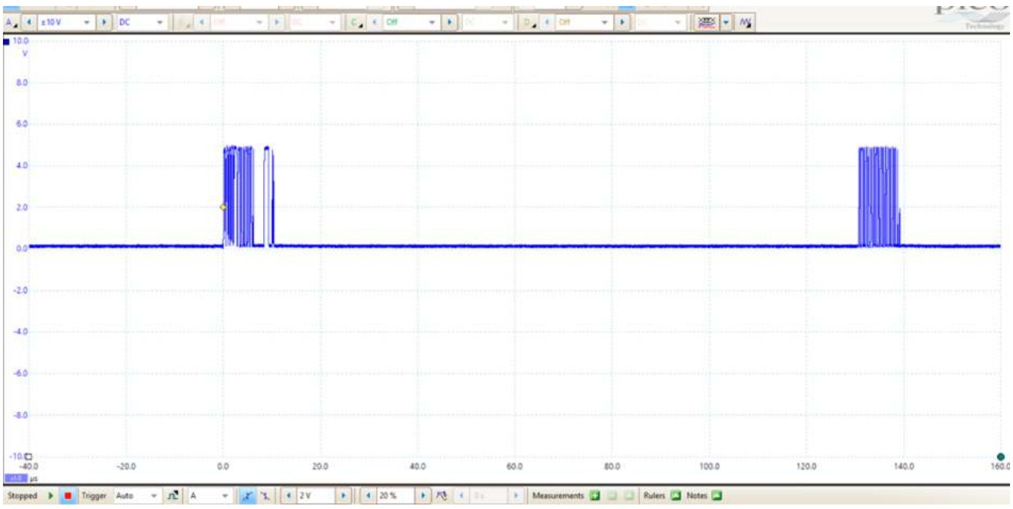
Figure 3.9: Max14808: (a) High voltage pulser evaluation board,
(b) Power connections of the MAX14808 used in the project.

3.2 Receive Phase

Echo signals from the sequentially excited transducers pass through the T/R switch of the Max14808 high voltage pulser and, in turn, pass through another analog multiplexer (DG408) controlled by the MCU, to reach the receiver circuit. Since the raw echo signals (Figure 3.10a) are out of the detectable range of the MCU, In the receive phase, there is a circuit consisting of two opAmps and low-pass filters. Such an amplification process increases the echo signal amplitude approximately 100 times. After that, an amplified echo signal passes through the comparator converting the analog signal to digital. With the threshold value of the comparator, it detects and converts to 5V (High) if the echo signal suppresses the threshold value; if not, the converted signal goes 0V (Low) (Figure 10b).



(a)



(b)

Figure 3.10: Echo Signals: (a) Raw, (b) Digitized

Echo signals converted to digital signals are then fed to the MCU pin to detect the rising edges of digital signals. These rising edges give insight into when these echo signals occur, and detecting these echo timestamps is very important for calculating the bladder volume measurement. Capturing rising edges of the digital signal is made on the edge detector hardware of the MCU. When the input comes, the current value of the counter is recorded in memory for further processing. Since the excitation frequency of the transducers is 2 MHz, echo timestamps need to be recorded fastest on the MCU flash memory. Therefore, instead of frequent interruption of the processor, which increases the processor's workload, direct memory access (DMA) is used for fast recording. DMA is a bridge that enables high-speed data transfer between the peripherals and memory or memory to memory. In this case, DMA with a peripheral to memory is used, and the recorded echo timestamps are sequentially sent to the ST Ble Sensor App, which ST Microelectronics supports for further sending to the Amazon cloud service.

3.2.1 Pulse & Echo time measurement

The first pulse burst, and echo times are measured and recorded to calculate the time-of-flight value (t_echo-t_pulse in Figure 3.11). However, there is a little problem with that. The echo signal's time and pulse burst time are measured in 32 bits, but the pulse burst time is displayed as 16 bits. In addition, the pulse burst time is reset before the finish time of the echo time. The pulse burst time must be updated every event, but the ST Ble Sensor App does not show the timestamps bigger than 16 bits.

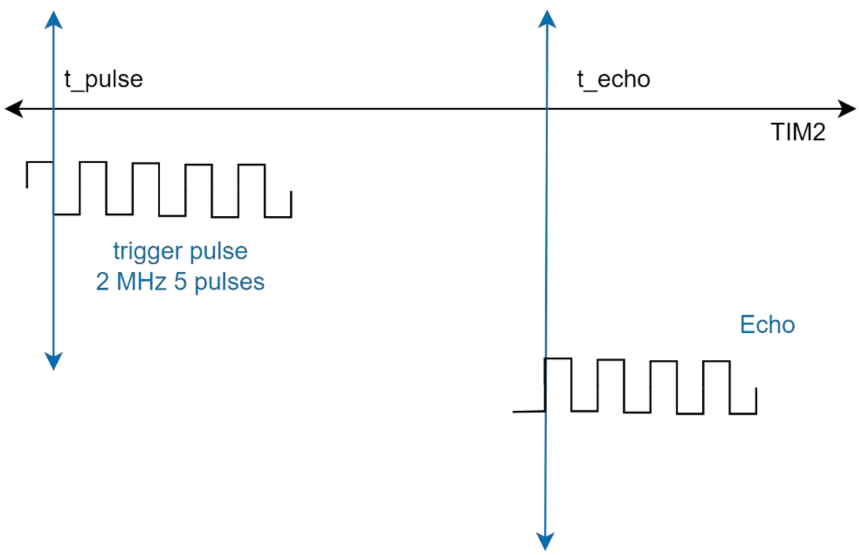


Figure 3.11: Suggested measurement of the time-of-flight

To eliminate this problem, timer 2 (TIM2), a timer that detects the rising edges of the digital echo signals, is started when the first trigger pulse is transmitted, assuming the beginning of the pulse is zero. Then the time-of-flight value becomes t_echo/2 (Figure 3.12).

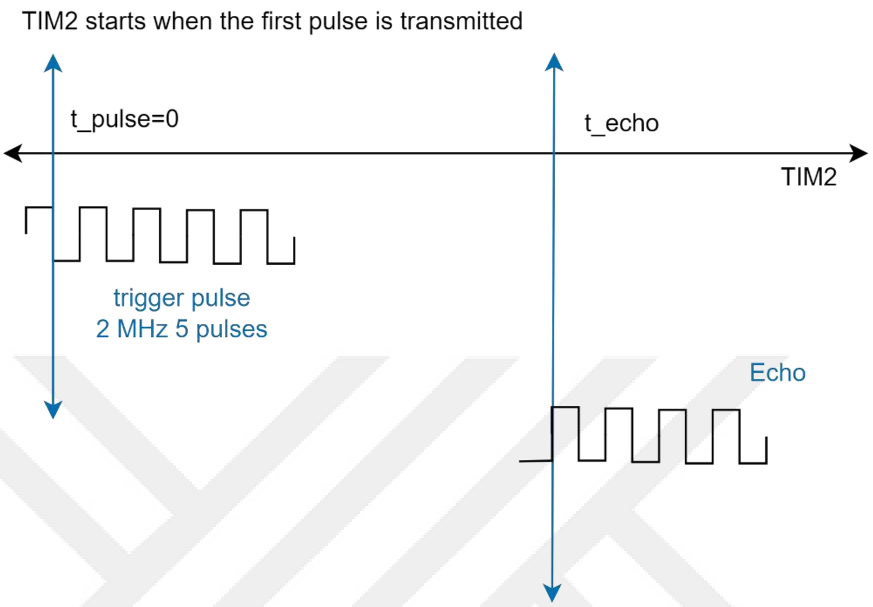


Figure 3.12: Improved measurement of the time-of-flight

3.2.2 Complete algorithm for four transducers

To measure the bladder volume continuously, the algorithm side of the project needed to be appropriately designed. Each transducer is excited four times with 2 MHz five pulses; after each excitation, the corresponding transducer's echo signal immediately comes to the MCU's capturing pin passing through the receiver circuit. Then, the echo timestamps are captured using capturing mode, recorded via DMA, and transferred to a phone using the Bluetooth feature of the MCU. Next excitation signal is sent after all of the echo timestamps recorded in the memory are transferred to a phone. When one of the transducers is excited four times, the MCU changes the switch of the multiplexer (DG409) to convey the excitation signal to another transducer. Transmitting one excitation signal and sending the echo timestamps to the phone is called "one cycle," it lasts approximately 2.5 seconds. Thus, the complete cycle for one transducer lasts 10 seconds. The whole algorithm pipeline is shown on the flow chart (Figure 3.13) and the whole code can be found in Appendix A1.

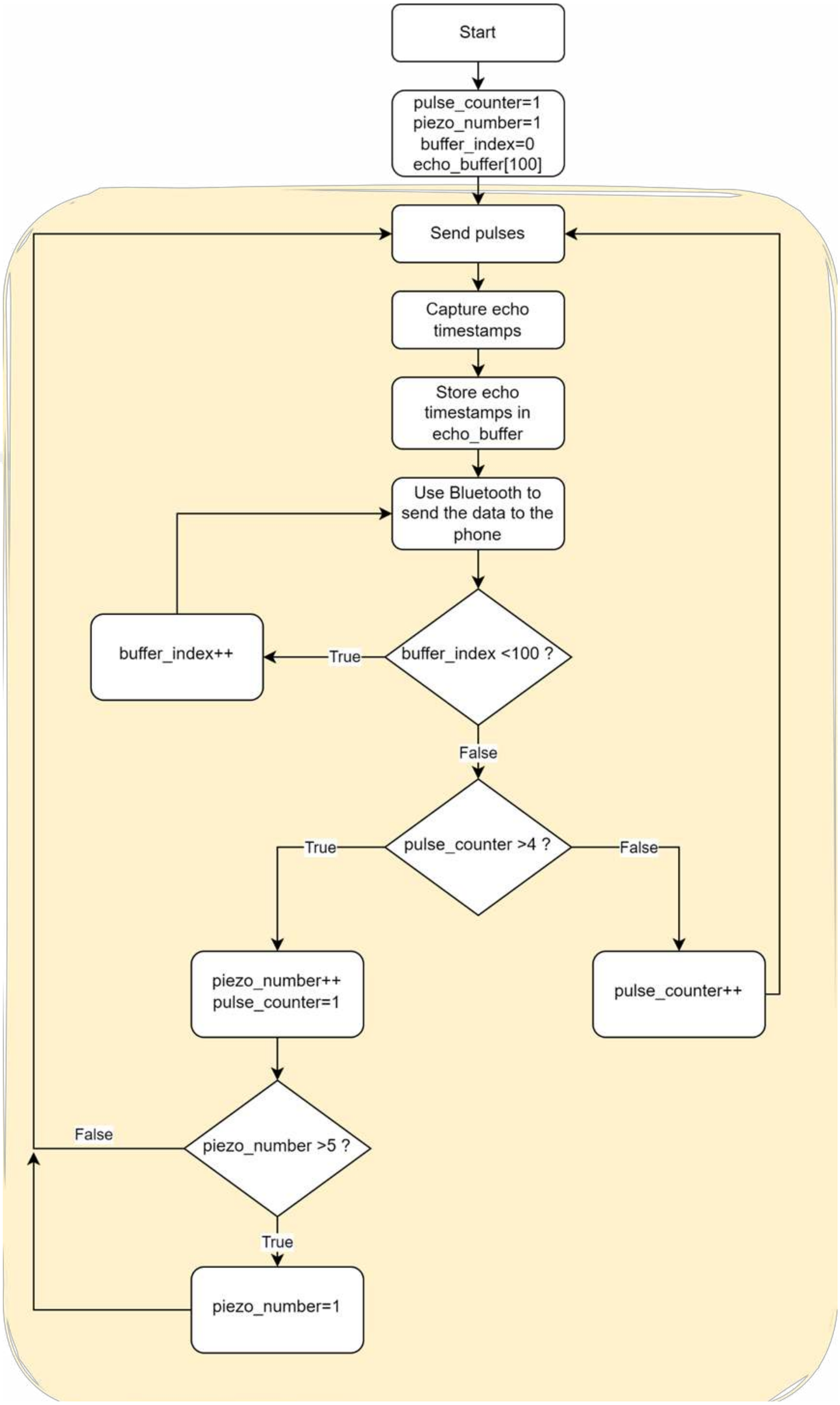


Figure 3.13: Flow chart of the complete algorithm

3.2.3 Bluetooth operation

Echo timestamps in the memory of the processor are transmitted to the phone via Bluetooth Low Energy consisting of a Generic Attribute Profile (GATT) server that includes custom services and characteristics. In the context of capturing echo timestamps, one service that contains a data capturing parameter as a characteristic and transfers the data over BLE was used. This characteristic can be accessed by other BLE devices, allowing them to read, write, or be notified. The data send over BLE is then passed through a custom application on an Android phone for further processing in Amazon Web Services (AWS). The phone between the wireless MCU and AWS behaves as a gateway for conveying the data using the Internet. The data processing stages involved in AWS allow for the transmission of bladder volume information to the phone, passing through the following services: AWS IoT, AWS Kinesis Data Firehose, AWS S3 Bucket, AWS Lambda, AWS AppSync, AWS Dynamo Table, and finally, the user application. GraphQL language is employed in this pipeline to facilitate interaction between users and the services involved. Users can utilize GraphQL's capabilities to perform operations such as reading and writing data, as well as connecting their smartphones to the pipeline through an Application Programming Interface (API).

