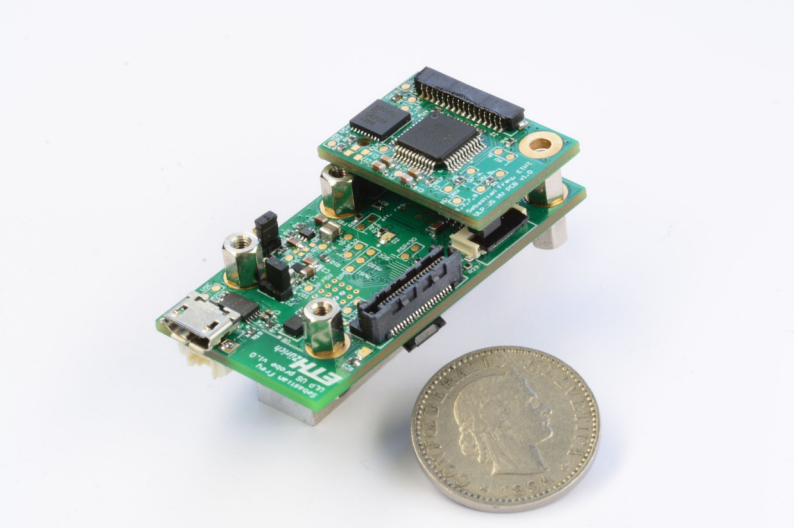
Integrated Systems Laboratory,
Department of Information Technology and
Electrical Engineering

WULPUS User Manual



Authors :

Sergei Vostrikov	vsergei@iis.ee.ethz.ch
Sebastian Frey	sefrey@iis.ee.ethz.ch
Cedric Hirschi	cehirschi@student.ethz.ch
Josquin Tille	jtille@student.ethz.ch
William Bruderer	wbruderer@student.ethz.ch
Luca Benini	lbenini@iis.ee.ethz.ch
Andrea Cossettini	cossettini.andrea@ethz.ch

December 20, 2024

Contents

Chapter 1

System Overview

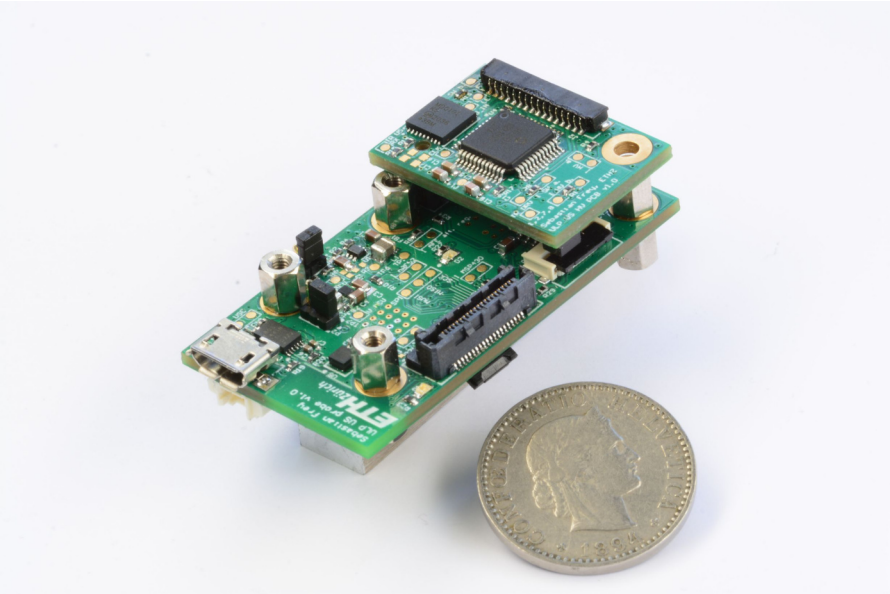


Figure 1.1: The WULPUS system compared to a coin.

1.1 What is the WULPUS probe?

The Wearable Ultra Low-Power Ultrasound (WULPUS) probe is the first open-source ultrasound platform developed for wearable applications. WULPUS was first presented at IUS 2022 [1], demonstrating a real-time acoustical monitoring of common carotid artery and gastrocnemius muscle. Further developments presented a WULPUS-based armband

for hand gesture recognition [2, 3], an improved armband for hand movement regression [4] and a chest patch for complete cardiorespiratory monitoring [5] (i.e., simultaneous respiration and heart rate extraction). The prime objective of the WULPUS platform is to provide a compact, energy-efficient, and wireless solution for ultrasound on-body sensing.

Unlike other sensing technologies for deep tissue inspection, ultrasound is non-ionizing, making it safe for regular use. It also offers high temporal resolution and is cost-effective. The WULPUS probe amplifies these benefits by being wearable and energy-efficient, aiming at multi-day continuous operation without the need for frequent recharging. Furthermore, WULPUS can be flexibly configured at receive/transmit, and also provides researchers access to raw digitized US data, facilitating algorithm development.

The device is structured with modularity in mind, encompassing an Acquisition PCB (Printed Circuit Board) and a High-Voltage PCB (High Voltage Multiplexer PCB). The Acquisition PCB is the heart of the system, controlling ultrasound measurements and data flow, while also managing the Bluetooth Low Energy (BLE) connection for wireless data transmission. The High-Voltage PCB, on the other hand, is responsible for driving ultrasound transducer with high-voltage pulses, multiplexing transmit/receive channels and providing simple analog filtering for the receive path.

WULPUS is not just a hardware device, but a comprehensive solution. It comes with a Python library and GUI for seamless data logging, processing and visualization on a computer. The data acquired can be utilized for automatic analyses, paving the way for developing custom algorithms.

This document aims to assist new users in getting started with the WULPUS system with minimal efforts. Chapter 1 offers an overview of the system design. Additionally, chapter 2 provides guidance on creating a custom WULPUS probe from scratch, covering the steps from PCB production to firmware flashing and silicone package fabrication. The following chapter 3 walks through Graphical User Interface (GUI) of the WULPUS platform. Meanwhile, in chapter 5 users are led through simple water bath experiment using the system. In addition to that, chapter 4 provides information for flexible configuration of the WULPUS system to tailor the platform for specific application scenario. Finally, troubleshooting common issues is covered in chapter 6, and the critical bugs are described in chapter 7.

1.2 Specifications

The full technical specifications of the WULPUS system are available in the Table below.

Table 1.1: Overview of the specifications of individual parts of the US probe system

Feature	Details
Number of channels	**8, time-multiplexed**
Acquisition	**8 Msps, 12 bit** Analog-to-Digital converter
Amplification	**30.8dB** Programmable-Gain Amplifier + **10dB** fixed amplification stage
Communication	**Bluetooth Low Energy** (BLE) link with 320 kbps throughput
Frame rate	**50 acquisitions per second** (APS) (400 samples per acquisition, raw data streaming mode)
Dimensions	**46×25mm** footprint
Power consumption	**22mW** (raw data streaming mode, 50 APS)

1.3 System Components

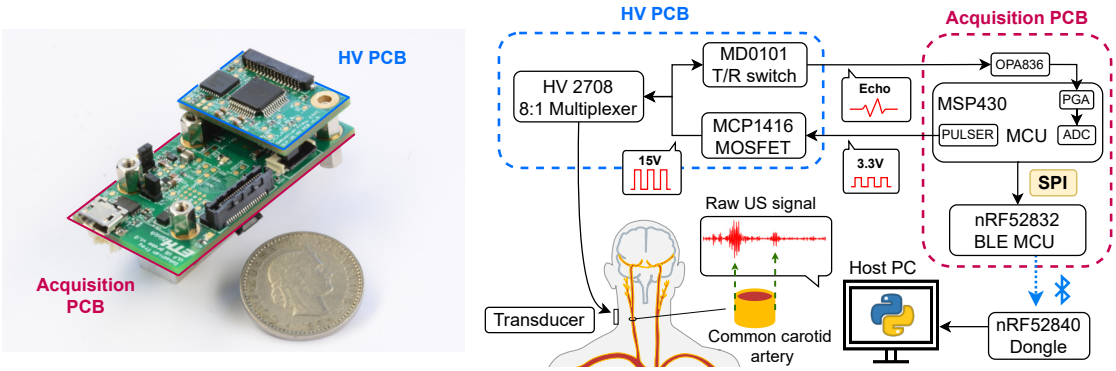


Figure 1.2: System overview of the WULPUS system.

The US probe was developed with the following guiding principles: it should be possible to adapt and further improve the device (improving the signal generation/acquisition, and including on-board intelligence) without the need for a complete redesign. This led to a modular approach that is described in the following. The different parts of WULPUS are shown in 1.2 and described here shortly:

- **Acquisition PCB:** The Acquisition PCB controls the US measurements and handles the data flow (see 1.3.1 for details).
- **High-Voltage PCB:** The High-Voltage PCB can be plugged on top of the Acquisition PCB and handles low to high excitation level translation, transmit/receive signals multiplexing and simple analog filtering of the receive signal (see 1.3.2 for details).

- **Transducer:** The transducer converts the US pulses from the electrical to the acoustic domain and vice versa.

  > **Note:** WULPUS can operated with different piezoelectric transducers in wide frequency range (≈ 100 kHz - 4 MHz). A user should connect a custom transducer to the connector on the HV PCB.

- **Receiver Dongle:** The nRF52840 Dongle is a commercially available USB dongle that is used to receive the US data from the probe (see 1.3.3 for details).
- **Battery:** The lithium polymer battery powers the WULPUS probe. For testing, it's also possible to use a lab supply or a micro USB cable.
- **Python GUI:** The python GUI runs on a windows computer and manages connection to the WULPUS probe, data collection, visualization and logging.

1.3.1 Acquisition PCB

The Acquisition PCB is the central unit of the US probe system. Its primary functions are:

US Measurements: The MSP430FR5043 SoC is a central component here. It's not only power-efficient but also has the capability to handle US measurements due to its advanced features. The SoC can generate pulses, manage signal amplification, and sample the incoming data with low noise.

Data Transmission: The nRF52832 SoC ensures the seamless transmission of the acquired US frames to the receiver dongle via BLE. Known for its high-speed data transfer capabilities, this chip ensures the data is relayed quickly and power efficiently.

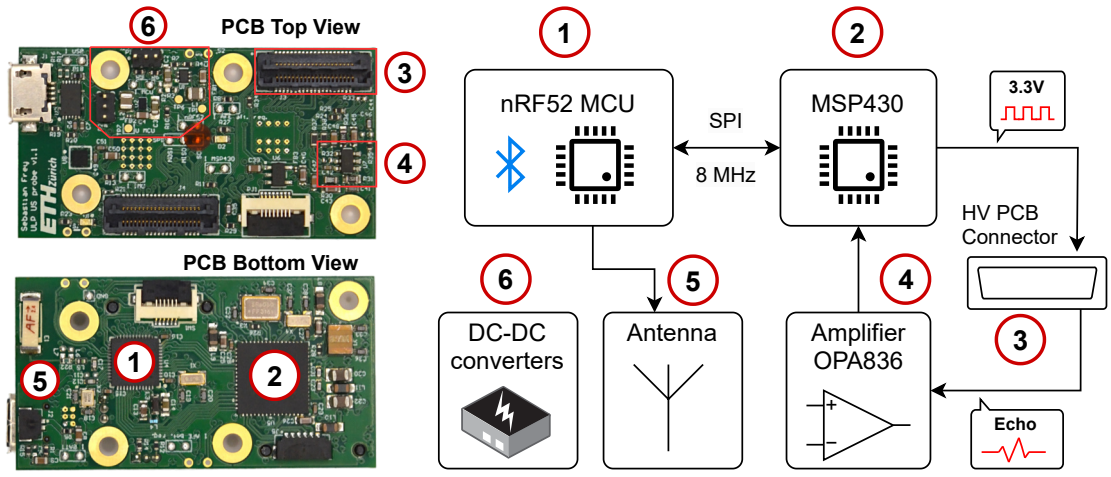


Figure 1.3: The acquisition PCB

Figure 1.3 offers a clear depiction of how the various components of the Acquisition PCB interact in order to fulfill these functions. At the start, the MSP430 generates low voltage pulses. These are then sent to the High-Voltage PCB, which communicates with the transducer. Once the reflections are captured, they are sent back to the Acquisition PCB, amplified by the OPA836 opamp, and processed further by the MSP430. Upon completion, the nRF52 sends the data wirelessly to the receiver dongle using the BLE connection.