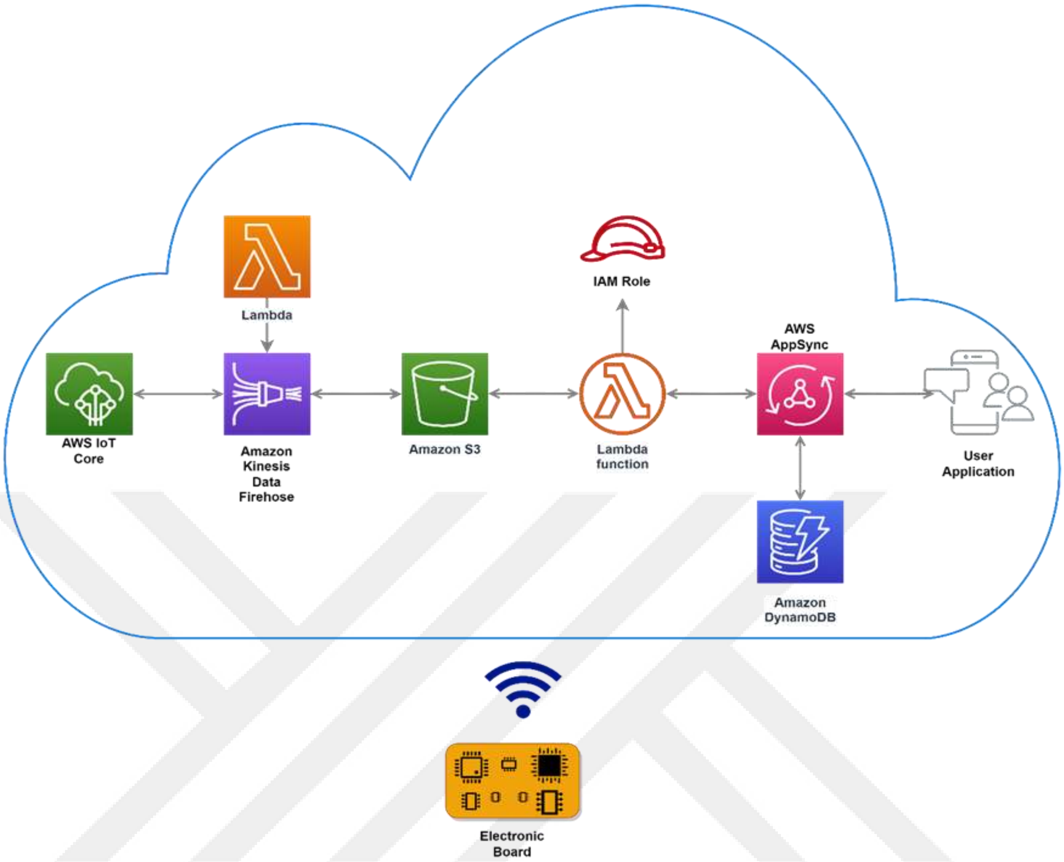
3.2.4 Amazon Web Services (AWS) & User Application

The complete services used in the AWS pipeline are shown in Figure 3.14a to calculate and display the bladder volume. AWS AppSync offers Graphql API that provides a connection between the server and the client. The simple example that Graphql API offers is looking up the weather app on the phone. That app connects the weather database in the cloud and displays the weather forecast to the user. Additionally, Graphql API enables reaching the data from different services (AWS DynamoDB, Aurora RDS) using only a single endpoint, so there is no need to create more requests to connect different services. The AWS DynamoDB table is used in this project as a database. Graphql API is created using the command window so that amplify cli, node.js, and npm modules are downloaded first. To create an application on the phone, expo cli and react native modules are used. This website [50] gives more insight into how to create an application with Graphql API. The default schema is preferred during setup because it efficiently creates a reliable pipeline. It creates a DynamoDB table linked to the AWS AppSync. Once the application-creating processes are finished, the rest are made using

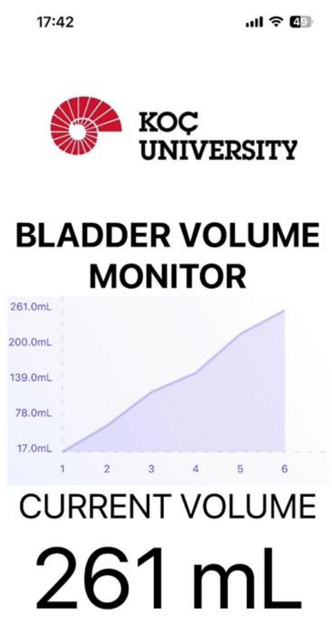
JavaScript language on visual studio code ide (VS Code). The stored data is easily manipulated using query, mutation, and subscription operations that Graphql offers. Query offers the list of the data on the database; mutation offers to update the data (create, put, delete), and subscription offers to listen to real-time data updates. Lambda function provides access to AWS AppSync using graphql endpoint so data can be created and updated using a mutation in the DynamoDB table.

The bladder volume measurement is made on the lambda function using Python language. First, the echo timestamps reach the ST Ble Sensor application on the phone via Bluetooth; they are transferred to AWS IoT Core. This application behaves as a gateway between the electronic board and AWS. The MQTT protocol establishes the connection between AWS and the phone application. Since IoT devices have a limited bandwidth to operate, this protocol provides the IoT device with a cloud connection using minimal sources and power. It has security credentials to establish a connection, which is unavailable for traditional network communication mechanisms with low bandwidth and high latency. It offers three options to get the data reliably in quality service; more than once (0), at least once (1), and exactly once (2). In this project, the MCU sends the data, so it becomes the publisher; AWS gets this data and becomes the client. However, there is a broker between the publisher and the client to convey the data to the client by filtering the data. Therefore, the publisher and client do not know each other's network locations, IP addresses, or port numbers. AWS offers security certificates to publishers while creating an account, and publishers must share these certificates while connecting to AWS to establish a secure link.

Once the echo timestamps reach the AWS IoT Core, they are extracted from AWS Kinesis Data Firehose at predetermined intervals. This service offers a record time from 60 seconds up to 300 seconds. File type conversion can be made in the lambda function connected to this service. The data coming from the IoT service is recorded as a CSV file for easy processing. The destination of the CSV files is S3 Bucket, a data storage for long-term access. With 120 seconds intervals, the Kinesis Data Firehose service provides a file to the S3 Bucket as a CSV. Every file uploading to S3 Bucket triggers the primary lambda function, which measures the bladder volume. Calculated bladder volume data is then transferred to the DynamoDB table to be accessible from the user application through mutation in the lambda function. The calculated bladder volume values in the DynamoDB table are taken using the JavaScript language in the VS Code using the Graphql API and displayed in the application (Figure 3.14b).



(a)



(b)

Figure 3.14: Application: (a) Complete pipeline used in AWS, (b) User Application

3.3 *Complete System & PCB manufacturing*

3.3.1 The proof of the working systems

Every component stm32wb55 discovery board, Max14808 pulser board (PCB), one analog mux (DG408), and a receiver circuit are connected for the whole system test (Figure 3.15). One square transducer is also used instead of using commercial products. Max14808 PCB board is used instead of its evaluation board in this setup. The detailed design steps will be discussed in the next chapter. For testing purposes, a flask of 500 mL is used, as shown in the Figure 3.16. After connecting each component appropriately according to the specifications on the product datasheets, logging the echo timestamps is started and recorded as a CSV file using the ST BLE Sensor application. Detected echo timestamps are transferred to the application via the Bluetooth feature of the MCU. The comparator output of the echo signals is shown in Figure 3.17, and the data log is shown in Figure 3.18.

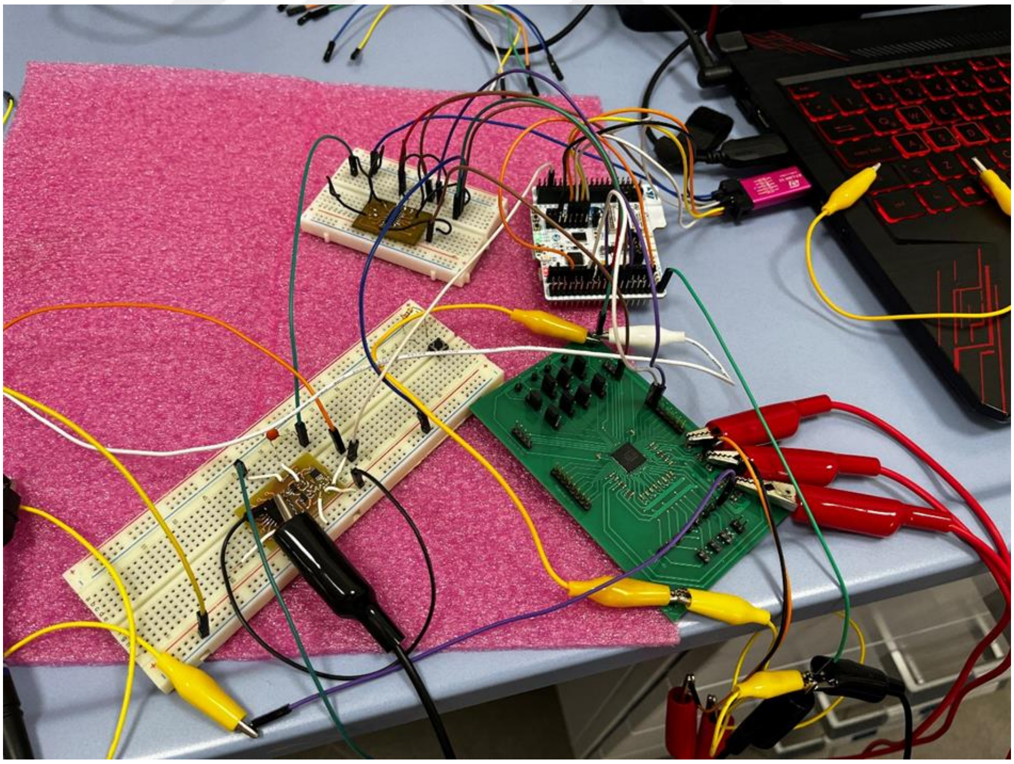


Figure 3.15: Complete working system using the development boards of the components.

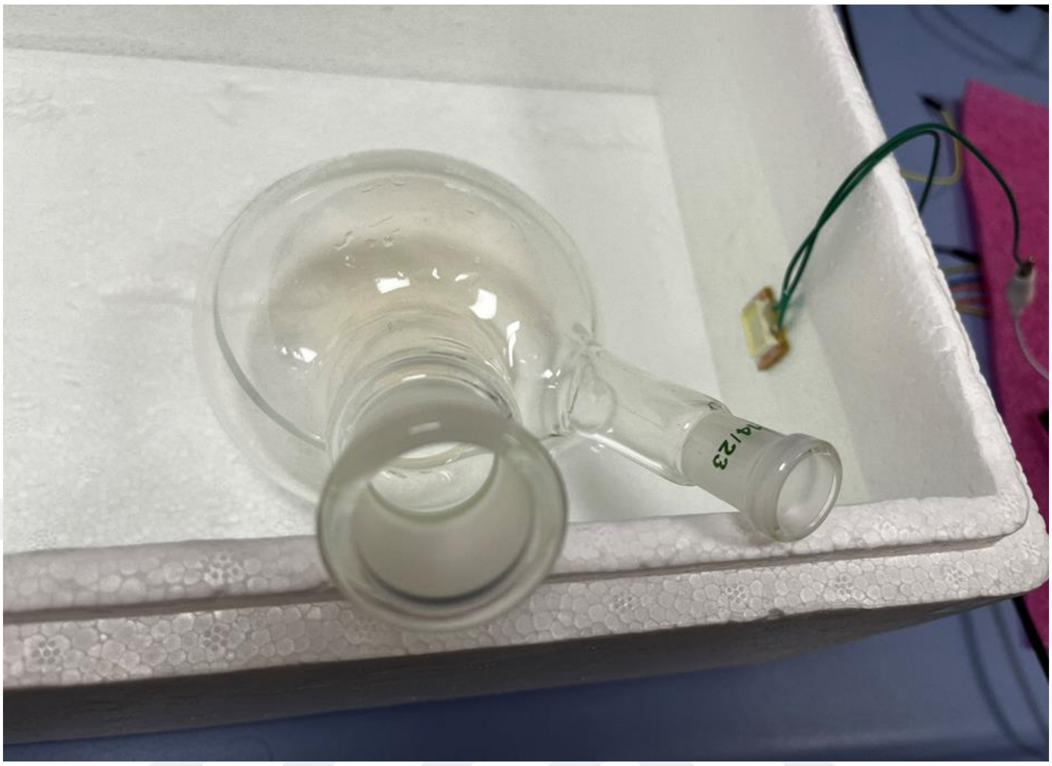


Figure 3.16: A flask of 500 mL and fabricated piezo transducer for testing purpose

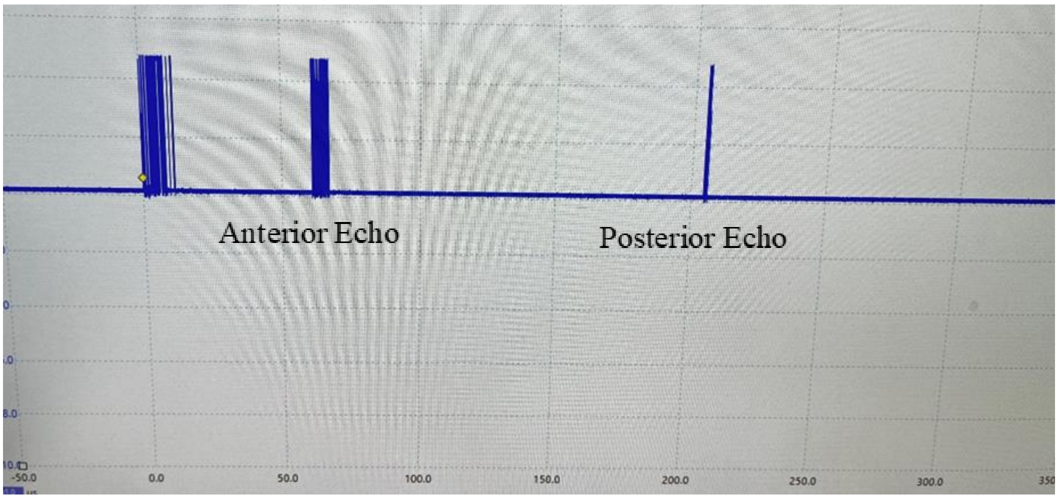


Figure 3.17: The comparator output of the echo signals

157	24/11/2022 16:36:23.321	3665	57.265625
158	24/11/2022 16:36:23.321	3786	59.15625
159	24/11/2022 16:36:23.351	3817	59.640625
160	24/11/2022 16:36:23.351	3878	60.59375
161	24/11/2022 16:36:23.381	12809	200.140625
162	24/11/2022 16:36:23.381	12837	200.578125
163	24/11/2022 16:36:23.411	12867	201.046875
164	24/11/2022 16:36:23.411	959988	14999.8125
165	24/11/2022 16:36:23.441	57	0.890625
166	24/11/2022 16:36:23.441	27	0.421875
167	24/11/2022 16:36:23.471	87	1.359375
168	24/11/2022 16:36:23.471	116	1.8125
169	24/11/2022 16:36:23.501	286	4.46875
170	24/11/2022 16:36:23.501	147	2.296875
171	24/11/2022 16:36:23.530	525	8.203125
172	24/11/2022 16:36:23.561	3846	60.09375
173	24/11/2022 16:36:23.561	3664	57.25
174	24/11/2022 16:36:23.561	3692	57.6875
175	24/11/2022 16:36:23.590	3875	60.546875
176	24/11/2022 16:36:23.621	3935	61.484375
177	24/11/2022 16:36:23.621	3905	61.015625
178	24/11/2022 16:36:23.651	3965	61.953125
179	24/11/2022 16:36:23.651	12721	198.765625
180	24/11/2022 16:36:23.681	960002	15000.03125
181	24/11/2022 16:36:23.681	959970	14999.53125

Figure 3.18: Data logs of the echo signals displayed on the excel table.

According to the comparator output on the picoscope, the anterior and posterior echoes are shown in approximately 60 and 200 microseconds, respectively. These values were also detected from the electronic board with high accuracy and sent to the application via Bluetooth. The raw echo signals are shown in the second column. On test day, the operating clock frequency of the MCU is 64 MHz. The echo timestamps are calculated as follows; every tick of the MCU is 1/64, and the sample raw echo time is 12809 (row of 161), making the actual time 200,14 (1/64*raw echo timestamp). Once these values are sent to the AWS, the bladder volume is measured using the time-of-flight value, which helps to determine the distance from the transducers. Further bladder volume calculation uses the least square methods to find the best-fit sphere according to the detected echo timestamp points.

3.3.2 PCB manufacturing

The system's wearability is the project's crucial point to enable continuous monitoring of the bladder volume. Therefore, first, the schematic and layout design of the high-voltage pulser (Max14808) is made to ensure its operability since it requires a high-voltage and well-designed circuit. The schematic and layout design can be found in the Appendix A3. The printed board is shown in Figure 3.19. Three types of capacitors are used during the design, which are 0.1 uF, 1uF, and 10 uF. The small value capacitors are placed as close as possible to the chips to eliminate high-frequency noises. The high-value capacitors are used to eliminate noises on the power lines providing smoother signals to the electronic components.

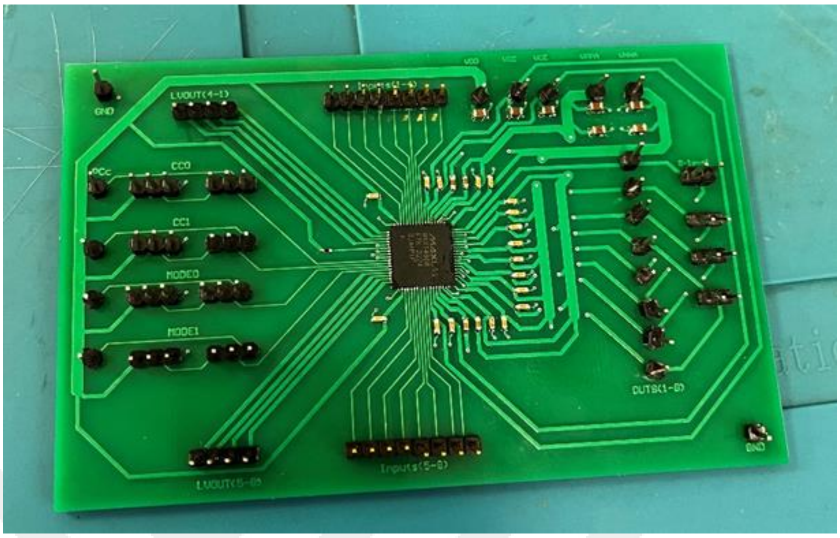


Figure 3.19: The printed board for Max14808 chip

After ensuring the operability of the high voltage pulser, the complete system design, including the stm32wb55rgv6 MCU, analog muxes (DG409, DG408), opAmps, and filters, are realized for miniaturization and wearability (Figure 3.20). The electronic system is designed on a 47.1 x 63.8 mm double-sided printed circuit board (PCB) using FR4. The top and bottom layer layouts are shown in Appendix A4. During the circuit design, ground layer size is observed to affect the components' performance significantly. Therefore, the ground layer area is designed as large as possible to improve the performance of the components. The Stm32wb55rgv6 MCU is used to transmit, collect, and process the data and control the analog multiplexers, and a high voltage pulser (MAX14808) is used to generate excitation signals. A 32 MHz crystal oscillator and a 2.4 GHz Bluetooth antenna are used as the MCU manufacturer suggests. All components used in the board are listed in Appendix A5. The design for the stm32wb55rgv6 found in Appendix A6 is made according to the product datasheet. The MCU's serial wire connection consists of GND, NRST, SWDIO, SWCLK, and VDD (3.3V) via an st-link debugger to directly upload the code to the chip.

Various types of machines are used for printed circuit board fabrication. First, drilling processes are realized by the LPKF Protomat E44 mechanical PCB cutter. The alignment is crucial since the board has two sides, meaning a double-sided circuit board, so the process begins; fiducial holes are drilled, then vias and holes. Then, the LPKF Contac S4

PCB through-hole plating and LPKF SMT ProtoFlow S4 reflow oven systems fill the copper inside the vias and holes and set the top-bottom electrical connections, respectively. After the filling steps, the LPKF Protolaser U4 is used to pattern the PCB. In the final step, the patterned PCB is separated from the FR4 PCB sheet using either LPKF Protolaser U4 or an LPKF Protomat E44 mechanical PCB cutter.

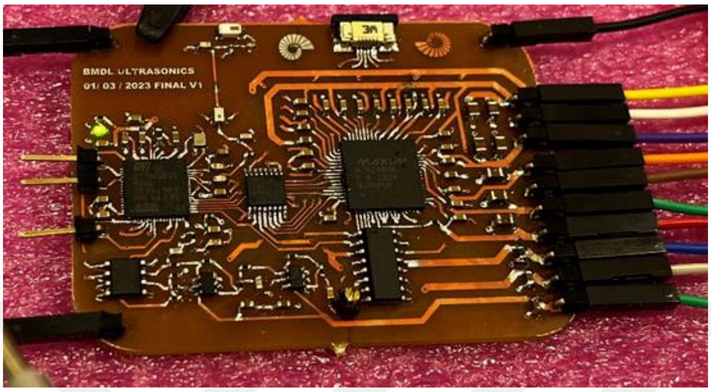


Figure 3.20: Fabricated and miniaturized electronic board

The electronic module of the system consumes low power while operating, which is highly appropriate in wearable ultrasound technology. The average current consumption of the electronic components is shown in Table 3.1.

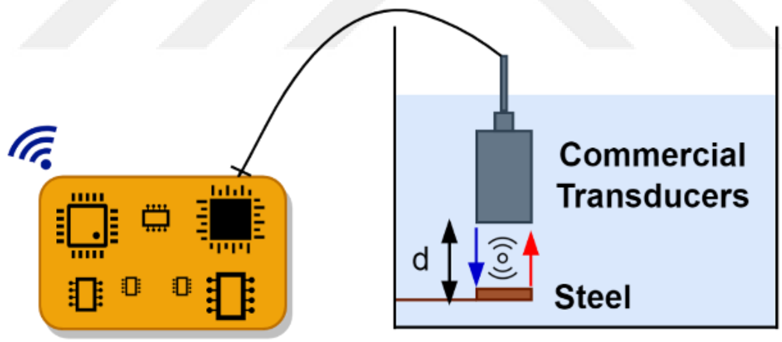
Table 3.1: The average current consumption of the electronic components

Circuit Components	**Average Current (mA)**
Stm32wb Bluetooth Chip	8 mA
Dg409 Analog Mux	5 mA
Max14808 HV Pulser	30 mA
Dg408 Analog Mux	5 mA
OpAmps	10 mA
Comparator	12 mA

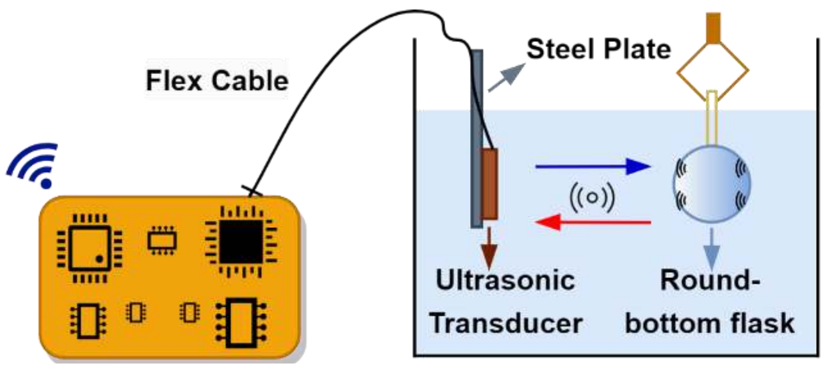
3.4 Tests

3.4.1 In Vitro tests

An assessment in vitro is carried out to gauge the effectiveness of the electronics module. Before using the custom ultrasonic patch, several commercially available transducers are used to conduct a pulse-echo experiment in a water tank. The code in the MCU is adjusted to generate the excitation signals that stimulate the multiple ultrasonic transducers at their specific frequencies. The captured echo timestamps are then transmitted wirelessly to the custom application from the MCU. In order to mimic the distances between the anterior and posterior walls, a steel reflector is placed at various distances from the transducers shown in the Figure 3.21a. The results, illustrated in Table 3.2, demonstrate the successful wireless pulse-echo measurements performed by the UBVM electronics throughout the frequency range (1,2,3,5,8 MHz) relevant to medical applications. After verifying the functionality of the UBVM electronics module, the complete ultrasonic patch with the electronics module underwent testing within a water tank, utilizing round-bottom flasks shaped like spheres to replicate the characteristics of a bladder (Figure 3.21b). The measurements are performed in deionized water, which possesses an acoustic impedance comparable to tissue [51]. Two round-bottom flasks with 250 mL and 500 mL volumes are employed as the examination objects. As depicted in Table 3.3, the UBVM device estimated the volumes of the flasks to be 274 mL and 547 mL, respectively, exhibiting an error rate of less than 10%.



(a)



(b)

Figure 3.21: In Vitro tests: (a) using commercial transducers and a steel reflector, (b) using a custom transducer and a round-bottom flask

Table 3.2: Testing results of the electronic board with commercial transducers and a steel reflector

Operating Frequency	**Distance**	**Distance**
MHz	**cm**	**cm**
	Raw Echo	Phone Data
1	4,22	4,33
1	14,28	14,37
2	4,25	4,16
2	14,07	13,8
3	3,77	3,89
3	13,91	13,94
5	4,01	3,96
5	13,78	13,78
8	4,03	4
8	13,95	13,97

Table 3.3: Testing results of the electronic board with a custom transducer and round-bottom flasks

Flask1		**Flask2**	
Real Volume	**UBVM**	**Real Volume**	**UBVM**
250 mL	274 mL	500 mL	547 mL

3.4.2 In Vivo tests

After the successful results of in vitro experiments, in vivo experiments are performed on volunteers to demonstrate the system's operability. Since the system's primary purpose is continuous monitoring of the bladder volume, a custom app, at a specific time interval, shows the bladder volume measurement data on the phone. A custom app also advises the user about the ideal placement of the ultrasonic patch on the lower abdomen. In the ideal scenario, the custom app should receive eight echo signals (four from the anterior, the other four from the posterior) to calculate the bladder volume accurately. The minimum number of received echo signals should be five due to the requirement for an accurate spherical fitting; otherwise, the app warns the user to change the ultrasonic patch position.