1.3.2 High-Voltage PCB

Now, how does the WULPUS probe interact with the transducer? That's where the High-Voltage PCB comes in, which also plays a pivotal role in the WULPUS probe. Its main functions are:

Voltage Translation: The core purpose of the High-Voltage PCB is to amplify the low voltage excitation pulses that come from the Acquisition PCB. This amplification is vital for adequately exciting the US transducers. The MOSFET driver MCP1416 switches on/off based on these low voltage pulses, producing unipolar HV pulses with an amplitude of +15 V.

Transducer Channel Switching: The analog HV switch HV2708, acting as a multiplexer, provides flexibility in controlling the transducer channels. It can selectively enable or disable individual transducer channels or groups of them for either transmission or reception of the US pulses.

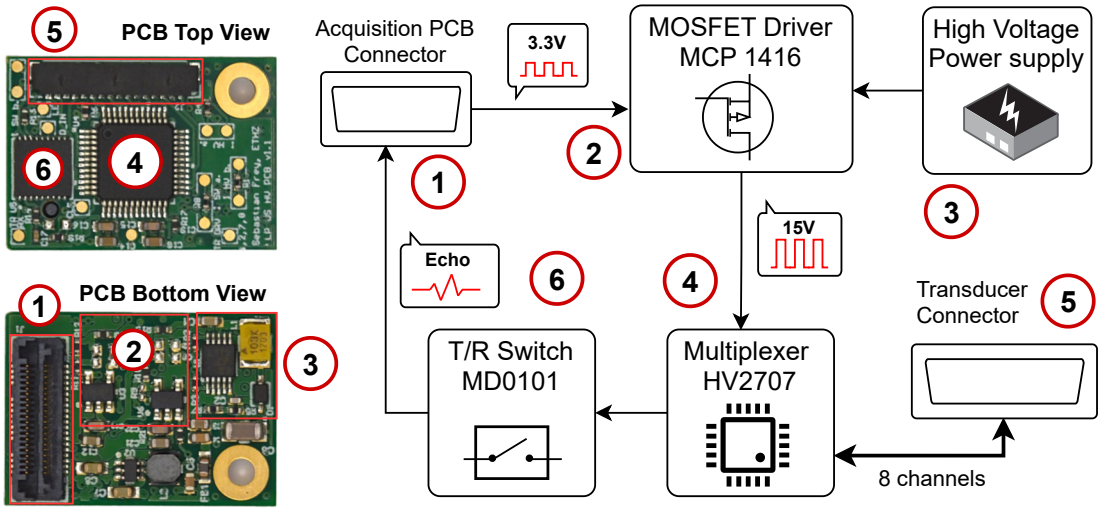


Figure 1.4: The High Voltage PCB

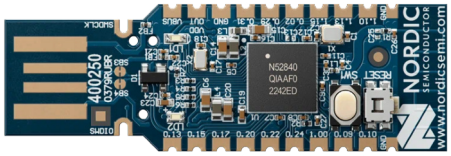


Figure 1.5: nRF52840 Dongle from Nordic Semiconductor. Image adapted from [6].

Figure 1.4 offers a comprehensive view of the interactions between the various components of the High-Voltage PCB. The low voltage pulses originating from the Acquisition PCB are amplified to generate the required high voltage. This boosted voltage then interacts with the transducers before the reflected signal makes its way back to the connector of the Acquisition PCB.

1.3.3 USB Dongle and Host PC

After the probe is done acquiring and sending the data, is where the dongle comes in. The nRF52840 USB Dongle is a crucial intermediate component in the WULPUS probe design. Its role is to receive, process, and forward the data relayed by the Acquisition PCB to the Host PC for further interpretation and display.

The **USB Dongle** is responsible for multiple tasks:

Wireless Reception: At its core, the Dongle contains an nRF52 SoC, which is responsible for wirelessly receiving the data transmitted from the Acquisition PCB. This SoC is recognized for its high-speed data transfer capabilities, ensuring data integrity and timely reception.

Interface with Host PC: Once the data is received by the Dongle, it interfaces with the Host PC, often through a virtual COM port. This enables seamless communication and data transfer between the Dongle and the Host PC.

Finally, on the **Host PC**, a WULPUS GUI provides the following functionality to a user:

Data Reception and Processing: The Host PC runs a Python code which interprets and processes the incoming data. This includes data readout from the COM port, extracting information from raw packages and interpreting it.

Data Display and Storage: The Host PC serves as the primary display and storage medium. Captured ultrasound acquisitions can be visualizaed in real-time, offering immediate feedback and insights. Simple data processing such as band-pass filtering and envelope extraction is available to a user during visualization. The data can be also saved to a file for later retrieval, further analysis, or sharing.

Measurement Configuration: The Python GUI also facilitates user interaction with the system, enabling adjustments in settings, preferences, and other functionalities that can influence the measurement process and display.

In essence, the Dongle and the Host PC collectively serve as the endpoint in the WULPUS system. The High-Voltage PCB and Acquisition PCB handle the initial measurements, and these final components ensure the data is appropriately processed, displayed, and stored.

Chapter 2

How to build your own WULPUS probe?

2.1 PCB manufacturing and assembly

Hardware files (Altium Nexus CAD projects, .pdf schematics, .xlsx BOMs) can be found in the *hw* directory in the GitHub repository.

The manufacturing of Printed Circuit Boards (PCBs) is a flexible process that can be tailored to specific needs and expertise levels. Generally, there are two main approaches to consider:

1. **Self-Assembly:** Order the PCBs and personally handle the assembly of all components, including microcontrollers, passive elements, and other necessary parts.
2. **Full-Service Manufacturing:** Opt for a comprehensive solution where the manufacturer not only produces the PCBs but also takes care of assembling all the components onto the boards.

The choice between these options largely depends on your experience, resources, and project requirements.

2.1.1 Self-Assembly

When choosing to assemble the PCBs on your own, the PCB manufacturer of your choice only needs to fabricate the PCBs itself. For this, you can provide the manufacturer with the following files:

1. **Gerber files:** These files play a crucial role as they are the standard format used to describe the layout of the PCB. They contain information about the copper layers, solder mask, silkscreen, and any additional layers that are part of the PCB design. Essentially, Gerber files communicate with the manufacturing equipment what to produce, showing where to lay down copper, where to apply protective coatings, where to place holes, and so on. They are used to guide the photolithography machines that create the PCB layers. Each layer of the PCB (for example, top copper, bottom copper, solder mask layers, etc.) will have its own Gerber file. The files can be found in *hw/<pcb>/docs/fabrication_files/Gerber*.

2. **NC Drill files:** These files are used to instruct the drilling machines on where to drill holes in the PCB. These files provide the coordinates and sizes of the holes that need to be drilled, including vias (which connect different layers of the PCB), mounting holes, and any other through-holes required for components. The NC drill files are located in *hw/<pcb>/docs/fabrication_files/NC Drill*.

3. **The stackup** of the PCB refers to the arrangement of copper layers and insulating materials (dielectrics) in a PCB to form the complete board structure. It's an essential part of PCB design and manufacturing, as the stackup affects the board's electrical performance, including its impedance, signal integrity, and thermal management. The suggested stackup is located in *hw/<pcb>/docs/fabrication_files/<pcb>_stackup.xls*

Self-assembly is only recommended for those who have previous experience with soldering and have good soldering equipment such as a variable temperature soldering iron, soldering dispenser, solder wick, flux, tweezers, a magnifying glass or microscope, and ideally a hot air station or a reflow oven.

2.1.2 Full-Service Manufacturing

A convenient way to get your own WULPUS is to opt for full-service manufacturing. In this case, the manufacturer is not only producing the PCBs, but he also takes care of ordering the electronic components and their assembly. In addition to the files mentioned above, you should provide the manufacturer with the following ones:

1. **Pick-and-place files:** They are needed during the assembly process, particularly during the automated placement of components onto the board. These files contain detailed information about the components to be placed on the PCB, including their exact locations, orientations, and reference designators. This information guides the automated pick-and-place machines that physically place components on the board during assembly. The pick-and-place files can be found in *hw/<pcb>/docs/fabrication_files/Pick Place*

2. **Assembly drawing:** This is a detailed diagram that provides comprehensive information on how various components are to be assembled on the PCB. This drawing

serves as a visual guide for the assembly process, highlighting the placement of components, their orientations, and any special assembly instructions. The assembly drawing is located in *hw/<pcb>/docs/<pcb>_assembly_drawing.pdf*.

3. **The Bill of Materials (BOM):** A list that contains all components that should be assembled on the PCB. The BOM specifies not just what components are needed but also their designator and quantities. The BOM is located in *hw/<pcb>/docs/<pcb>_bom.xls*.

2.1.3 Cost estimation

The overall cost of a WULPUS probe is highly dependent on how many pieces per production batch are produced. In the case of a batch of around 10 to 20 probes, the expected cost per WULPUS probe lies in the range of **150 to 200 USD**, depending on the chosen PCB manufacturer, the component costs and if the assembly is handled by the manufacturer or not.

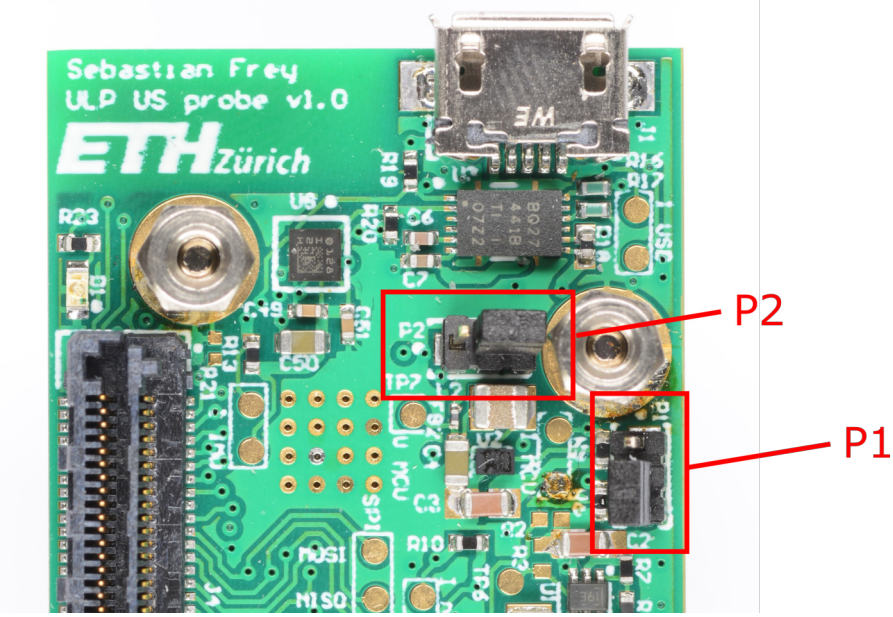


Figure 2.1: Default positions of the power jumpers

2.2 Powering up the US probe

There are four different powering configurations for WULPUS. You can either power it from USB or from the battery port. The system voltage $V_{SYS,MCU}$ can also be configured to be either 2.5V or 3.3V.