Table 3.4 shows a volunteer's complete cycle of micturition. The volunteer started the experiment with an empty bladder and placed the ultrasonic patch on the lower abdomen (Figure 3.22) with the help of the app guidance. When the optimal position is realized, the recording is started; the app displays the bladder volume at every 30 minutes auto measurement. Table 3.4 shows that as time passes, the distance between the anterior and posterior walls of the bladder increases, and the bladder volume increases accordingly.

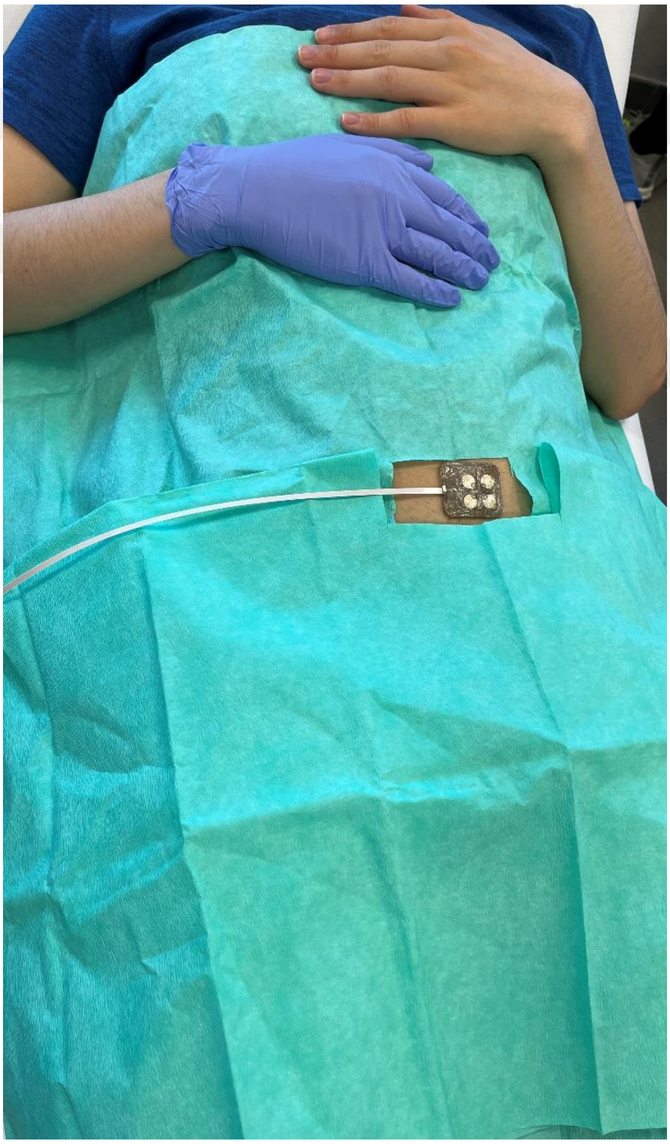
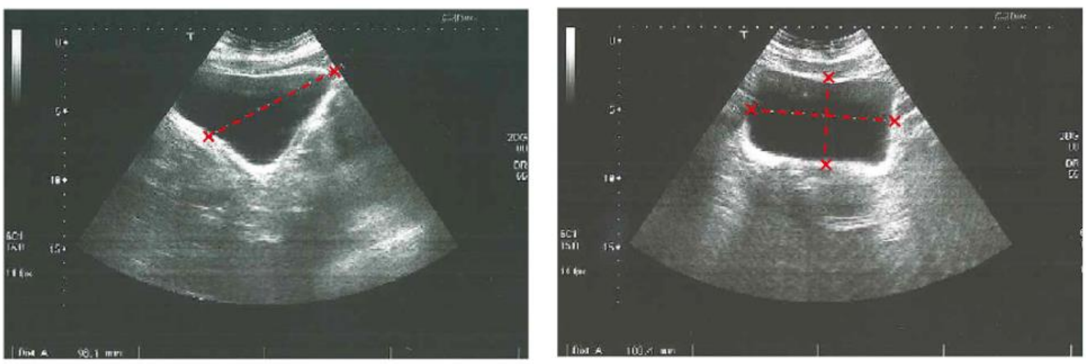


Figure 3.22: The ultrasonic patch on the lower abdomen

Table 3.4: Complete micturition cycle of a volunteer

Measurement	UBVM Volume	Ant-Post Distance
	mL	mm
1	17	15,9956
2	62	21,2965
3	120	28,23886
4	153	38,31529
5	220	57,81993
6	261	81,74915

For further validation of the system, five healthy volunteers (three females and two males) are joined to the experiment. Before the experiment, each volunteer's bladder is examined, and volume is measured by the experienced urologist using a commercial ultrasound imaging system (Figure 3.23a and Figure 3.23b). Table 3.5 shows the estimated bladder volume using UBVM versus measurements made by the commercial ultrasound imaging system. All ultrasound images of the volunteers are attached in the Appendix A7. According to the results, the UBVM performs well in estimating the bladder volume ranging from 84 mL to 800 mL of the volunteers. The UBVM device has also demonstrated good operability in different bladder shapes and sizes.



(a) (b)

Figure 3.23: Ultrasonic images for bladder volume measurement: (a) side of the bladder (b) height and width of the bladder

Table 3.5: Commercial Ultrasound System vs UBVM

	Ultrasound	UBVM
Volunteer	**Volume**	**Volume**
	mL	**mL**
A	84	100
B	175	143
C	201	199
D	260	280
E	310	314
F	310	305
G	322,5	325
H	800	708

Chapter 4:
CONCLUSION

An electronic module-integrated flexible ultrasonic system that enables continuous monitoring of bladder volume is demonstrated. The ultrasonic system consists of a flexible ultrasonic patch, highly conformable to the user's daily life. An integrated electronic module enables to user to monitor bladder volumes continuously using a phone with the help of Bluetooth technology. The electronic system is miniaturized and consumes low power, which is highly demanded by wearable technology. Simple fabrication methods of the ultrasonic patch enable low driving voltages (30 Vpp), so there is no need to use high driving voltages. The UBVM device can give insight into the current wearable ultrasound technology to open up new application areas.

Future research will concentrate on reducing the overall size of the system. Increasing the transducers' frequency provide an increase in the number of transducers in the same area while reducing the dimensions. The electronic module of the UBVM can be further miniaturized with the careful design of the PCB. Overcoming these challenges is expected to make the UBVM device highly beneficial in addressing different scenarios, including lower urinary tract symptoms and postoperative urinary retention.

BIBLIOGRAPHY

[1] GBD 2019 Benign Prostatic Hyperplasia Collaborators (2022). The global, regional, and national burden of benign prostatic hyperplasia in 204 countries and territories from 2000 to 2019: a systematic analysis for the Global Burden of Disease Study 2019. The lancet. Healthy longevity, 3(11), e754–e776.

[2] Phé, V., Chartier-Kastler, E., & Panicker, J. N. (2016). Management of neurogenic bladder in patients with multiple sclerosis. Nature reviews. Urology, 13(5), 275–288.

[3] Panicker, J. N., Fowler, C. J., & Kessler, T. M. (2015). Lower urinary tract dysfunction in the neurological patient: clinical assessment and management. The Lancet. Neurology, 14(7), 720–732.

[4] D'Ancona, C., Haylen, B., Oelke, M., Abranches-Monteiro, L., Arnold, E., Goldman, H., Hamid, R., Homma, Y., Marcelissen, T., Rademakers, K., Schizas, A., Singla, A., Soto, I., Tse, V., de Wachter, S., Herschorn, S., & Standardisation Steering Committee ICS and the ICS Working Group on Terminology for Male Lower Urinary Tract & Pelvic Floor Symptoms and Dysfunction (2019). The International Continence Society (ICS) report on the terminology for adult male lower urinary tract and pelvic floor symptoms and dysfunction. Neurourology and urodynamics, 38(2), 433–477.

[5] Peterson, N. T., & Vezina, C. M. (2022). Male Lower Urinary Tract Dysfunction: An Underrepresented Endpoint in Toxicology Research. Toxics, 10(2), 89.

[6] Irwin, D. E., Kopp, Z. S., Agatep, B., Milsom, I., & Abrams, P. (2011). Worldwide prevalence estimates of lower urinary tract symptoms, overactive bladder, urinary incontinence and bladder outlet obstruction. BJU international, 108(7), 1132–1138.

[7] Launer, B. M., McVary, K. T., Ricke, W. A., & Lloyd, G. L. (2021). The rising worldwide impact of benign prostatic hyperplasia. BJU international, 127(6), 722–728.

[8] Nieuwhof-Leppink, A. J., Schroeder, R. P. J., van de Putte, E. M., de Jong, T. P. V. M., & Schappin, R. (2019). Daytime urinary incontinence in children and adolescents. The Lancet. Child & adolescent health, 3(7), 492–501.

[9] Gammie, A., Speich, J. E., Damaser, M. S., Gajewski, J. B., Abrams, P., Rosier, P. F. W. M., Arlandis, S., Tarcan, T., & Finazzi Agrò, E. (2020). What developments are needed to achieve less-invasive urodynamics? ICI-RS 2019. Neurourology and urodynamics, 39 Suppl 3, S36–S42.

[10] Smith, A., Bevan, D., Douglas, H. R., & James, D. (2013). Management of urinary incontinence in women: summary of updated NICE guidance. BMJ (Clinical research ed.), 347, f5170.

[11] Bright, E., Cotterill, N., Drake, M., & Abrams, P. (2014). Developing and validating the International Consultation on Incontinence Questionnaire bladder diary. European urology, 66(2), 294–300.

[12] Mickle, A. D., Won, S. M., Noh, K. N., Yoon, J., Meacham, K. W., Xue, Y., McIlvried, L. A., Copits, B. A., Samineni, V. K., Crawford, K. E., Kim, D. H., Srivastava, P., Kim, B. H., Min, S., Shiuan, Y., Yun, Y., Payne, M. A., Zhang, J., Jang, H., Li, Y., … Rogers, J. A. (2019). A wireless closed-loop system for optogenetic peripheral neuromodulation. Nature, 565(7739), 361–365.

[13] Hannah, S., Brige, P., Ravichandran, A. & Ramuz, M. (2019). Conformable, Stretchable Sensor to Record Bladder Wall Stretch. ACS Omega 4, 1907–1915.

[14] Stauffer, F. et al. (2018). Soft Electronic Strain Sensor with Chipless Wireless Readout: Toward Real-Time Monitoring of Bladder Volume. Adv Mater Technol 3, 1800031.

[15] Mandal, S., Dehghanzadeh, P., Zamani, H. & Shaik, S. (2022). A Wireless Implantable Microsystem for Real-Time Bladder Volume Monitoring.

[16] Gerber, M. T. (2015).Impedance-based bladder sensing.

[17] Thüroff, J. W. et al. (2011). EAU Guidelines on Urinary Incontinence. Eur Urol 59, 387–400.

[18] Abelson, B., Majerus, S., Sun, D., Gill, B. C., Versi, E., & Damaser, M. S. (2019). Ambulatory urodynamic monitoring: state of the art and future directions. Nature reviews. Urology, 16(5), 291–301.

[19] SENS-U KIDS Bladder Sensor is our first proof of wearable ultrasound care. https://novioscan.com/.

[20] DFree - US site wearable bladder scanner for incontinence. https://www.dfreeus.biz/.

[21] Lilium Otsuka Co., Ltd. https://www.lilium.otsuka/en/.

[22] van Leuteren, P. G., Nieuwhof-Leppink, A. J. & Dik, P. (2019). SENS-U: clinical evaluation of a full-bladder notification – a pilot study. J Pediatr Urol 15, 381.e1-381.e5.

[23] Hofstetter, S. et al. (2023). Dfree ultrasonic sensor in supporting quality of life and patient satisfaction with bladder dysfunction. International Journal of Urological Nursing 17, 62–69.

[24] Kamei, J., Watanabe, D., Homma, Y., Kume, H. & Igawa, Y. (2019). Feasibility of approximate measurement of bladder volume in male patients using the Lilium α-200 portable ultrasound bladder scanner. LUTS: Lower Urinary Tract Symptoms 11, 169–173.

[25] Niestoruk, L. et al. (2012). A concept for wearable long-term urinary bladder monitoring with ultrasound. Feasibility study. EDERC 2012 - Proceedings of the 5th European DSP in Education and Research Conference 134–138.

[26] Semproni, F., Iacovacci, V. & Menciassi, A. (2022). Bladder Monitoring Systems: State of the Art and Future Perspectives. IEEE Access 10, 125626–125651.

[27] Kiczek, B. et al. (2022). A wearable ultrasonic bladder monitoring device. Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM 886–888.

[28] Jo, H. G. et al. (2021) Forward-looking ultrasound wearable scanner system for estimation of urinary bladder volume. Sensors 21, 5445.

[29] Fournelle, M. et al. (2021) Portable Ultrasound Research System for Use in Automated Bladder Monitoring with Machine-Learning-Based Segmentation. Sensors 2021, Vol. 21, Page 6481 21, 6481.