To select these configurations, you have two jumpers **P1** and **P2**, which can both take two positions (1-2 and 2-3, thus pins 1 and 2 bridged or pins 2 and 3 bridged, respectively). Each jumper has its pin 1 marked by a dot.

The default configuration is powered from USB, 3.3V, as shown in figure 2.1. The jumper functions are again described in table 2.1.

	1-2	**2-3 (default)**
P1	2.5V	3.3V
P2	Battery	USB

Table 2.1: Power jumpers and their functions

2.3 Programming the MSP430

In order to program the MSP430, you need the following components (see figure 2.2):

1. MSP FET programmer
2. JTAG - Mr. Wolf adapter PCB
3. Molex cable 8-pin
4. 8 Jumper cables (female to female)
5. USB cable (type B to mini USB)

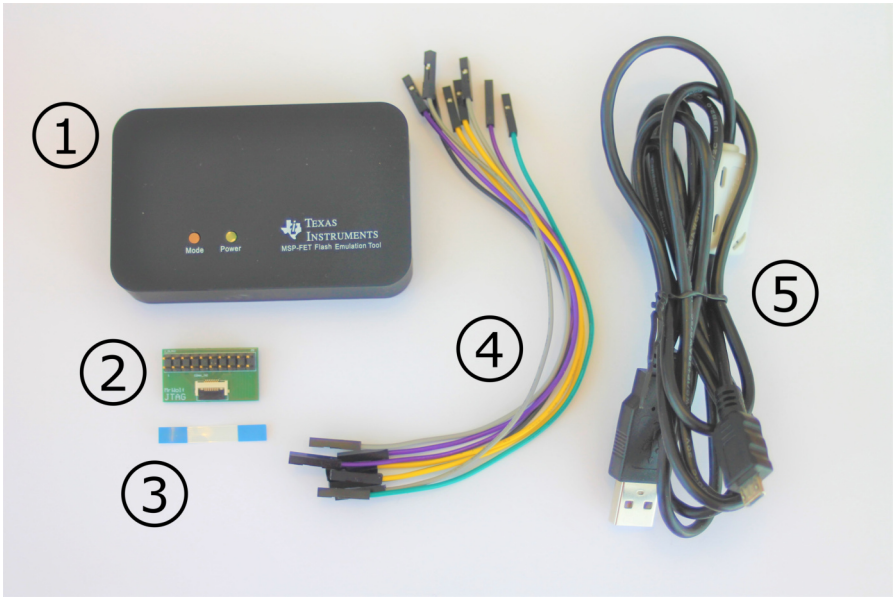


Figure 2.2: Components programming MSP430

2.3.1 Setting up the toolchain

TI Code Composer Studio (CCS) (V11.0.0.00012)

1. Go to `https://www.ti.com/tool/CCSTUDIO#downloads` and download the single file installer
2. Extract the downloaded file and run `ccs_setup_11.0.0.00012`
3. Choose `Custom installation`
4. In `Select Components`, select `MSP430 ultra-low power MCUs` and finish the installation

MSP430Ware (V3.80.14.01)

1. Open Code Composer Studio
2. Go to `View -> Resource Explorer`
3. Select `Software -> MSP430Ware (3.80.14.01)` and install it

Add driverlib to CCS (V2.91.13.01)

1. Close CCS
2. Go to `https://www.ti.com/tool/MSPDRIVERLIB#downloads` and download MSP-DRIVERLIB
3. Extract the downloaded file and copy the extracted folder into your CCS instalaltion (e.g. `C:\ti\msp430_driverlib_2_91_13_01`)
4. Open CCS
5. Go to `Windows -> Preferences -> Code Composer Studio -> Products`
6. Click on `Add...` to add the driverlib path to the product discovery path (e.g. `C:\ti\msp430_driverlib_2_91_13_01\driverlib`
7. Click on `Rediscover...`
8. Select the option for `driverlib`, press `Install` and agree to restart CCS

2.3.2 Flashing the device

Open the project in CCS

1. Open CCS and set your workspace
2. Select `File -> Import -> CCS Project -> Next`
3. Select `Search Directory`, set it to `wulpus/fw/msp430/wulpus_msp430_firmware`
4. Select discovered project `wulpus_msp430_firmware` and click `Finish`

Build the project

1. Select `Project -> Build Project`
2. Build the project

Flash the project

1. Connect the MSP FET programmer to the US probe using the 8-pin Molex connector according to Figure 2.3
2. Power the US probe as described in Section 2.2

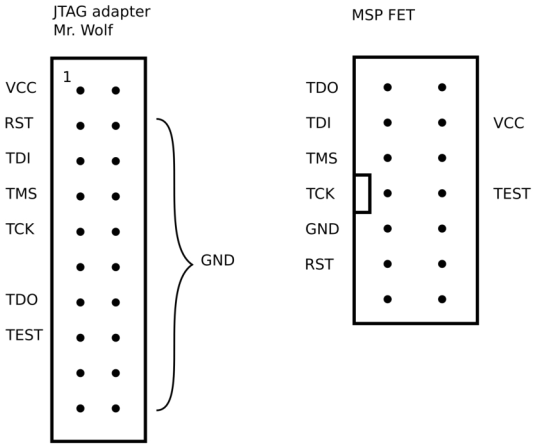


Figure 2.3: Connection between the MSP JTAG adapter and the MSP-FET

3. Select `Run -> Load -> Select Program to Load`
4. Navigate to the `Debug` subfolder of the CCS project and select the file with the extension `.out`
5. Flash the code via the Flash icon in CCS or via `Run -> Load -> ...`

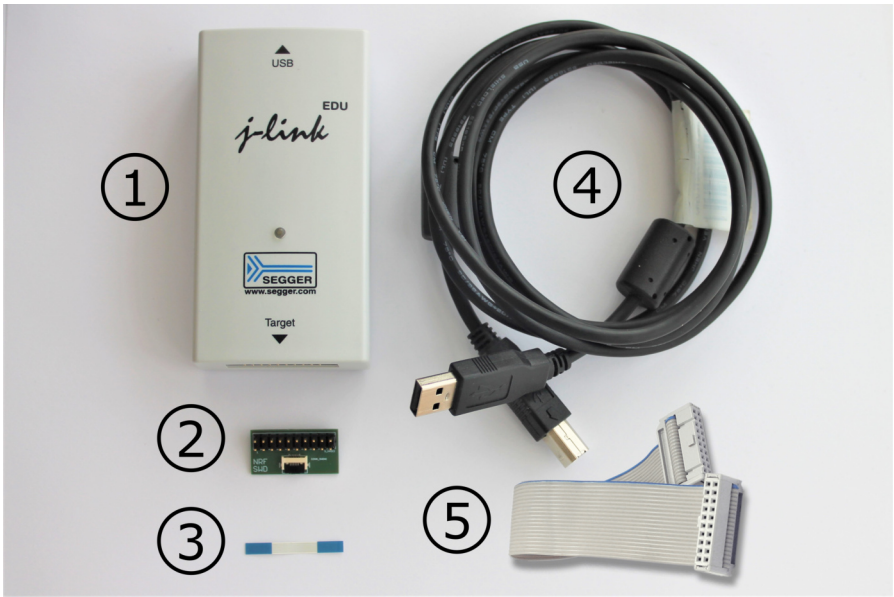


Figure 2.4: Components programming nRF52

2.4 Programming the nRF52 on the Acquisition PCB

The nRF52 MCU can be programmed with a J-Link Debugger. The following components are needed to do so (see figure 2.4):

1. J-Link Debugger
2. Adapter PCB NRF SWD
3. Molex cable 6-pin
4. USB cable (type A to type B)
5. JTAG 2x7 Ribbon Cable

2.4.1 Setting up the toolchain

For a more detailed Getting Started Guide with other IDE options, see `https://infocenter.nordicsemi.com/topic/ug_gsg_ses/UG/gsg/intro.html` and select `Getting started with nRF5 SDK and SES (nRF51 & nRF52 Series) -> Setting up your toolchain`. Be aware that the WULPUS firmware project is only configured in SES. If it were to be used in other IDEs, some configurations would have to be made manually.

Note: The nRF projects are compatible with the nRF SDK SES version (5.42a or 5.62). Under the new version (7.32) compilation of the Dongle firmware fails.

Segger Embedded Studio (SES) for ARM (v5.62)
Go to `https://www.segger.com/downloads/embedded-studio`, download and install the `v5.62` installer for your OS.

(Optional) Obtain a free license for SES
You can leave out this step if you only use SES for academic purposes and are ok with accepting a prompt every time you open SES.

1. Go to `https://license.segger.com/Nordic.cgi`
2. Fill in the requested information, the license will then be sent to your mail
3. Open Segger Embedded Studio
4. Click `Tools -> License Manager`
5. Select `Activate SEGGER Embedded Studio`
6. Paste in your license and click on `Install license`

Note: After doing this, it may take a few hours until the license is activated.

Segger J-Link Software and Documentation Pack (v7.56)
Go to `https://www.segger.com/downloads/jlink`, download and install the `v7.56` installer for your OS.

Nordic nRF5 SDK (V17.1.0)

1. Go to `https://www.nordicsemi.com/Products/Development-software/nRF5-SDK`, download and unpack the `v17.1.0` package.
2. Place the unpacked folder close to the root level of your file system (e.g. `C:/nordic/nRF5_SDK_17.1.0_ddde560/` or `~/nordic/nRF5_SDK_17.1.0_ddde560/`). Try not to use any spaces in the file path or folder name.
3. Copy the contents of the folders `ble_peripheral` and `peripheral` of the firmware section of the respository into the counterparts of the same name in the `examples` folder of the SDK (e.g. `C:/nordic/nRF5_SDK_17.1.0_ddde560/examples/`)
4. Inside each copied project (e.g. `US_probe_xxx_firmware`), in the subdirectory `pca100xx/s1xx/ses/`, you will find a `.emProject` file. You can then open this file with Segger Embedded Studio to open the project.

2.4.2 Flashing the device

1. Locate the `.emProject` file in the SDK's subfolder `examples/ble_peripheral/US_probe_nRF52_firmware` as described in 2.4.1 and open it with Segger Embedded Studio.
2. In the file `us_defines.h`, change `DEVICE_NAME` to the name you want to use for the BLE connection. The default value is `WULPUS_PROBE_3`.
3. Connect the US probe with the J-Link debugger and the 6-Pin Molex connector
4. Flash the code via `Build -> Build and Run`
5. Power cycle the probe to start the execution of the code

2.5 Programming the USB Dongle

The dongle can be programmed directly through USB; no additional hardware is needed.

2.5.1 Setting up the toolchain

Nordic nRF Connect for Desktop (V3.7.1 or newer versions)

1. Go to `https://www.nordicsemi.com/Products/Development-tools/nRF-Connect-for-desktop`, download and install the `V3.7.1` (or newer) installer for your OS.
2. Open nRF Connect for Desktop
3. `Install` the `Programmer` application

SoftDevice S140 (V7.2.0)

Go to `https://www.nordicsemi.com/Products/Development-software/S140`, download and unpack (no specific location) the `V7.2.0` version.

2.5.2 Flashing the device

Build the Firmware

1. Locate the `.emProject` file in the SDK's subfolder `examples/peripheral/US_probe_dongle_firmware` folder as described in 2.4.1 and open it with Segger Embedded Studio.
2. In the file `us_ble.c`, change `DEVICE_NAME_TO_CONNECT` to the **same name** you chose to use in the probe's firmware (e.g. `WULPUS_PROBE_3`).
3. Build the code via `Build -> Build`
4. The resulting `US_probe_dongle_firmware.hex` is then located in the folder `pca10059/s140/ses/Output/Debug/Exe/`

Flash both the Firmware and SoftDevice

1. Open nRF Connect for Desktop and launch the Programmer application
2. Plug the dongle into an USB port
3. Press the reset button to put the Dongle in DFU mode. Note that the reset button is the sideways button right next to the better visible SW1 button on the Dongle
4. Select the dongle from the dropdown in the upper left corner of the Programmer app
5. Click `Add HEX file` to select the firmware hex file built above
6. Click `Add HEX file` to select the Softdevice hex file downloaded previously
7. Click `Write` to write the firmware to the dongle

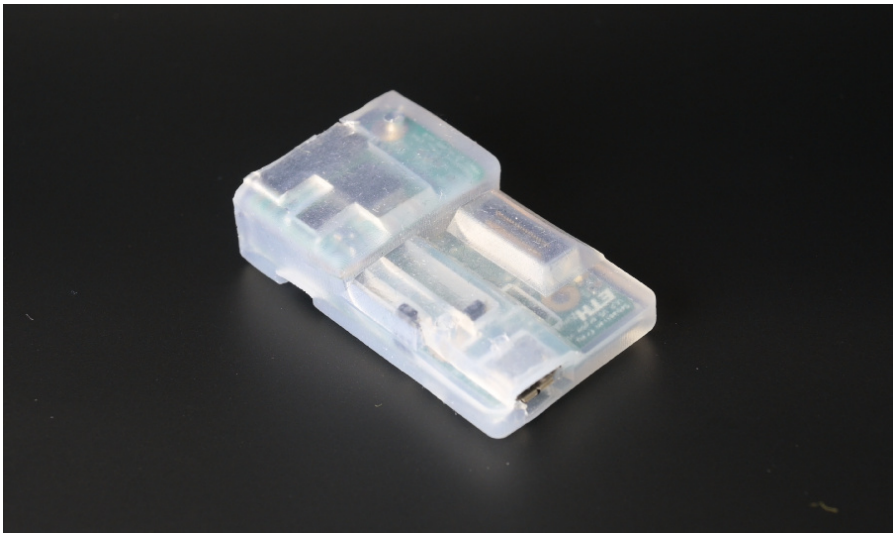


Figure 2.5: Silicone rubber package for WULPUS probe.

2.6 Silicone Rubber Package for WULPUS Probe

A silicone package (Fig. 2.5) can optionally be produced to protect the probe from mechanical shocks and electrostatic discharges. This section provides a list of materials required for creating this package, followed by detailed fabrication steps.

2.6.1 Bill of Materials

To build the silicone package, you will need the following items:

- Ecoflex^{TM}-0045 NEAR CLEAR, Silicone Rubber.
- The package's mold.
- 2x Dowel pins, 2.5 mm $\times$ 20 mm or longer.
- Disposable containers.
- Nitrile gloves.
- Vaseline.
- A vacuum pump.

Crafting the Mold

The files for the mold are available in the folder `hw/wulpus_silicone_package` of the repository. The mold can be fabricated using additive manufacturing techniques (3D printing such as FDM, SLA, DLS, etc.) or ordered from a 3D printing service.

DIY 3D Printing

If you can access an FDM 3D printer, you can use it to create the mold. It is recommended to use the files `*_fdm.stl`, which already incorporate tolerances for proper fitting. Standard materials such as PLA, ABS, or PETG are suitable. After printing and removing supports, the mold will be ready for use.

Ordering the Mold

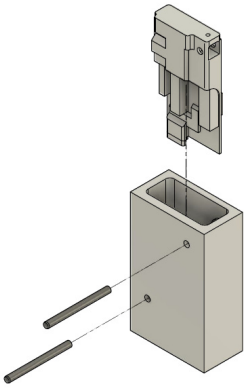
If a 3D printer is unavailable, you can order the mold from a 3D printing service provider, such as PCBWay. In this case, please use the SLA technology, upload the `.stl` files `*_resin.stl` and select "Resin, Standard white material (UTR 8360)" as the material. Ensure the option "Wall thickness risk" is allowed for the inner mold part.

2.6.2 Production Steps

It is highly recommended to wear nitrile gloves throughout these steps, especially when handling the Ecoflex^{TM} silicone rubber.

I. Preparing the Mold

1. Coat the walls of the hollow part of the mold with Vaseline.
2. Insert the inner mold part, ensuring the holes on both parts align.
3. Insert the dowel pins into the holes. Verify proper alignment by ensuring the inner and outer parts make contact at the designated points (see Fig. 2.6).



(a) Mold assembly.

(b) Assembled mold.

Figure 2.6: 3D printed mold for the WULPUS silicone package.

II. Mixing the Ecoflex™

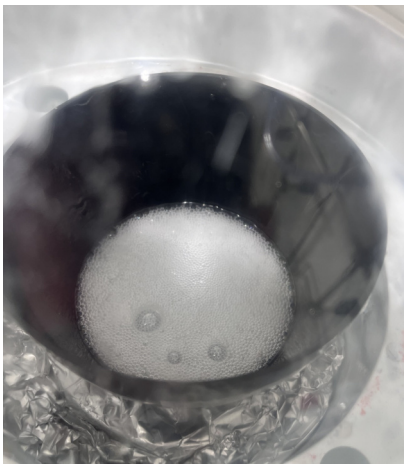
1. Thoroughly pre-mix Parts A and B before starting.
2. Dispense 10 ml of each part (A and B) into the mixing container (1:1 ratio by volume or weight).
3. Mix thoroughly, ensuring you scrape the container's sides and bottom multiple times.
4. Vacuum degas the mixture to remove air bubbles:
   a) Ensure your vacuum pump can achieve at least 29 inches of mercury (1 Bar / 100 kPa).
   b) Leave enough room in the container for material expansion.
   c) Continue vacuuming until the mixture rises, breaks, and falls, then vacuum for another minute.

Note: Refer to the Fig. 2.7 during the degassing procedure.

III. Pouring the Ecoflex™ Mixture

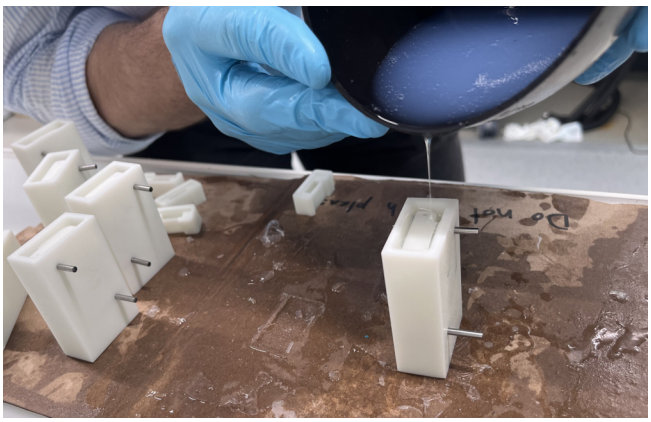
1. Secure the mold in a vertical position and slowly pour the Ecoflex mixture until the mold's top surface is reached.
2. Allow the silicone rubber to cure vertically for approximately four hours.

(a) Mixed EcoFlex components.



(b) Degassing the mixture.

Figure 2.7: Preparing the Ecoflex silicone rubber.



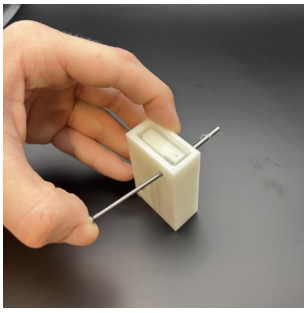
(a) Pouring silicone rubber.

(b) Poured mold.

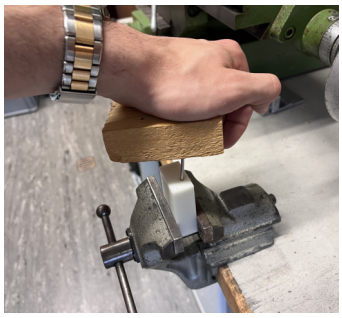
Figure 2.8: Pouring the Ecoflex mixture into the mold.

IV. Removing the Sacrificial Parts

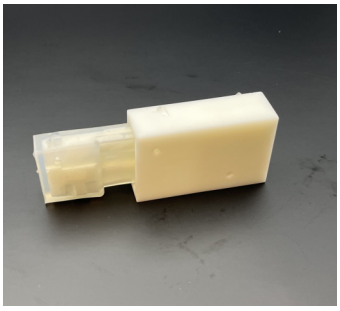
1. Once cured, remove the dowel pins (Fig. 2.9 a).
2. Gently remove the silicone blank from the mold (Figs. 2.9 b and 2.9 c).
3. Cut an opening for the transducer connector as shown in Fig. 2.10.



(a) Removing dowel pins.

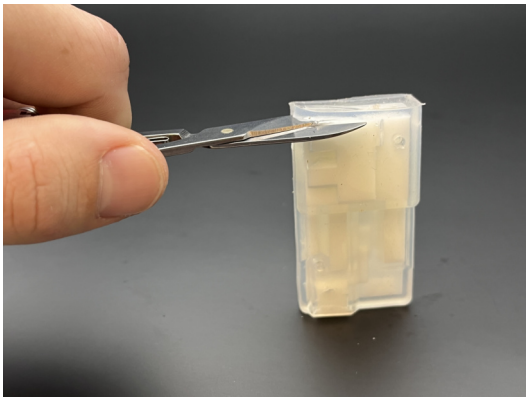


(b) Removing the silicone blank.

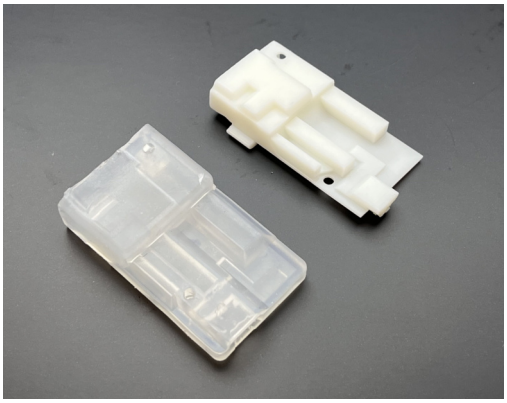


(c) Extracted silicone blank.

Figure 2.9: Removing the silicone blank from the mold.



(a) Cutting the transducer connector opening.