[30] Gao, Q. et al. (2021). Design and Development of a Bladder Volume Determination Device Based on A-mode Ultrasound. 2021 IEEE International Conference on Mechatronics and Automation, ICMA 2021 170–175.

[31] Kuru, K. et al. (2020) Intelligent autonomous treatment of bedwetting using non-invasive wearable advanced mechatronics systems and MEMS sensors: Intelligent autonomous bladder monitoring to treat NE. Med Biol Eng Comput 58, 943–965.

[32] Gaubert, V., Gidik, H. & Koncar, V. (2020). Smart underwear, incorporating textrodes, to estimate the bladder volume: proof of concept on a test bench. Smart Mater Struct 29, 085028.

[33] Shin, S. C. et al. (2017). Continuous bladder volume monitoring system for wearable applications. Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 4435–4438.

[34] Schlebusch, T., Nienke, S., Leonhardt, S. & Walter, M. (2014). Bladder volume estimation from electrical impedance tomography. Physiol Meas 35, 1813.

[35] Reichmuth, M., Schurle, S. & Magno, M. (2020). A Non-invasive Wearable Bioimpedance System to Wirelessly Monitor Bladder Filling. Proceedings of the 2020 Design, Automation and Test in Europe Conference and Exhibition, DATE 2020 338–341.

[36] Fong, D., Alcantar, A. V., Gupta, P., Kurzrock, E. & Ghiasi, S. (2018). Non-invasive bladder volume sensing for neurogenic bladder dysfunction management. 2018 IEEE 15th International Conference on Wearable and Implantable Body Sensor Networks, BSN 2018 2018-January, 82–85.

[37] Molavi, B., Shadgan, B., Macnab, A. J. & Dumont, G. A. (2014). Noninvasive optical monitoring of bladder filling to capacity using a wireless near infrared spectroscopy device. IEEE Trans Biomed Circuits Syst 8, 325–333.

[38] Fechner, P., König, F., Kratsch, W., Lockl, J., & Röglinger, M. (2023). Near-Infrared Spectroscopy for Bladder Monitoring: A Machine Learning Approach. ACM Trans Manag Inf Syst 14, 1–23.

[39] Koven, A. & Herschorn, S. (2022). NIRS: Past, Present, and Future in Functional Urology. Curr Bladder Dysfunct Rep 17, 241–249.

[40] Leonhäuser, D. et al. (2018). Evaluation of electrical impedance tomography for determination of urinary bladder volume: Comparison with standard ultrasound methods in healthy volunteers. Biomed Eng Online 17, 1–13.

[41] Hu, H. et al. (2018). Stretchable ultrasonic transducer arrays for three-dimensional imaging on complex surfaces. Sci Adv 4.

[42] Wang, C., Li, X., Hu, H., Zhang, L., Huang, Z., Lin, M., Zhang, Z., Yin, Z., Huang, B., Gong, H., Bhaskaran, S., Gu, Y., Makihata, M., Guo, Y., Lei, Y., Chen, Y., Wang, C., Li, Y., Zhang, T., Chen, Z., … Xu, S. (2018). Monitoring of the central blood pressure waveform via a conformal ultrasonic device. Nature biomedical engineering, 2(9), 687–695.

[43] Wang, F. et al. (2021). Flexible Doppler ultrasound device for the monitoring of blood flow velocity. Sci Adv 7, 9283–9310.

[44] Wang, C., Qi, B., Lin, M., Zhang, Z., Makihata, M., Liu, B., Zhou, S., Huang, Y. H., Hu, H., Gu, Y., Chen, Y., Lei, Y., Lee, T., Chien, S., Jang, K. I., Kistler, E. B., & Xu, S. (2021). Continuous monitoring of deep-tissue haemodynamics with stretchable ultrasonic phased arrays. Nature biomedical engineering, 5(7), 749–758.

[45] Wang, C. et al. (2022). Bioadhesive ultrasound for long-term continuous imaging of diverse organs. Science (1979) 377, 517–523.

[46] Hu, H., Huang, H., Li, M., Gao, X., Yin, L., Qi, R., Wu, R. S., Chen, X., Ma, Y., Shi, K., Li, C., Maus, T. M., Huang, B., Lu, C., Lin, M., Zhou, S., Lou, Z., Gu, Y., Chen, Y., Lei, Y., … Xu, S. (2023). A wearable cardiac ultrasound imager. Nature, 613(7945), 667–675.

[47] Hu, H. et al. (2023) Stretchable ultrasonic arrays for the three-dimensional mapping of the modulus of deep tissue. Nature Biomedical Engineering 2023 1–14.

[48] Yu, C.-C. et al. (2023) A Conformable Ultrasound Patch for Cavitation-Enhanced Transdermal Cosmeceutical Delivery. Advanced Materials 2300066.

[49] Lin, M., Zhang, Z., Gao, X., Bian, Y., Wu, R. S., Park, G., Lou, Z., Zhang, Z., Xu, X., Chen, X., Kang, A., Yang, X., Yue, W., Yin, L., Wang, C., Qi, B., Zhou, S., Hu, H., Huang, H., Li, M., … Xu, S. (2023). A fully integrated wearable ultrasound system to monitor deep tissues in moving subjects. Nature biotechnology, 10.1038/s41587-023-01800-0. Advance online publication.

[50] AWS AMPLIFY (accessed June 1,2023). React Native Application. https://docs.amplify.aws/start/getting-started/setup/q/integration/react-native/#create-a-new-project

[51] Kino, G. S. (1987). Acoustic Waves: Devices, Imaging, and Analog Signal Processing. Englewood Cliffs 100, 103.

Appendix A:
Electronic System Design Layouts

A1: The whole code written using STM32 Cube Ide for the MCU

A1.1: main.c

```
/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_DMA_Init();
MX_RF_Init();
MX_RTC_Init();
MX_RNG_Init();
MX_TIM1_Init();
MX_TIM16_Init();
MX_TIM2_Init();
MX_TIM17_Init();
/* USER CODE BEGIN 2 */

TIM16->CCR1=5;
HAL_TIM_PWM_Start(&htim16, TIM_CHANNEL_1);
TIM17->CCR1=5;
HAL_TIM_PWM_Start(&htim17, TIM_CHANNEL_1);
HAL_TIM_IC_Start_IT(&htim1, TIM_CHANNEL_3);
HAL_TIM_IC_Start_IT(&htim2, TIM_CHANNEL_3);
// HAL_TIM_Base_Start_IT(&htim1);

/* USER CODE END 2 */
```

A1.2 app_entry.c

```
void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef*htim)
{
    UNUSED(htim);
    static uint8_t loop_counter=0;
    if(htim->Instance==TIM1)
    {
            switch(htim->Channel)
            {
            case HAL_TIM_ACTIVE_CHANNEL_3:
            {
                    if((TIM1->CCER & TIM_CCER_CC3P)==0){
                            switch(trigger_cycle)
                            {
                            case 1:
                            {
                                    if(num_trigger==1)
                                    {
                                    loop_counter++;
                                    num_trigger=0;
                                    array_is_full=send_receive_data();
                                    }
                                    if(array_is_full==1)
                                    {
                                    piezo_id++;
                                    APP_BLE_Key_Button1_Action();
                                    }
                                    if((array_is_full & finished_data_transfer) ==1)
                                    {
                                            array_is_full=0;
                                            if (loop_counter<4) {num_trigger=1;}
                                            else {loop_counter=0;num_trigger=2;
trigger_cycle=2; current_buffer = (current_buffer+1) % Num_Buffers;piezo_id=2000; }
```

```
}
break;
}
case 2:
{
if(num_trigger==2)
{
loop_counter++;
num_trigger=0;
array_is_full=send_receive_data();
}
if(array_is_full==1)
{
piezo_id++;
APP_BLE_Key_Button1_Action();
}
if((array_is_full & finished_data_transfer) ==1)
{
array_is_full=0;
if (loop_counter<4) {num_trigger=2;}
else {loop_counter=0;num_trigger=3;
trigger_cycle=3;  current_buffer = (current_buffer+1) % Num_Buffers;piezo_id=3000;}
}
break;
}
case 3:
{
if(num_trigger==3)
{
loop_counter++;
num_trigger=0;
array_is_full=send_receive_data();
}
if(array_is_full==1)
```

```
                {
                piezo_id++;
                APP_BLE_Key_Button1_Action();
                }
                if((array_is_full & finished_data_transfer)==1)
                {
                        array_is_full=0;
                        if (loop_counter<4) {num_trigger=3;}
                        else {loop_counter=0;num_trigger=4;
trigger_cycle=4;  current_buffer = (current_buffer+1) % Num_Buffers;piezo_id=4000;}
                }
                break;
        }
        case 4:
        {
                if(num_trigger==4)
                {
                loop_counter++;
                num_trigger=0;
                array_is_full=send_receive_data();
                }
                if(array_is_full==1)
                {
                piezo_id++;
                APP_BLE_Key_Button1_Action();
                }
                if((array_is_full & finished_data_transfer) ==1)
                {
                        array_is_full=0;
                        if (loop_counter<4) {num_trigger=4;}
                        else {loop_counter=0;num_trigger=1;
trigger_cycle=1;  current_buffer = (current_buffer+1) % Num_Buffers;piezo_id=1000;}
                }
                break;
```

```
                    }
                }
            }
        }
        }
    }
}

int send_receive_data()
{

    if(current_buffer==0){ HAL_GPIO_WritePin(GPIOA,
A0_Pin|A1_Pin,GPIO_PIN_RESET); }                              //S1
    if(current_buffer==1){ HAL_GPIO_WritePin(GPIOA,A0_Pin,
GPIO_PIN_SET); HAL_GPIO_WritePin(GPIOA,A1_Pin, GPIO_PIN_RESET);}
//S2
    if(current_buffer==2){ HAL_GPIO_WritePin(GPIOA,A0_Pin,
GPIO_PIN_RESET); HAL_GPIO_WritePin(GPIOA,A1_Pin, GPIO_PIN_SET);}
     //S3
    if(current_buffer==3){ HAL_GPIO_WritePin(GPIOA,
A0_Pin|A1_Pin,GPIO_PIN_SET);}
                                                  //S4
    HAL_TIM_IC_Stop_DMA(&htim2, TIM_CHANNEL_1);
    HAL_TIM_IC_Start_DMA(&htim2, TIM_CHANNEL_1,
echo_timestamps[current_buffer], BUFFER_SIZE);
    array_is_full = 1;
    HAL_TIM_Base_Start(&htim1);
    HAL_TIM_Base_Init(&htim2);
    HAL_TIM_OnePulse_Start(&htim1, TIM_CHANNEL_1);
    HAL_TIM_OnePulse_Start(&htim1, TIM_CHANNEL_2);
    HAL_TIM_Base_Stop(&htim1);

    return array_is_full;
}
```

A1.3 p2p_server_app.c

```
/* Includes ------------------------------------------------------------------*/
#include "main.h"
#include "app_common.h"
#include "dbg_trace.h"
#include "ble.h"
#include "p2p_server_app.h"
#include "stm32_seq.h"

/* Private includes ----------------------------------------------------------*/
/* USER CODE BEGIN Includes */
#include "stm32wbxx_nucleo.h"
//#include "app_entry.h"

//extern uint32_t rise1Data[100];
//extern uint32_t rise2Data[100];
//extern uint32_t rise3Data[100];
//extern uint32_t rise4Data[100];
//
//extern uint32_t time_counter;
extern uint8_t  current_buffer;
extern uint32_t echo_timestamps[4][100];
uint8_t finished_data_transfer=0;
extern uint16_t piezo_id;

/* USER CODE END Includes */

/* Private typedef -----------------------------------------------------------*/
/* USER CODE BEGIN PTD */
 typedef struct{
    uint16_t             TimeStamp;
    uint32_t             Value;
 }TEMPLATE_TemperatureCharValue_t;
```

```
typedef struct{
  uint8_t          Device_Led_Selection;
  uint8_t          Led1;
}P2P_LedCharValue_t;

typedef struct{
  uint8_t          Device_Button_Selection;
  uint8_t          ButtonStatus;
}P2P_ButtonCharValue_t;

typedef struct
{
  uint8_t              Notification_Status; /* used to check if P2P Server is enabled to
Notify */
  P2P_LedCharValue_t   LedControl;
  P2P_ButtonCharValue_t ButtonControl;

  TEMPLATE_TemperatureCharValue_t Co;
  uint16_t             ConnectionHandle;
} P2P_Server_App_Context_t;
/* USER CODE END PTD */
///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void P2PS_Send_Notification(void)
{
   P2P_Server_App_Context.Co.TimeStamp=piezo_id;
   P2P_Server_App_Context.Co.Value=echo_timestamps[current_buffer][counter;
   value[0] = P2P_Server_App_Context.Co.TimeStamp;
   value[1] = P2P_Server_App_Context.Co.TimeStamp>>8;
   value[2] = P2P_Server_App_Context.Co.Value;
   value[3] = P2P_Server_App_Context.Co.Value>>8;
   value[4] = P2P_Server_App_Context.Co.Value>>16;
   value[5] = P2P_Server_App_Context.Co.Value>>24;
```

```
        if(counter==99)
        {
                finished_data_transfer=1;
                counter=0;
        }
        else
        {
                finished_data_transfer=0;
                counter++;
        }

       if(P2P_Server_App_Context.Notification_Status){
        APP_DBG_MSG("-- P2P APPLICATION SERVER  : INFORM CLIENT
BUTTON 1 PUSHED \n ");
        APP_DBG_MSG(" \n\r");
        P2PS_STM_App_Update_Char(P2P_NOTIFY_CHAR_UUID, (uint8_t
*)&value);
       } else {
        APP_DBG_MSG("-- P2P APPLICATION SERVER : CAN'T INFORM
CLIENT -  NOTIFICATION DISABLED\n ");
       }
      return;
    }
```

A1.4 p2p_stm.c

```
#define COPY_P2P_SERVICE_UUID(uuid_struct)
COPY_UUID_128(uuid_struct,0x00,0x00,0xfe,0x40,0xcc,0x7a,0x48,0x2a,0x98,0x4a,0x7f,0x2e,0xd5,0xb3,0xe5,0x8f)
#define COPY_P2P_WRITE_CHAR_UUID(uuid_struct)
COPY_UUID_128(uuid_struct,0x00,0x00,0xfe,0x41,0x8e,0x22,0x45,0x41,0x9d,0x4c,0x21,0xed,0xae,0x82,0xed,0x19)
#define COPY_P2P_NOTIFY_UUID(uuid_struct)
COPY_UUID_128(uuid_struct,0x00,0x00,0x80,0x00,0x00,0x01,0x11,0xE1,0xAC,0x36,0x00,0x02,0xA5,0xD5,0xC5,0x1B)

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void P2PS_STM_Init(void)
{

  Char_UUID_t  uuid16;

  /**
   *	Register the event handler to the BLE controller
   */
  SVCCTL_RegisterSvcHandler(PeerToPeer_Event_Handler);

    /**
     *  Peer To Peer Service
     *
     * Max_Attribute_Records = 2*no_of_char + 1
     * service_max_attribute_record = 1 for Peer To Peer service +
     *                                2 for P2P Write characteristic +
     *                                2 for P2P Notify characteristic +
     *                                1 for client char configuration descriptor +
     *
     */
    COPY_P2P_SERVICE_UUID(uuid16.Char_UUID_128);
```

```
aci_gatt_add_service(UUID_TYPE_128,
            (Service_UUID_t *) &uuid16,
            PRIMARY_SERVICE,
            8,
            &(aPeerToPeerContext.PeerToPeerSvcHdle));

/**
 *  Add LED Characteristic
 */
COPY_P2P_WRITE_CHAR_UUID(uuid16.Char_UUID_128);
aci_gatt_add_char(aPeerToPeerContext.PeerToPeerSvcHdle,
            UUID_TYPE_128, &uuid16,
            2,
            CHAR_PROP_WRITE_WITHOUT_RESP|CHAR_PROP_READ,
            ATTR_PERMISSION_NONE,
            GATT_NOTIFY_ATTRIBUTE_WRITE, /* gattEvtMask */
            10, /* encryKeySize */
            1, /* isVariable */
            &(aPeerToPeerContext.P2PWriteClientToServerCharHdle));

/**
 *   Add Button Characteristic
 */
COPY_P2P_NOTIFY_UUID(uuid16.Char_UUID_128);
aci_gatt_add_char(aPeerToPeerContext.PeerToPeerSvcHdle,
            UUID_TYPE_128, &uuid16,
            6,
            CHAR_PROP_NOTIFY,
            ATTR_PERMISSION_NONE,
            GATT_NOTIFY_ATTRIBUTE_WRITE, /* gattEvtMask */
            10, /* encryKeySize */
            1, /* isVariable: 1 */
            &(aPeerToPeerContext.P2PNotifyServerToClientCharHdle));
```

```
#if(BLE_CFG_OTA_REBOOT_CHAR != 0)
    /**
     *  Add Boot Request Characteristic
     */
    aci_gatt_add_char(aPeerToPeerContext.PeerToPeerSvcHdle,
                      BM_UUID_LENGTH,
                      (Char_UUID_t *)BM_REQ_CHAR_UUID,
                      BM_REQ_CHAR_SIZE,
                      CHAR_PROP_WRITE_WITHOUT_RESP,
                      ATTR_PERMISSION_NONE,
                      GATT_NOTIFY_ATTRIBUTE_WRITE,
                      10,
                      0,
                      &(aPeerToPeerContext.RebootReqCharHdle));
#endif
  return;
}
/**
 * @brief  Characteristic update
 * @param  UUID: UUID of the characteristic
 * @param  Service_Instance: Instance of the service to which the characteristic
belongs
 *
 */
tBleStatus P2PS_STM_App_Update_Char(uint16_t UUID, uint8_t *pPayload)
{
  tBleStatus result = BLE_STATUS_INVALID_PARAMS;
  switch(UUID)
  {
    case P2P_NOTIFY_CHAR_UUID:

     result = aci_gatt_update_char_value(aPeerToPeerContext.PeerToPeerSvcHdle,
                          aPeerToPeerContext.P2PNotifyServerToClientCharHdle,
                           0, /* charValOffset */
```

```
                    6, /* charValueLen */
                    (uint8_t *)  pPayload);

      break;
    default:
      break;
  }
  return result;
}/* end P2PS_STM_Init() */
```

A1.5 app_ble.c

```
#if (P2P_SERVER1 != 0)
static const char a_LocalName[] = {AD_TYPE_COMPLETE_LOCAL_NAME ,
'P', '2', 'P', 'S', 'R', 'V', '1'};
uint8_t a_ManufData[14] = {sizeof(a_ManufData)-1,
                    AD_TYPE_MANUFACTURER_SPECIFIC_DATA,
                    0x01,                   /*SKD version */
                    CFG_DEV_ID_P2P_SERVER1,         /* STM32WB - P2P
Server 1*/
                    0x00,                   /* GROUP A Feature */
                    0x00,                   /* GROUP A Feature */
                    0x80,                   /* GROUP B Feature */
                    0x00,                   /* GROUP B Feature */
                    0x00,                   /* BLE MAC start -MSB */
                    0x00,
                    0x00,
                    0x00,
                    0x00,
                    0x00,                   /* BLE MAC stop */
                    };
#endif /* P2P_SERVER1 != 0 */
```

A1.6 app_conf.h

```
#if defined(STM32WB5Mxx)
  #define CFG_BLE_LSE_SOURCE
(SHCI_C2_BLE_INIT_CFG_BLE_LSE_NOCALIB |
SHCI_C2_BLE_INIT_CFG_BLE_LSE_MOD5MM_DEV)
#else
#define CFG_BLE_LSE_SOURCE
(SHCI_C2_BLE_INIT_CFG_BLE_LSE_CALIB |
SHCI_C2_BLE_INIT_CFG_BLE_LSE_OTHER_DEV)
#endif

#define CFG_RTCCLK_DIV            (16)
#define CFG_RTC_WUCKSEL_DIVIDER   (0)
#define CFG_RTC_ASYNCH_PRESCALER  (127)
#define CFG_RTC_SYNCH_PRESCALER   (7812)

#define CFG_TS_TICK_VAL           DIVR( (CFG_RTCCLK_DIV * 1000000),
HSE_VALUE/32 )
#define CFG_TS_TICK_VAL_PS        DIVR( ((uint64_t)CFG_RTCCLK_DIV * 1e12),
(uint64_t)LSE_VALUE )
```

A1.7 stm32wbxx_hal_msp.c

```
/**
* @brief RTC MSP Initialization
* This function configures the hardware resources used in this example
* @param hrtc: RTC handle pointer
* @retval None
*/
void HAL_RTC_MspInit(RTC_HandleTypeDef* hrtc)
{
 RCC_PeriphCLKInitTypeDef PeriphClkInitStruct = {0};
 if(hrtc->Instance==RTC)
 {
 /* USER CODE BEGIN RTC_MspInit 0 */
     HAL_PWR_EnableBkUpAccess();




     HAL_PWR_EnableBkUpAccess();

     __HAL_RCC_RTC_CONFIG(RCC_RTCCLKSOURCE_HSE_DIV32);

 /* USER CODE END RTC_MspInit 0 */

 /** Initializes the peripherals clock
 */
   PeriphClkInitStruct.PeriphClockSelection = RCC_PERIPHCLK_RTC;
   PeriphClkInitStruct.RTCClockSelection =
RCC_RTCCLKSOURCE_HSE_DIV32;
   if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInitStruct) != HAL_OK)
   {
     Error_Handler();
```

```
    }
    /* Peripheral clock enable */
    __HAL_RCC_RTC_ENABLE();
    __HAL_RCC_RTCAPB_CLK_ENABLE();
    /* RTC interrupt Init */
    HAL_NVIC_SetPriority(RTC_WKUP_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(RTC_WKUP_IRQn);
  /* USER CODE BEGIN RTC_MspInit 1 */
    HAL_RTCEx_EnableBypassShadow(hrtc);
  /* USER CODE END RTC_MspInit 1 */
  }
}
/**
* @brief RTC MSP De-Initialization
* This function freeze the hardware resources used in this example
* @param hrtc: RTC handle pointer
* @retval None
*/
void HAL_RTC_MspDeInit(RTC_HandleTypeDef* hrtc)
{
  if(hrtc->Instance==RTC)
  {
  /* USER CODE BEGIN RTC_MspDeInit 0 */
  /* USER CODE END RTC_MspDeInit 0 */
    /* Peripheral clock disable */
    __HAL_RCC_RTC_DISABLE();
    __HAL_RCC_RTCAPB_CLK_DISABLE();
    /* RTC interrupt DeInit */
    HAL_NVIC_DisableIRQ(RTC_WKUP_IRQn);
  /* USER CODE BEGIN RTC_MspDeInit 1 */
  /* USER CODE END RTC_MspDeInit 1 */}
}
```