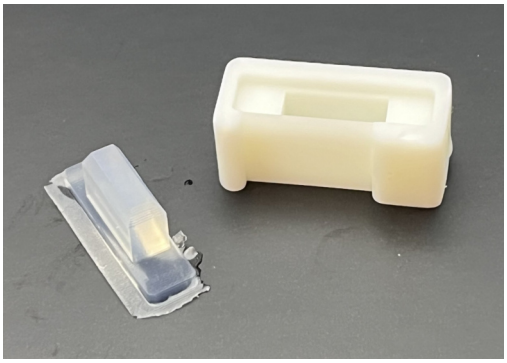
(b) Silicone package and removed plastic insert.

Figure 2.10: Preparing the final silicone package.

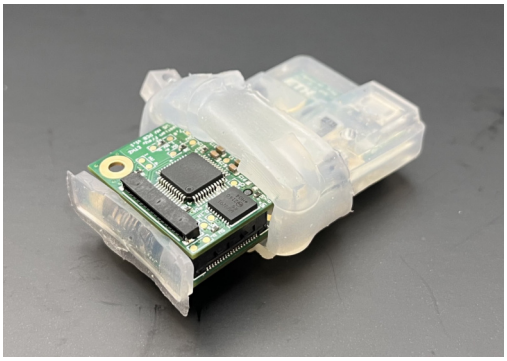
2.6.3 Producing a Silicone Cap

For additional protection, consider fabricating a simple silicone cap to be inserted between the HV and acquisition PCBs near the transducer connector (Fig. 2.11 b). The 3D model for the mold is available in the repository (file `wulpus_silicone_cap_mold.stl`).

Follow the same production steps outlined in Sect. 2.6.2, using Fig. 2.11 as a reference.



(a) Silicone cap and mold.



(b) Cap inserted into WULPUS.

Figure 2.11: Silicone cap for the WULPUS probe.

Chapter 3

How to use the probe?

This chapter explains the needed steps to get started with the WULPUS probe.

The first section focuses on the hardware requirement and describes the equipment you need to take the first measurements. The second section discusses necessary software installations and the WULPUS graphical user interface (GUI). The last section guides you through the test measurement.

3.1 Hardware requirements

To conduct US measurements with WULPUS, you will need the components listed in the following (see figure 3.1).

1. Acquisition PCB
2. High Voltage PCB
3. Your ultrasound transducer (with compatible connector)
4. nRF USB Dongle
5. Windows (or Linux) computer/laptop with USB port
6. Power supply: USB cable/battery/lab supply

Note: The HV PCB is equipped with a 16-pin **DF52-16S-0.8H** connector. To connect a custom transducer to this connector, the user should utilize a compatible **DF52-16P-0.8C** mating connector, along with pre-crimped **DF52-2832PF1571-28A9-300** cables. The pre-crimped cables. should then be directly soldered to the transducer terminals.

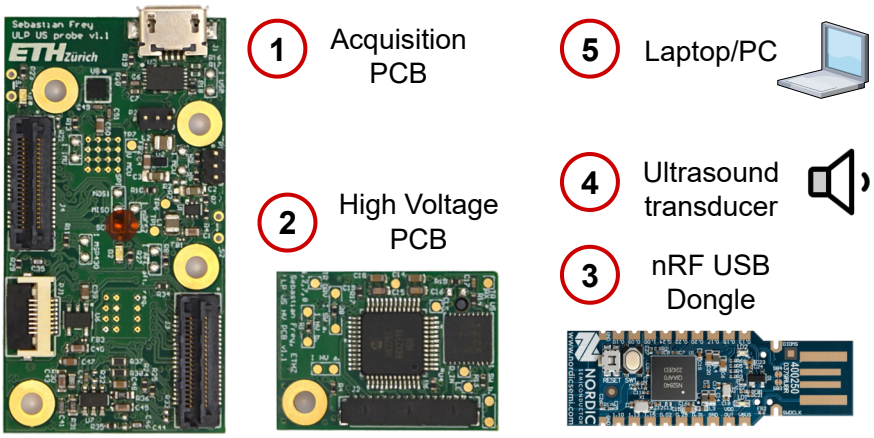


Figure 3.1: Overview of the hardware components. The image of the USB Dongle is adapted from [6].

3.2 Software requirements

The GUI of WULPUS uses the following key technologies (among others):

1. **Python** 3.9
2. Interactive **Jupyter Notebook**
3. **Matplolib** visualization library
4. **IPyWidgets**/**ipympl** backend for interactive Matlab features and widgets
5. **Pyserial** for serial communication
6. **Scipy** for data processing
7. **Multithreading**

The following guides the user step by step through the requirements installation:

1. Install Anaconda packet manager https://docs.conda.io/projects/miniconda/en/latest/
2. Download WULPUS GitHub repository https://github.com/pulp-bio/wulpus
3. Find **requirements.yaml** file in **sw** folder
4. Open terminal (Windows: Anaconda Prompt) and navigate to **sw** folder
5. Execute the following command to create environment that includes the necessary packages:

```
conda env create -f requirements.yml
```

6. In a new terminal launch

```
conda activate wulpus_env
```

and then start Jupyter Notebook server using the command

```
jupyter notebook
```

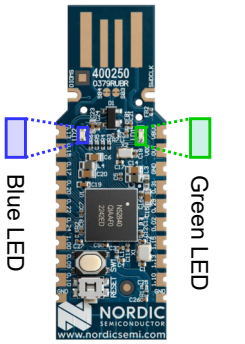
7. The command above opens a webpage. Navigate to **sw** folder and click on **wulpus_gui.ipynb**. Now we are all set up to make a test measurement, which will be described in the next section.

3.3 GUI Overview

Once a user have a WULPUS device and installed all the software dependencies, he or she is ready for running the first measurement. In this section, we will give a short overview of the main GUI. To get started, open an example jupyter notebook called **wulpus_gui.ipynb** (described above) and follow the instructions in the notebook till you launch the main GUI.

3.3.1 Connecting to the probe

Before proceeding, plug the flashed USB dongle into the computer you want to use for the data acquisition. Then, power the WULPUS probe. The onboard LEDs of the dongle indicate whether the bluetooth connection has been successfully established or not (see Fig. 3.2 for details).



State	Description
Green LED off	WULPUS is disconnected
Green LED on	WULPUS is connected
Green LED blinking	Ultrasound data transmission
Blue LED changes state	Configuration package or restart command received from the PC

Figure 3.2: LED locations on the USB dongle and their indications. Note: after running at least one measurement, the green LED does not show the BLE connection status anymore. It just keeps the last state when the stop command arrived.

Note: If the dongle did not connect successfully to the probe, consult chapter 6.

After stablishing a successful connection between the dongle and the probe, switch to the main GUI in the jupyter notebook.

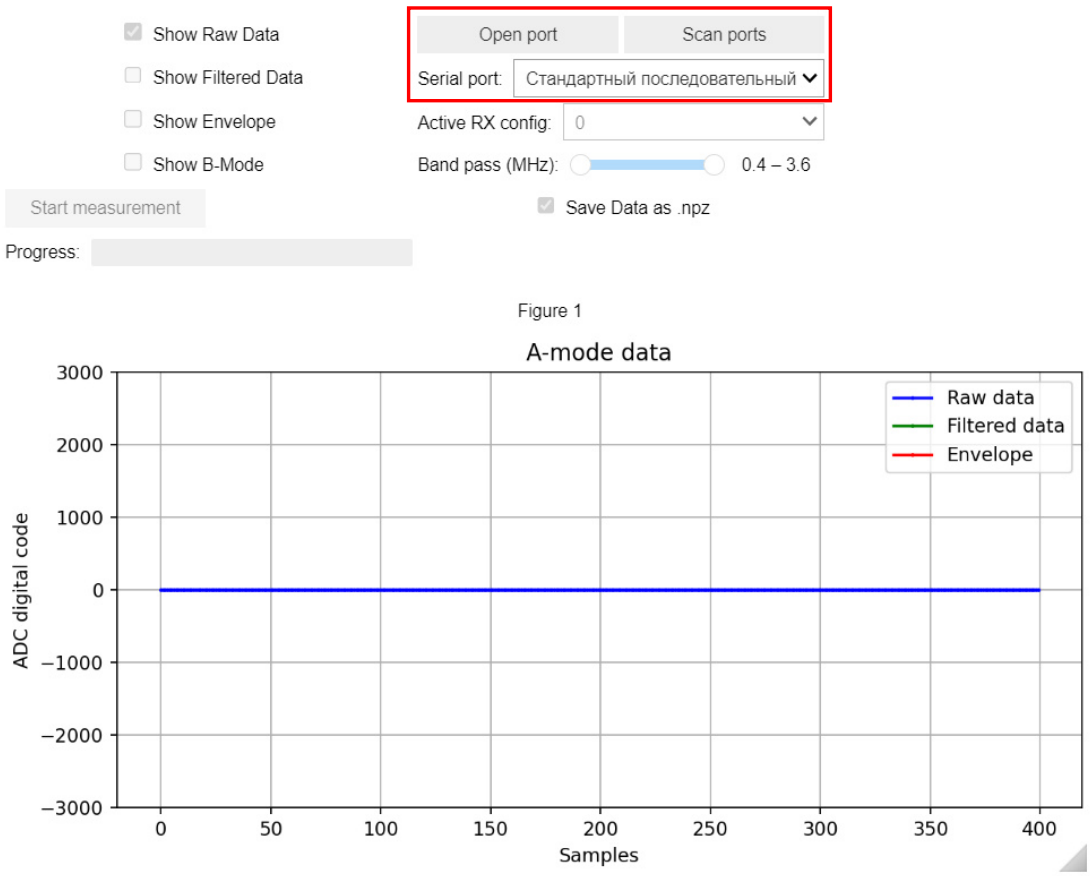


Figure 3.3: The connection section of the main GUI (highlighted in red).

Fig. 3.3 shows a screenshot of the GUI in its initial state and highlights the toolbar for connecting to the USB dongle. With these controls, do the following:

1. **Scan ports**: scan for serial devices available in the operation system and update the internal list of ports.
2. Use the **Serial port** dropdown to choose your device from the list.

   Note: The name of the dongle in the list isn't shown as "WULPUS_X" directly. You need to figure out the right port name e.g. by comparing the list once before and after plugging the dongle in.

3. **Open port**: open a connection to the selected serial device (the USB dongle).

After opening the correct COM port, the GUI is successfully connected to the dongle, and the measurement can be triggered.

3.3.2 Starting a measurement

> **Note:** From now on, we call **acquisition** a certain number of samples with a period between them determined by the ADC sampling frequency. Moreover, we refer to a set of acquisitions as **measurement**.

In this section of the main GUI (highlighted in Fig. 3.4), a user can manage the measurement. Corresponding control buttons are available only when the dongle is connected (COM port is selected and opened).