A2: Different pulse amplitudes provided from Max14808 evaluation board

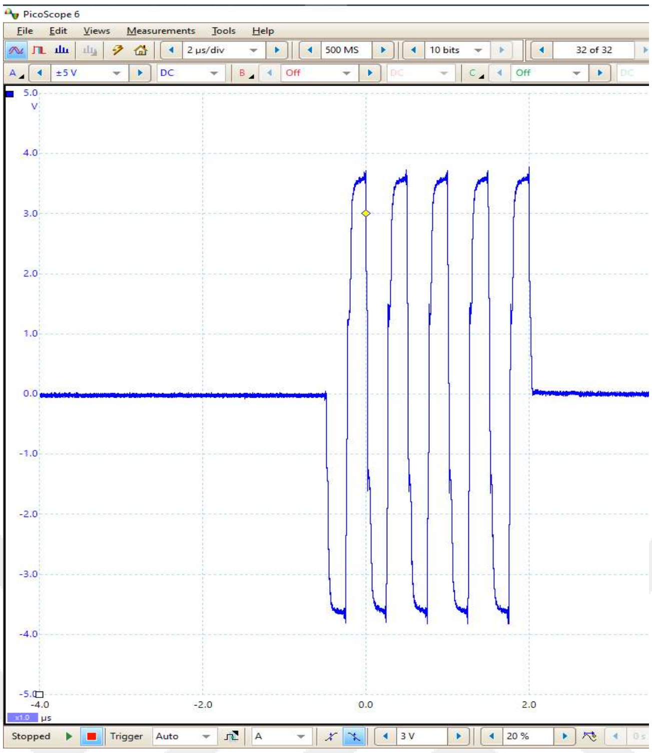


Figure A2. 1: 10V peak-to-peak amplitude of pulses

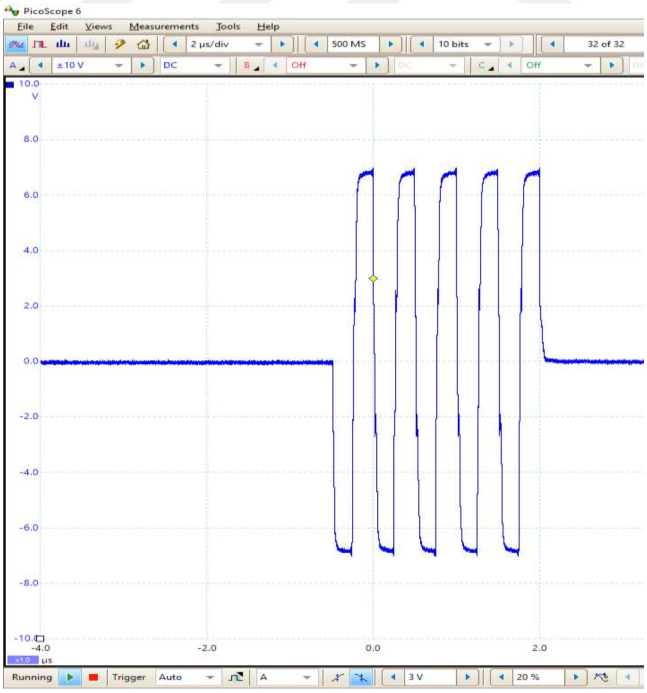


Figure A2. 2: 16V peak-to-peak amplitude of pulses

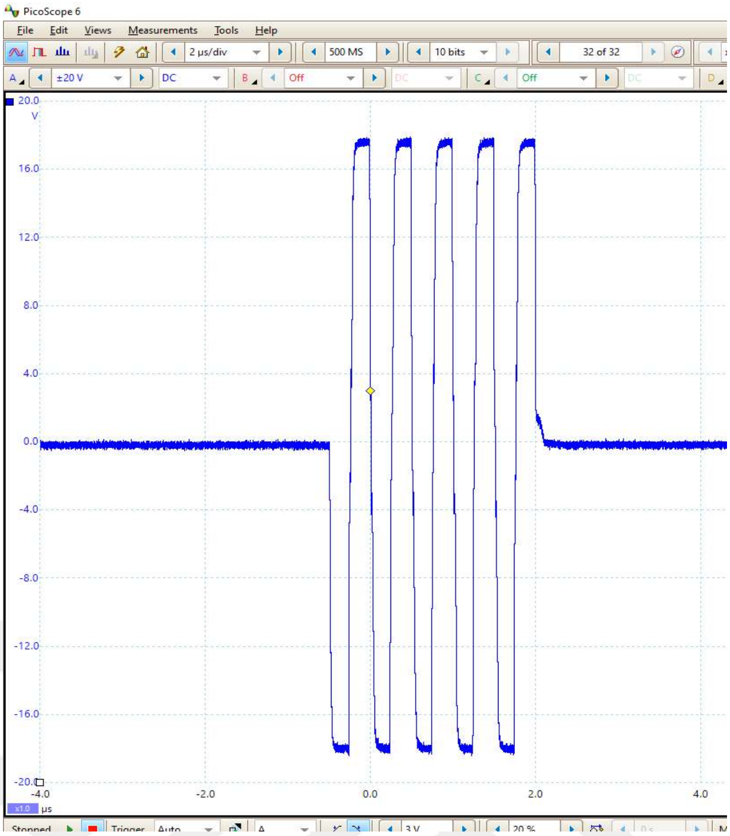


Figure A2. 3: 36V peak-to-peak of pulses

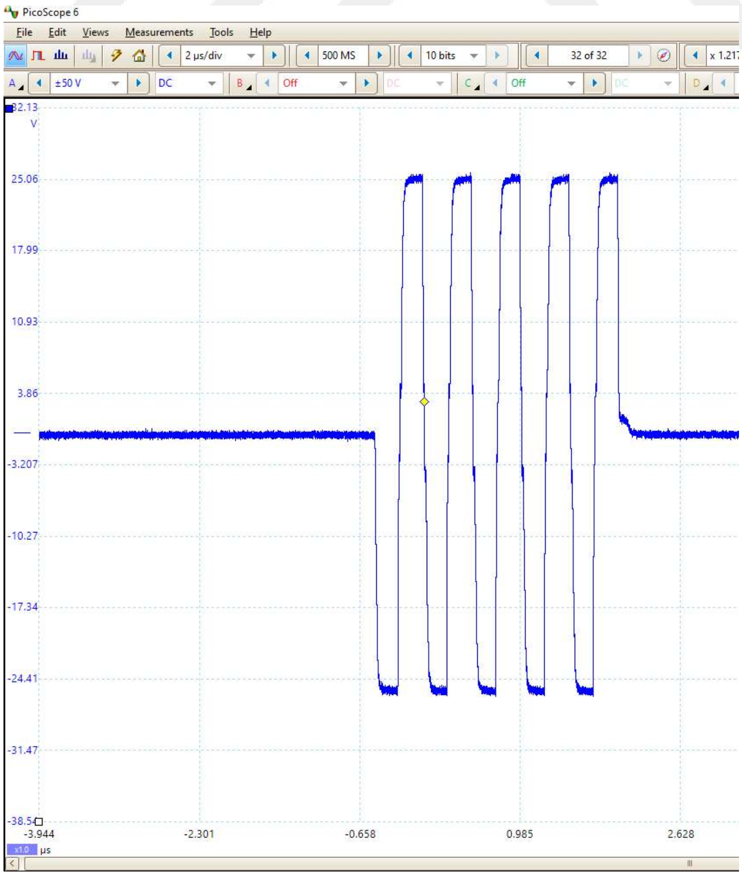


Figure A2. 4: 50V peak-to-peak of pulses

A3: Schematic and PCB layouts of the MAX14808 chip.

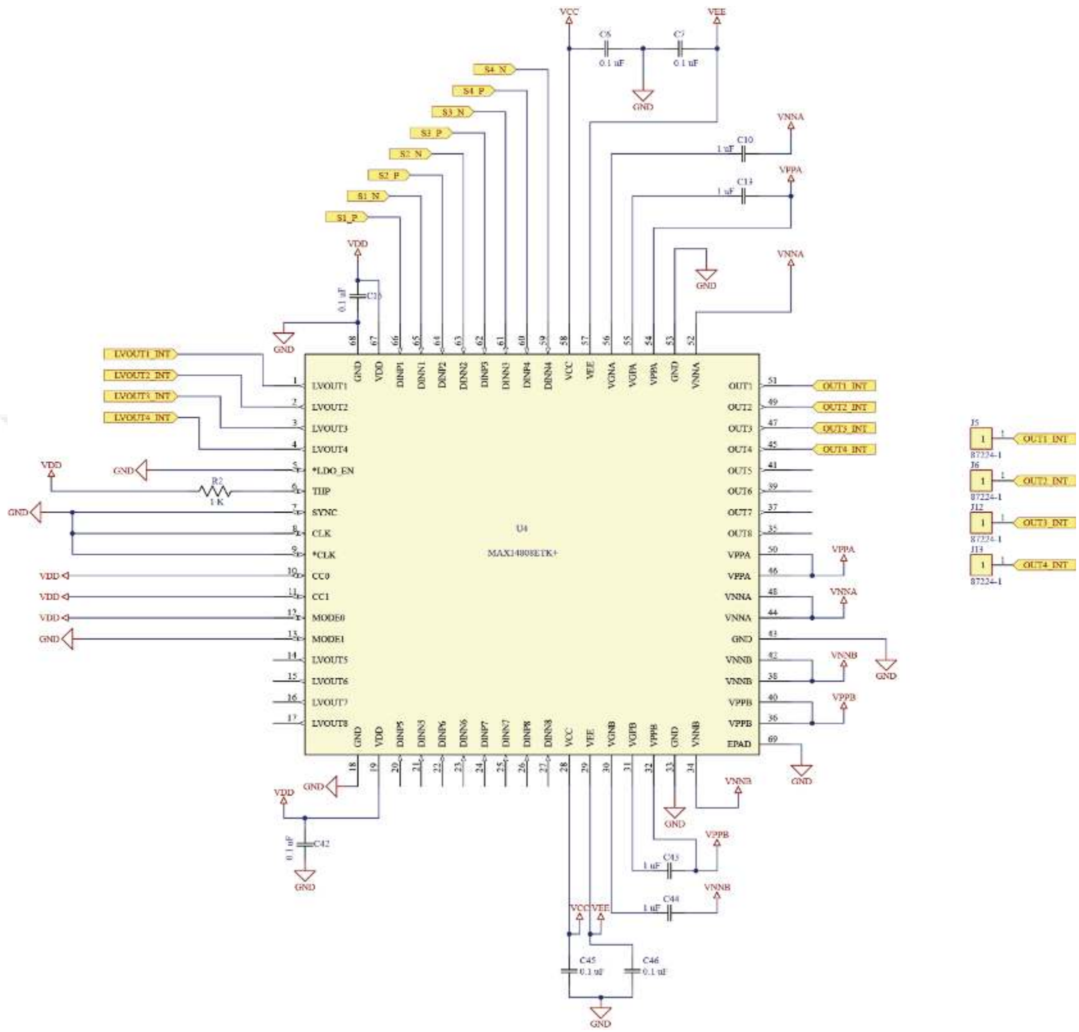


Figure A3. 1: MAX14808 high voltage pulser system layout

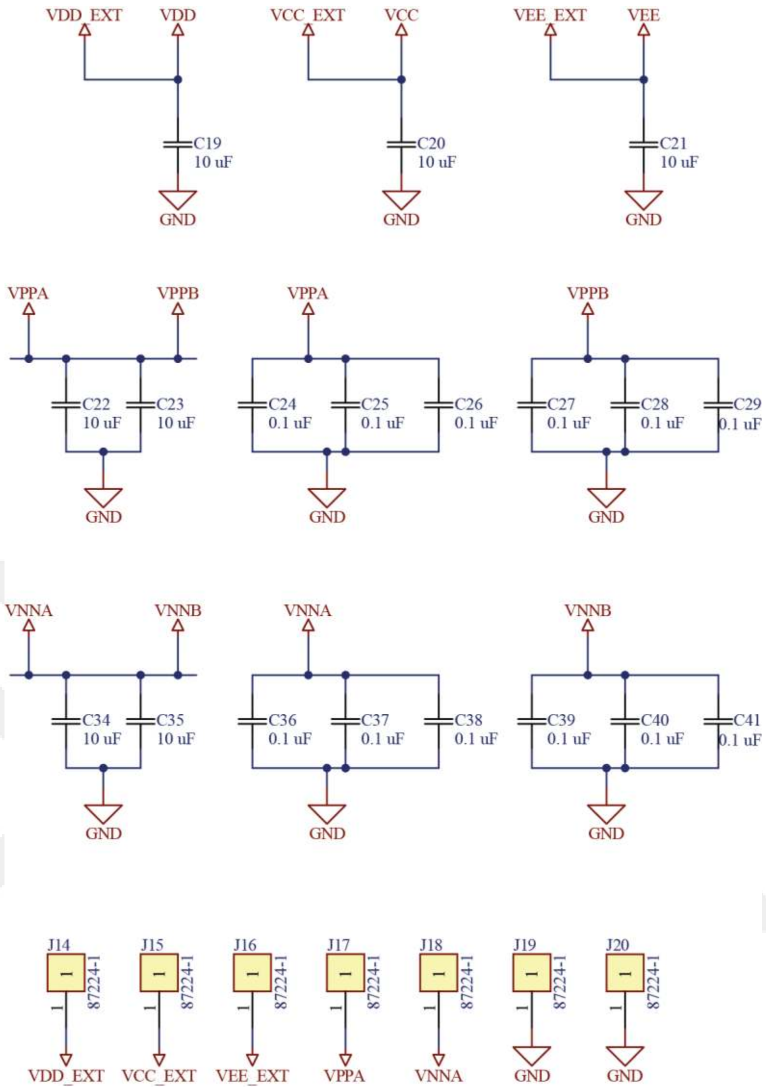


Figure A3. 2: MAX14808 high voltage pulser power layout

A4: Miniaturized PCB design layouts.

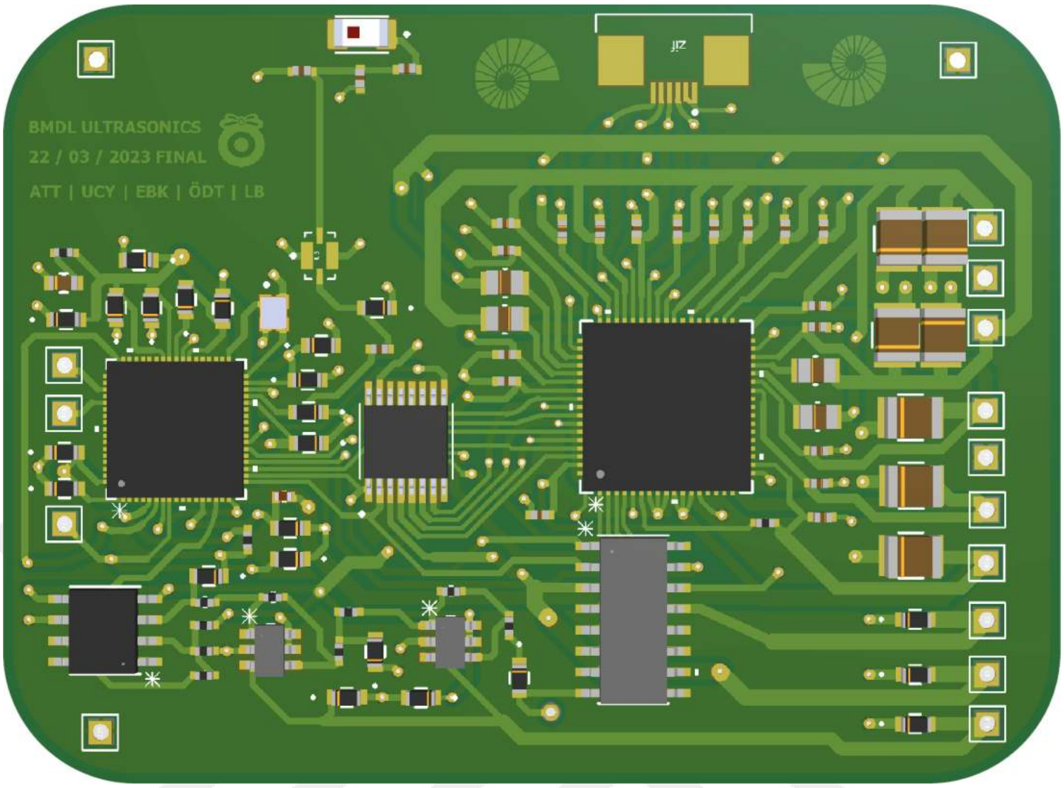


Figure A4. 1: Top layer layout of the miniaturized PCB (3d)