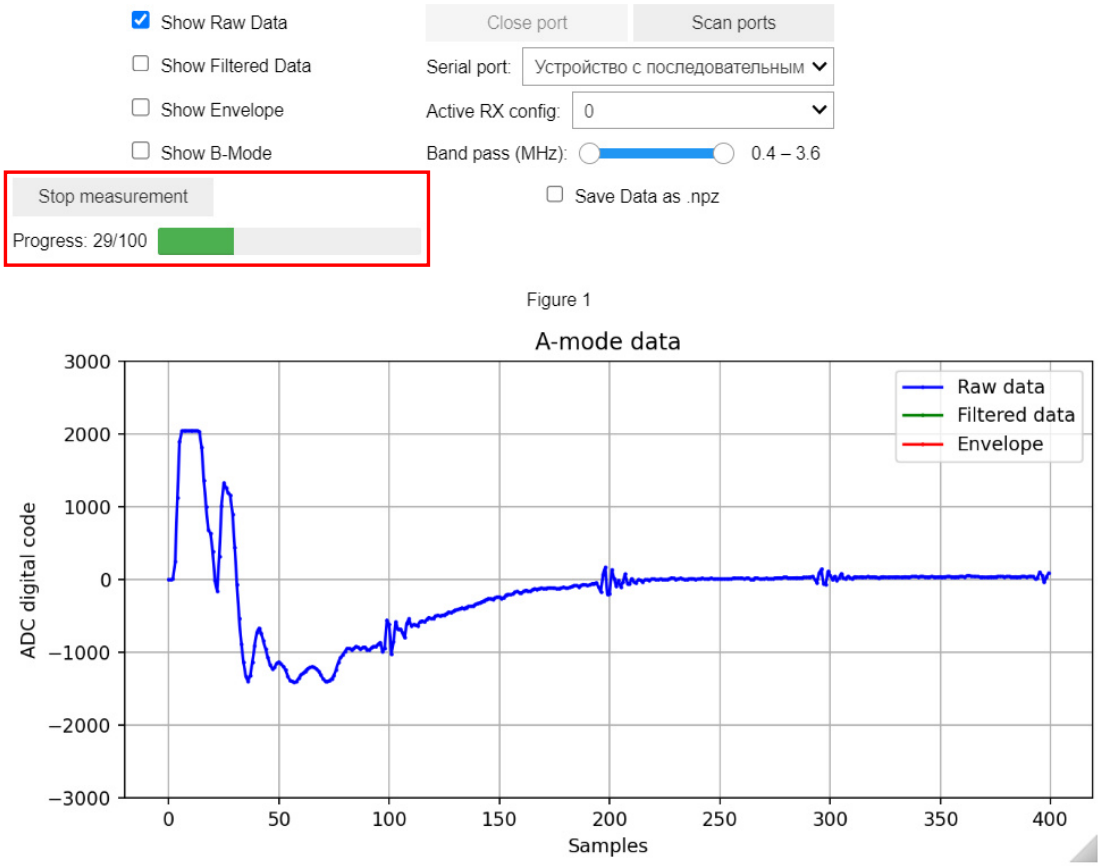


Figure 3.4: The measurement section of the main GUI. The measurement captures four vertical scatters (10 mm spacing) of the CIRS 040 GSE phantom.

To proceed with the measurement, follow the steps below:

1. Start measurementby clicking on the **Start measurement** button.
2. Observe the progress of the current measurement in the **Progress** bar.
3. Either stop the measurement in the middle of the process by clicking on the **Stop measurement** button or wait until the programmed number of acquisitions is collected. In the last case, the measurement will be stopped automatically.

Note: After stopping the measurement (by pressing a button or automatically), the green LED will normally preserve its last state (on or off). At this point, the green LED does not represent connection status (like after a power up), and a user can start a new measurement safely even if the green LED is off.

Note: Fig. 3.4 also shows the real raw ultrasound data acquired from CIRS 040 GSE phantom with 2.25 MHz linear array transducer. To get more information about connecting a custom transducer to WULPUS probe and configuring the custom measurement, refer to section 5.1.

3.3.3 Visualizing the data in real-time

In the next section of the main GUI, which is displayed in Fig. 3.5, a user can configure the settings of real-time data visualization.The following functionality is available:

- **Show Raw Data** checkbox toggles visibility of the raw acquired echo signal.
- **Show Filtered Data** checkbox toggles visibility of the band pass filtered echo signal.
- **Band pass (MHz)** slider adjusts the pass band of the filter.
- **Show Envelope** checkbox toggles visibility of the envelope extracted from the filtered signal.
- If multiple TX/RX configs are active, the one to be plotted can be chosen with the **Active RX config** dropdown menu.

  Note: See section 4.1 for more details about RX/TX configurations.

- There are two different plotting methods: Single channel A-mode and 8-channel B-mode. This can be toggled with **Show B-Mode** checkbox.

  Note: Visit section 4.1.3 for more details about B-mode option.

The options described above can be accessed while the measurement is running. When the measurement is stopped, visualization settings can not be changed.

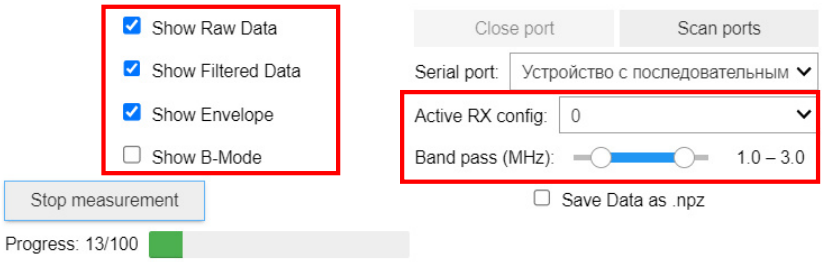


Figure 1

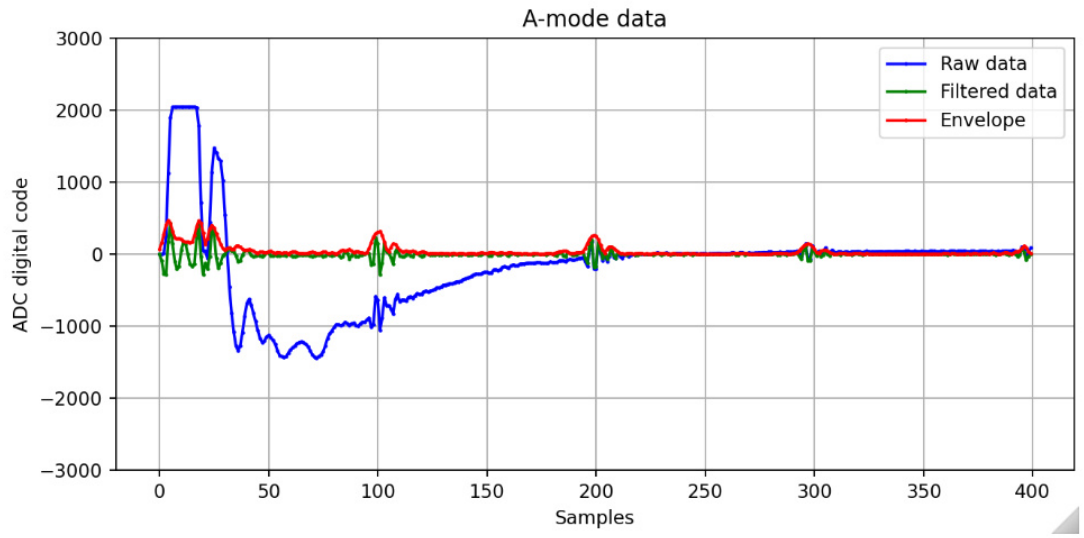


Figure 3.5: The plotting sections of the main GUI.The measurement captures four vertical scatters (10 mm spacing) of the CIRS 040 GSE phantom.

3.3.4 Saving and loading the data

With the help of the dedicated checkbox (**Save data as .npz**, see Fig. 3.6), a user can instruct the GUI to save the raw data to a file after completion of the measurement. The file will be located in the same directory as the jupyter notebook and named **data_x.npz** where **x** starts from 0 and automatically gets incremented. If there is already a file with the name **data_x.npz**, the GUI will increment **x** until there is no file with this name and then save the data. Please, remember to activate the checkbox in advance, since it will be automatically disabled after the measurement completed (stopped or finished automatically).

Note: Only the raw ultrasound data will be saved. The filtered data and envelope are not saved.

The concept of the **.npz** files and their content is explained in the test juputer notebook.

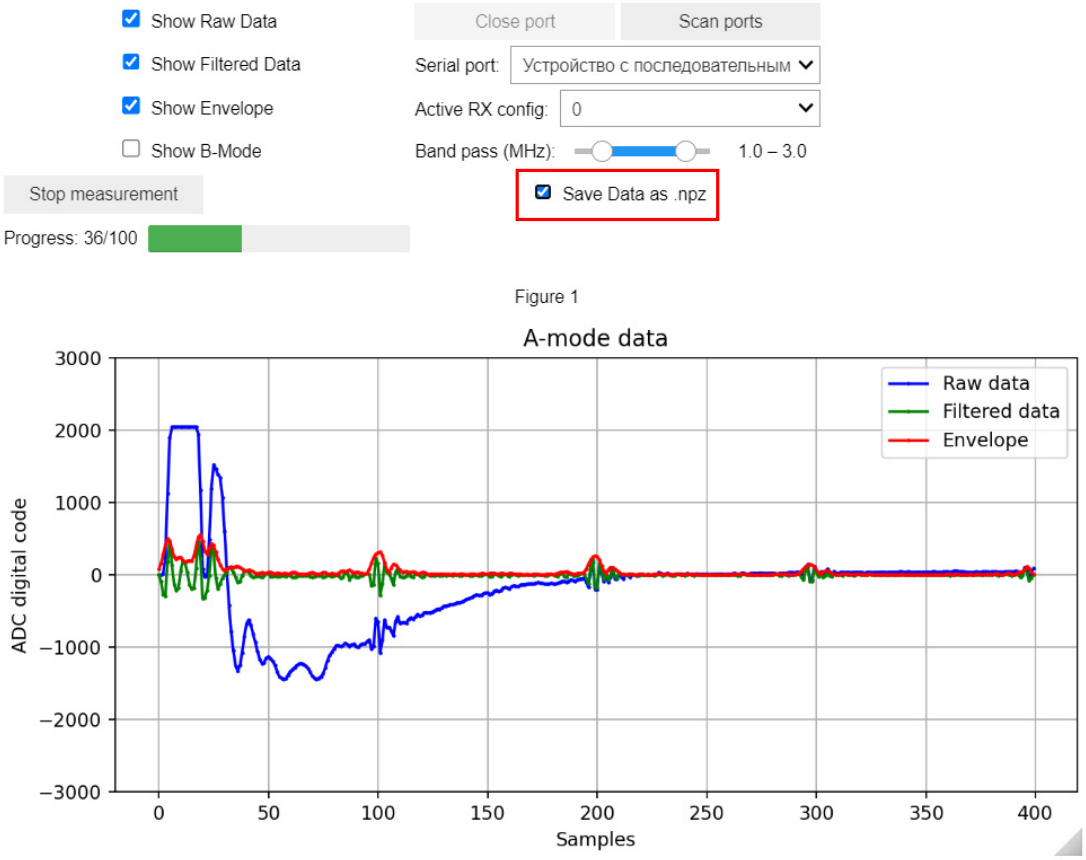


Figure 3.6: The saving section of the main GUI.