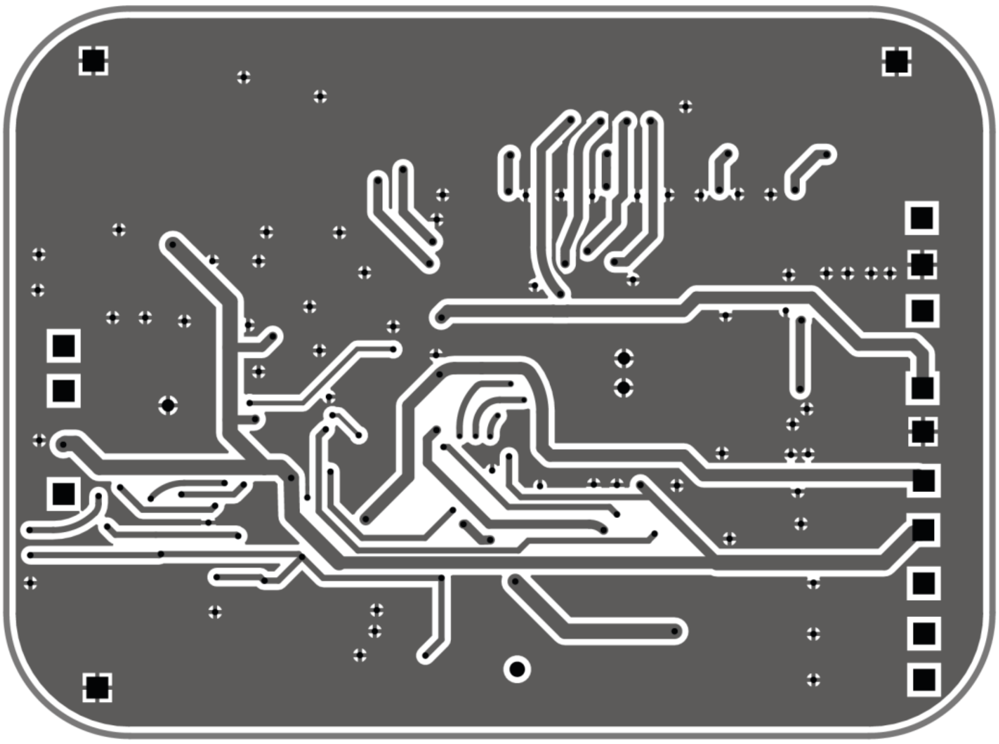


Figure A4. 2: Bottom layer layout of the miniaturized PCB (2d)

A5: List of the components used in the miniaturized PCB.

Comment	Description	Designator	Footprint	LibRef	Quantity
2450AT18A100E	2.4GHz Chip RF Antenna 2.4GHz ~ 2.5GHz 0.5dBi Solder Surface Mount	AE1	ANTC3216X140N	2450AT18A100E	1
0.1 uF	0603, No Description Available	C1, C2, C3, C4, C5, C6, C7, C9, C15, C16, C17, C24, C25, C26, C27, C28, C29, C32, C36, C37, C38, C39, C40, C41, C42, C45, C46, C47, C50, C51	CAP_1608_TDK	0.1 uF, 1 uF	30
100 pF	0603	C8	CAP_1608_TDK	100 pF	1
1 uF	No Description Available	C10, C13, C43, C44	CAP_1608_TDK	1 uF	4
10 pF	No Description Available	C11, C12	CAP_1608_TDK	10 pF	2
4.7 uF	0603	C14, C18	CAP_1608_TDK	0.3 pF	2
10 uF	0805 50V GRM21BR61H106KE43K	C19, C20, C21, C22, C23, C34, C35	CAP_GRM21BR61H106KE43K_MUR	10 uF_1	7
0.8 pF	0603	C30	CAP_1608_TDK	0.8 pF	1
0.3 pF	0603	C31	CAP_1608_TDK	0.3 pF	1
1pF	0402	C33	IND_LQG15HNR10J02_MUR	1pF	1
82 pF	0603	C48, C49	CAP_1608_TDK	0.1 uF	2
BLM15BX601SN1D	600 Ohms @ 100MHz 1 Signal Line Ferrite Bead 0402 _1005 Metric_ 350mA 460mOhm	FL1	BEADC1005X55N	BLM15BX601SN1D	1
87224-1	AMPMODU 1 Position 2.54 mm Single Row Through Hole Straight Header	J2, J3, J4, J5, J6, J7, J8, J9, J10, J11, J12, J13, J14, J15, J16, J17, J18, J19, J20, J21, J22, J23, J24	HDRV1W68P0_1X1_213X213X1036	87224-1	23
10 uH_1	0603 for stm32wb board	L1	CAP_1608_TDK	10 uH_1	1
2.7 nH	0402 for stm32wb board	L2, L4	IND_LQG15HNR10J02_MUR	2.7 nH	2
3.9 nH	0402 for stm32wb board	L3	IND_LQG15HNR10J02_MUR	3.9 nH	1
150060VS55040	No Description Available	LED1	LED_150060B_WRE	150060VS55040	1
10 K	No Description Available	R1	RC0402N_PAN	12 K	1
1 K	No Description Available	R2, R7, R9	RC0402N_PAN	12 K	3
12 K	No Description Available	R3	RC0402N_PAN	12 K	1
20 K	No Description Available	R4, R6	RC0402N_PAN	12 K	2
60 K	No Description Available	R5	RC0402N_PAN	12 K	1
330 R	No Description Available	R8, R10	RC0402N_PAN	12 K	2
8 K	No Description Available	R11, R12	RC0402N_PAN	12 K	2
DG409CUE		U1	SOP65P640X110-16N	DG409CUE	1
STM32WB55RGV6	No Description Available	U2	VFQFN68-8x8mm	STM32WB55RGV6	1
LFB212G45SG8A192	No Description Available	U3	FIL_LFB212G45SG8C341	LFB212G45SG8A192	1
MAX14808ETK+	No Description Available	U4	21-0142E_T6800-2_MXM	MAX14808ETK+	1
DG408DY+	No Description Available	U5	21-0041_S16+1_MXM	DG408DY+	1
OPA357AIDBVT	No Description Available	U6, U7	DBV6	OPA357AIDBVT	2
MAX941ESA+	No Description Available	U8	21-0041B_8_MXM	MAX941ESA+	1
ECS-320-8-37CKM	32MHz Â±10ppm Crystal 8pF 60 Ohms 4-SMD, No Lead	Y1	XTAL_ECS-320-8-37CKM	ECS-320-8-37CKM	1
CM9V-T1A-32.768KHZ	LOW FREQUENCY CRYSTAL	Y2	XTAL_CM9V-T1A-32.768KHZ-12.5PF-20PPM-TA-QC	CM9V-T1A-32.768KHZ-12.5PF-20PPM-TA-QC	1

Figure A5. 1: The component list belonging to the miniaturized PCB.

A6: STM32WB55RGV6 microcontroller system layout

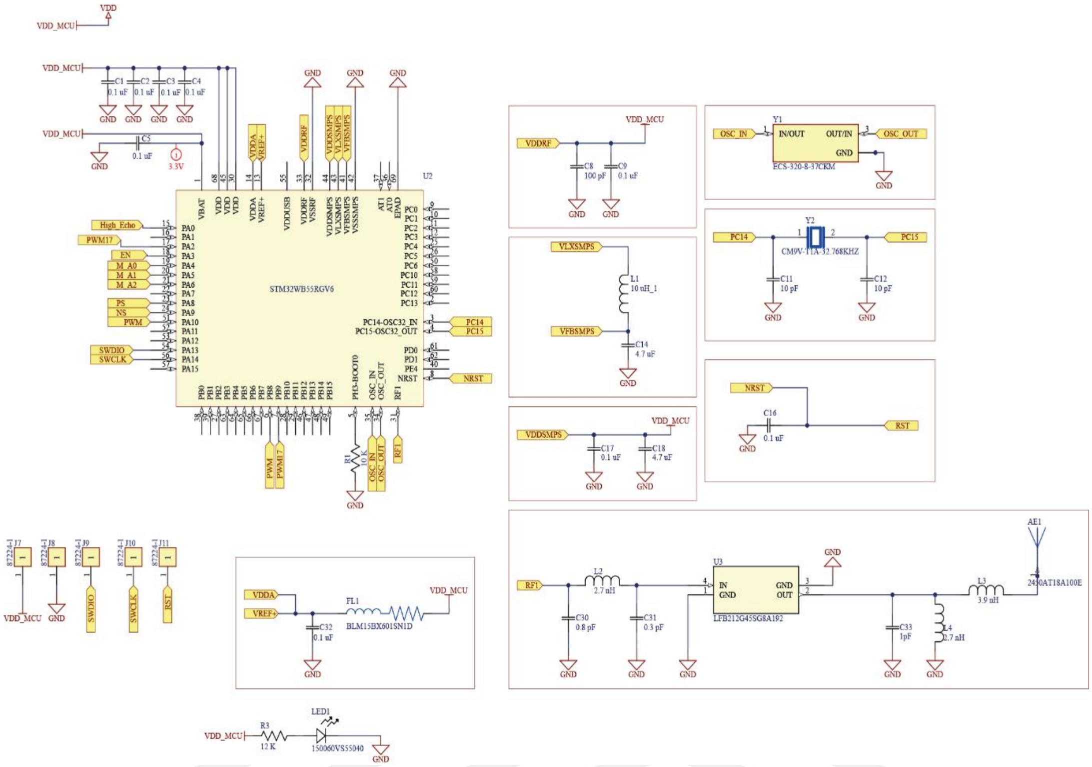


Figure A6. 1: The schematic of the microcontroller design

A7: In vivo test results / ultrasound images

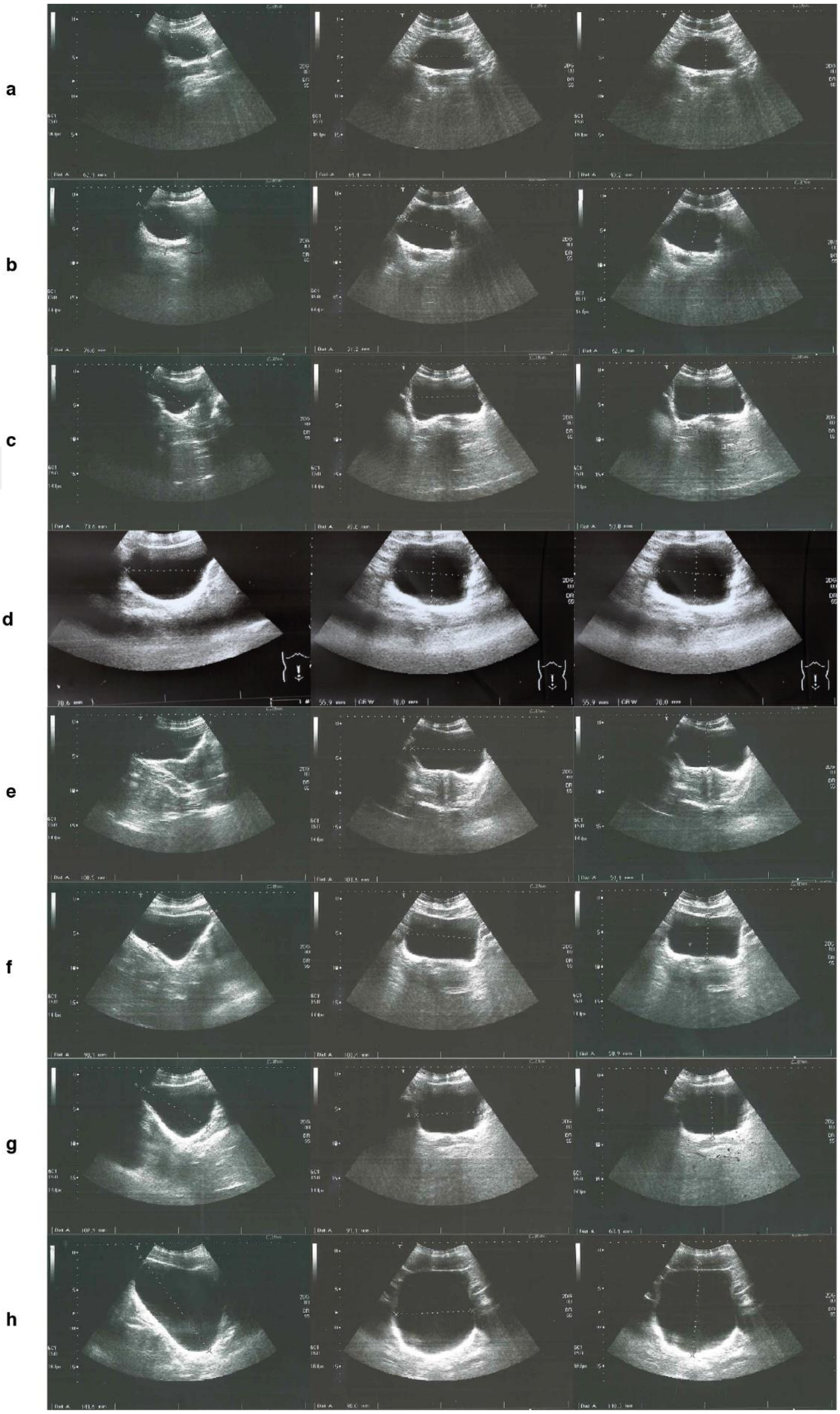


Figure A7. 1: Ultrasound images of the volunteers' bladder taken by a doctor