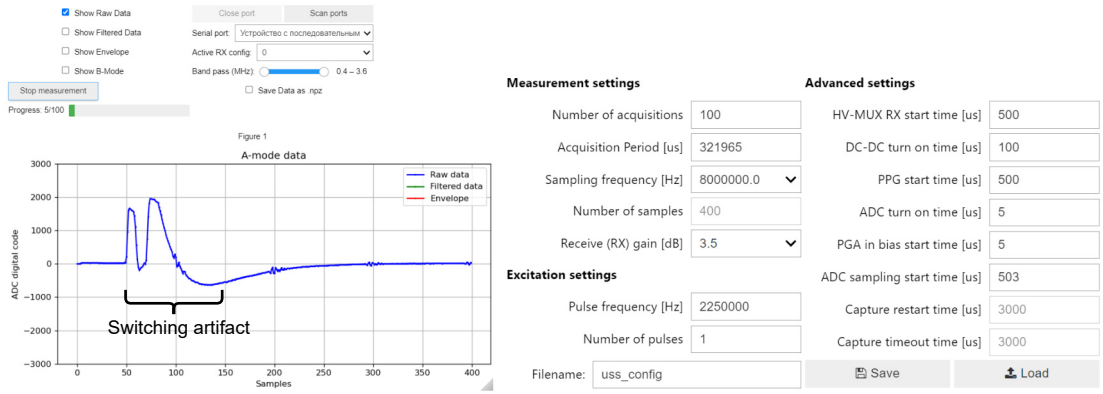
Please, refer to the repository.

3.4 MUX artifact mitigation

In order to switch between the TX and RX phases of an acquisition, the WULPUS system uses a high-voltage multiplexer. This multiplexer behaves non-ideally and introduces some charge injection during the switching event, leading to a voltage spike artifact that will be picked up by the ADC during acquisition. The more WULPUS channels are connected to receive, the larger is the artifact.

Figure 3.7a displays the MUX artifact during an acquisition when a 2.25 MHz linear array transducer (we use only 8 channels) was used. Eight channels are excited at transmit and the same 8 channels are activated at receive phase resulting in the largest-possible switching artifact (worst case). The signal was acquired from the vertical nylon scatters (0.1 mm diameter, 10 mm spacing) of the CIRS 040 GSE phantom in the same position

as for the Fig. 3.4, 3.5 and 3.6. To collect the data shown in the Fig. 3.7a, we used the settings of the ultrasound subsystem depicted in Fig. 3.7b.



(a) Annotated WULPUS GUI screenshot. (b) Measurement settings.

Figure 3.7: High voltage multiplexer switching artifact with default settings.

There are three main ways to mitigate and adjust this artifact, which will be described individually in the following subsections.

3.4.1 HV MUX start time

Since a user can configure the time of switching from TX to RX state (see chapter 4 for details), to mitigate the artifact, we can reduce the parameter **HV-MUX RX start time** from the default value of 500 us to e.g. 494 us. We should not make this value too low, otherwise switching will intersect with the pulsing event, and the transducers will not be excited.

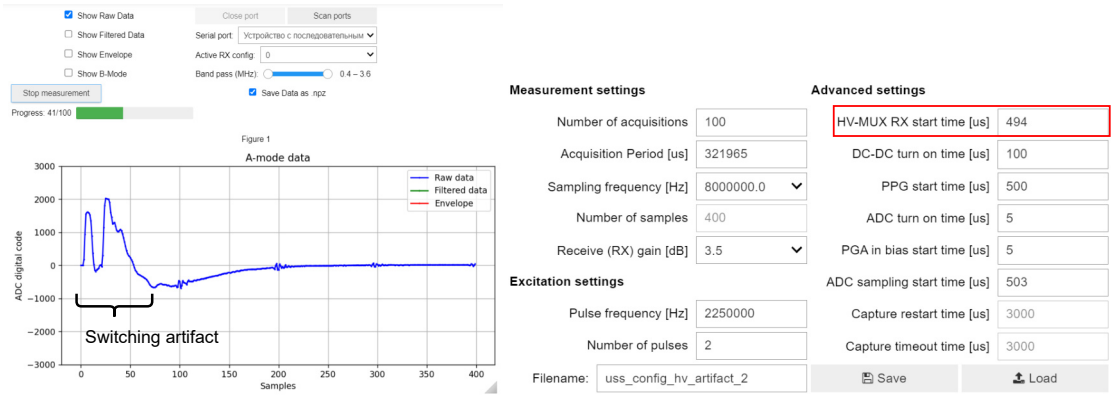
Fig. 3.8a and 3.8b demonstrate that early switching shifts artifact to the left, while the back-scattered echo signals remain on their places on the timeline.

3.4.2 ADC sampling start time

Another way of adjusting the artifact is to tune the **ADC sampling start time**, such that the ADC only start sampling after the whole or most of the artifact has passed. Below we compare the acquisitions with two different ADC sample start timings, with the default and with a delayed start.

Figure 3.9a shows the artifact with an additionally delayed **ADC sampling start time**. The starting time is set in the red box (Fig. 3.9b to 509 μs.

As the **ADC sampling start time** in 3.9a is set to a later time, the MUX artifact moves to the left. This method doesn't reduce the dimensions of the artifact (duration



(a) Annotated WULPUS GUI screenshot.

(b) Measurement settings.

Figure 3.8: High voltage multiplexer switching artifact with early multiplexer switching (**HV-MUX RX start time** = 494 us).

or amplitude), but tries to reduce the space the artifact takes up in the number of samples we set up. The downside is, that if important data were overlaid over the artifact and we removed the artifact entirely by starting the ADC even later, we may omit the important data. This would e.g. be the case if we are measuring shallow arteries, where the first wall has reflections which lie in the duration of the artifact.

3.4.3 Switching optimization

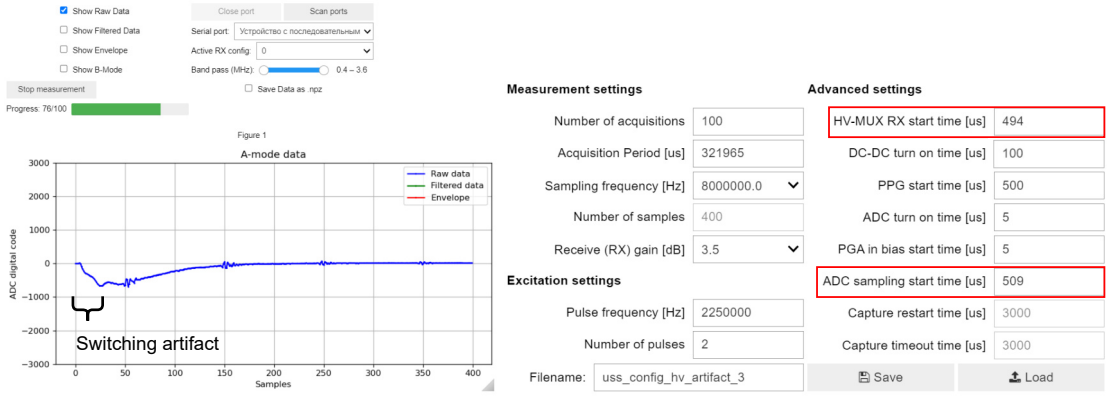
The other method of reducing the effect of the artifact is by actually trying to reduce the dimensions of the artifact.

Since the cause of the artifact is the switching, we have to work on it. The simplest idea may be to reduce the amount of times we switch the channels of the multiplexer by analyzing the sets of transmitting/receiving channels and their intersections. This is what we are trying to do with the **Optimized switching** method of the RX/TX config GUI (see figure 3.10).

> **Note:** The RX/TX configuration GUI and its functions will be discussed in detail in section 4.1.2.

Through a rather simple algorithm, the WULPUS analyzes the configuration sets for TX/RX channels, and tries to pre-configure the states of the switches, so that the amount of switching during a transition from transmit to receive is minimized. Figure 3.11a shows an example MUX artifact with optimized switching enabled.

As we can see from the Fig. 3.11a, employing an algorithm for switching optimization further reduces the artifact. At this point, we can safely increase the PGA receive gain



(a) Annotated WULPUS GUI screenshot. (b) Measurement settings.

Figure 3.9: High voltage multiplexer switching artifact with early multiplexer switching and delayed ADC sampling start (**ADC sampling start time** = 509 us).

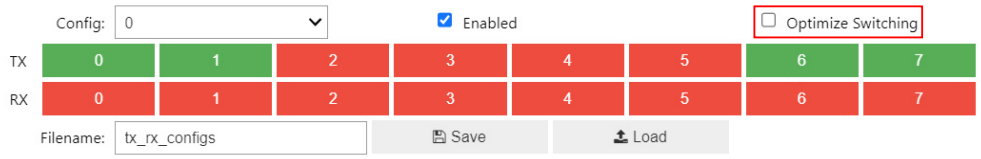
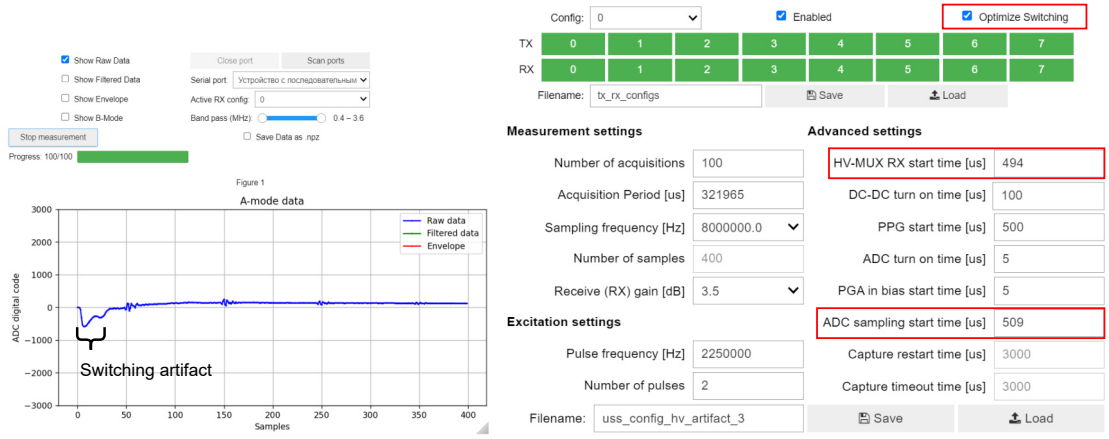


Figure 3.10: Location of the optimized switching choice in the RX/TX configs GUI

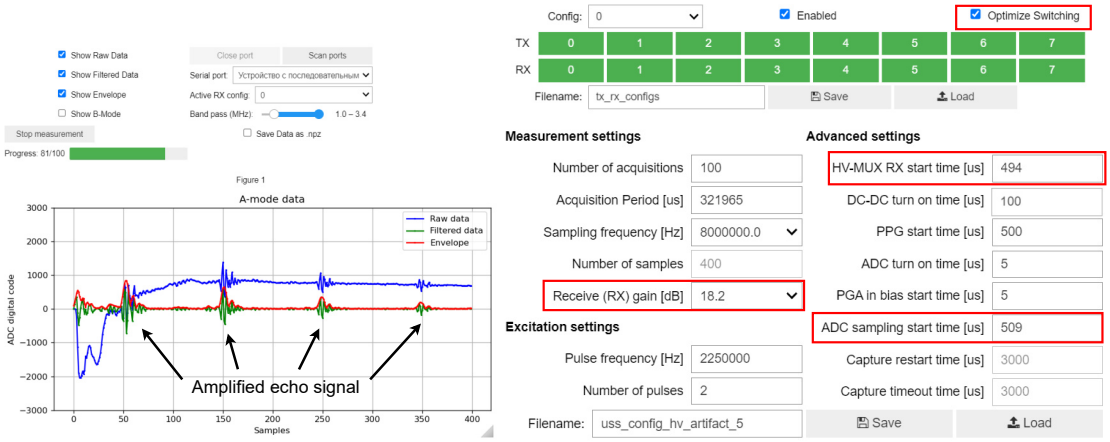
to amplify the echo signal. Fig. 3.12a demonstrates the amplified echoes obtained under the cumulative acquisition settings summarized in Fig. 3.12b . Although the artifact is still presented in the raw data, the band-pass filtered signal shows four target echoes with high signal to noise ratio.

Through an employment and combination of the described methods, the switching artifact of the HV multiplexer can be greatlyreduced, allowing for accurate data collection.



(a) Annotated WULPUS GUI screenshot. (b) Measurement settings

Figure 3.11: High voltage multiplexer switching artifact with all previous settings and enabled **Optimize Switching** option.



(a) Annotated WULPUS GUI screenshot. (b) Measurement settings

Figure 3.12: High voltage multiplexer switching artifact with all previous settings, enabled **Optimize Switching** option and increased receive gain (18.2 dB).

Chapter 4

Advanced Settings

This chapter describes the advanced settings of the WULPUS platform which can be tuned for a specific transducer and application.

4.1 Transmit/Receive (TX/RX) configurations

4.1.1 Theoretical overview

To perform an acquisition, WULPUS needs to know on which channels to transmit (receive) the pulses. This is done by supplying an array of TX (RX) channels. These two inputs (TX and RX array) determine which elements of the transducer array are active during acquisition.

TX channels
Multiple TX channels can be activated at a time, reflecting the elements of the transducer array that will transmit a signal simultaneously. This array of TX channels allows for a broader or more focused area of signal transmission, depending on the experiment's requirements.

RX channels
Multiple RX channels can be input at a time, reflecting the elements of the transducer array that will be connected in parallel to receive a signal simultaneously.

Note: Since WULPUS has a single ADC channel, multiple channels can not be sampled simultaneously. Instead, when activating a few channels in parallel for the reception, the echo signals from all the connected channels (transducer elements) will be summed up in the analog domain and then sampled by the ADC.

One set of TX/RX channels can be either TX or RX channels or both simultaneously. This set is also called "configuration set".

WULPUS supports multiple (16 max.) configuration sets. These configurations are stored and will be executed in a round-robin fashion. This means that WULPUS will cycle through the configurations sequentially, using each set for signal transmission and/or reception before moving to the next. This approach is beneficial for conducting experiments that require data from e.g. multiple transducer channels or for quickly iterating through different test scenarios without manual reconfiguration.

When setting up configurations, one must consider their experimental setup and the specific TX/RX channels that need to be activated. For instance, if the experiment involves scanning an area with a transducer array, the user would enable multiple TX channels to cover the area and select an appropriate RX channel to capture the reflected signal.

4.1.2 Programming TX/RX configurations via GUI

A user can use either the Python API or a small GUI to program TX/RX configurations. A GUI-based example is provided in Fig. 4.1. The configuration shown matches the `Transmit only` example visualized later in 4.1.3.

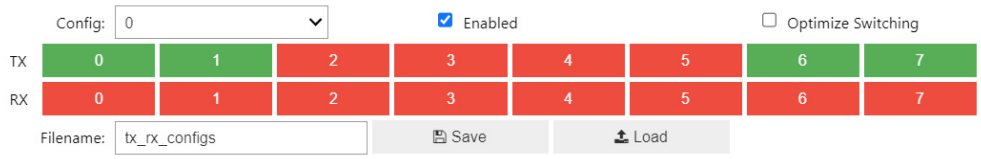


Figure 4.1: The TX/RX configuration GUI. In this example, configuration set 0 is selected, with channels 0, 1, 6 and 7 set to transmit, and no channels selected for receive.

Note: In the Python API, we are always talking about channels 0-7 (Zero-Indexed), whereas in the hardware design files we talk about channels 1-8.

In this user guide, we use the same numbering as in the Python API.

Configuration selection and activation
In the top part of the GUI, we have a dropdown list `Config` of all available configuration sets. Choose the configuration set you want to edit here and enable it via the `Enable` checkbox. A user can program multiple configurations which will then be executed in a round robin fashion starting from the config with id=0 and increasing for the next acquisition event(s).

Channel selection
The middle part presents a series of buttons for TX and RX channels. These buttons can be toggled to activate (green color) or deactivate (red color) the needed `TX`/`RX` channels in the configuration set.

Saving configurations to a file
The lower part of the GUI contains file management controls. A text box is provided to specify a `Filename` for saving configurations. `Save` and `Load` buttons facilitate the storage and retrieval of the configuration settings. The currently enabled list of configuration sets will be saved in a human readable `.json` format.

> **Note:** Disabled configuration sets will not be saved.

4.1.3 Example configurations

Three examples of using the TX/RX configurations are visualized in Fig. 4.2.

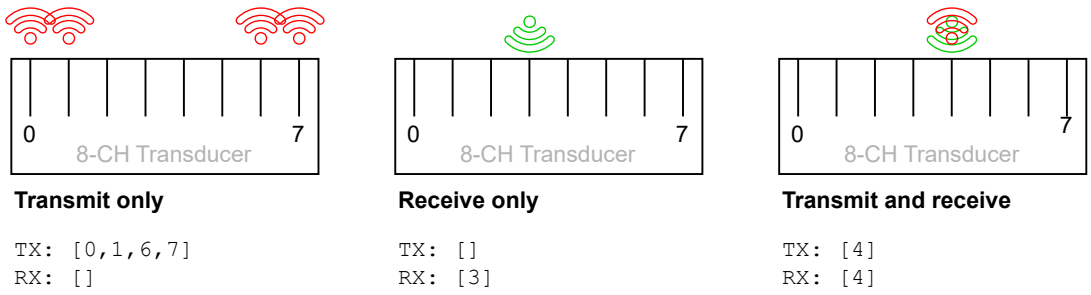


Figure 4.2: Three examples of using TX/RX configurations where the transmitting channels are indicated in red, the receiving channels in green.

In the first example denoted as **Transmit only** (Fig. 4.2, left), WULPUS is configured to send out pulses on channels `[0,1,6,7]` and not receive on any channel. This configuration corresponds to the GUI settings demonstrated in Fig 4.1. In the second **Receive only** example (Fig. 4.2, center), WULPUS is configured to receive on channel `3` only. In the third example called **Transmit and receive** (Fig. 4.2, right), WULPUS is configured to send and receive on channel `4` simultaneously.

> **Note:** In reality, the system doesn't transmit and receive exactly at the same time. "Simultaneously" means that the same channels are selected for TX and RX within the same excitation/readout cycle.

These three configuration sets could be assigned to different config ids (e.g to 0, 1, 2) and put in one configuration file. If all configurations are enabled, WULPUS would then execute them in a round-robin manner:

```
Transmit only -> Receive only -> Transmit and receive ->
 -> Transmit only -> Receive only -> Transmit and receive -> ...
```

Note: A user can program a sequence of a maximum of 16 configuration sets.

"B-mode" data acquisition and visualization

A user can potentially program 8 TX/RX configurations where each `n`'th config (with an id = `n`) receives the echo signal from the `n`'th channel of the linear array transducer (`n` ranges from 0 to 7). In this case, if the channels are ordered (pin mapping from the transducer's to WULPUS's channels is correct), the GUI can visualize the ultrasound echo data as a 2D image. An example of such an image is shown in Fig. 4.3.

Note: Although we often call it "B-mode", the GUI does not perform beamforming. Instead, the raw ultrasound data is band-pass filtered, and an envelope (A-scan) is extracted. Later, when all 8 envelopes (A-scans) are collected, they are stacked into a 2D matrix which is visualized.

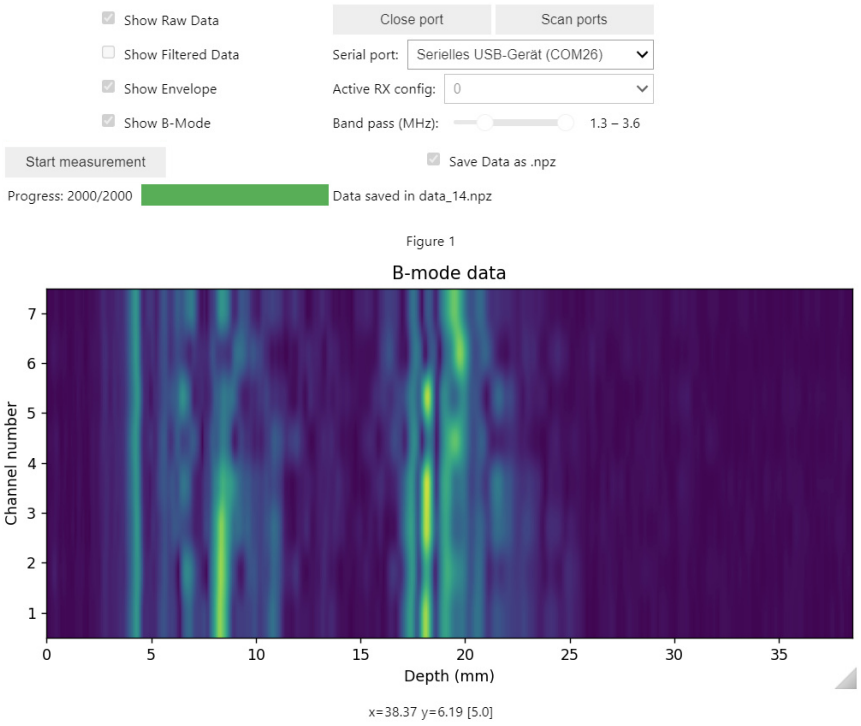


Figure 4.3: The main GUI in B-mode while imaging a carotid artery. A user can activate this mode by clicking on **Show B-mode** checkbox.

Precisely, the GUI takes data from the 'n'th config set and plots it along the x-axis (depth) at coordinate y=`n` (there are eight discrete y-coordinates). A user thus needs to

program eight TX/RX config sets, where for each config set one channel is activated to RX. For the best image quality, we suggest activating all the channels during TX.

For the example measurement shown in Fig. 4.3, the employed TX/RX configs were the following (see Tab. 4.1):

Config set no.	0	1	2	3	4	5	6	7
TX channels	$0-7$	$0-7$	$0-7$	$0-7$	$0-7$	$0-7$	$0-7$	$0-7$
RX channels	0	1	2	3	4	5	6	7

Table 4.1: RX/TX configs for B-mode

4.2 Configuration of Ultrasound Subsystem

Note: As defined earlier, we call **acquisition** a certain number of samples with a period between them determined by the ADC sampling frequency. Moreover, we call a set of acquisitions a **measurement**.

After the power-up, the WULPUS probe waits for the configiration package to arrive from the host PC (from Python GUI). This configuration package contains TX/RX configurations (see 4.1.2) along with the settings of the ultrasound subsystem responsible for managing all the steps of the acquisition (e.g. pulse generation, start ADC sampling etc). Tab. 4.2 contains the full list of the settings available for the user.

In total, there are three groups of configuration settings which are described in detail below.

Measurement settings
The first group is responsible for configuring general measurement settings. Particularly, `meas_period` defines the period between the consecutive acquisitions and is inversely proportional to the Acquisition Per Second (APS) rate. In its turn, `num_acqs` defines the number of acquisitions to be collected by GUI before stopping the measurement.

Note: `num_acqs` is relevant only for the Python GUI. The GUI is responsible for starting the WULPUS probe and later stopping it (thereby terminating the data collection) when the programmed number of acquisitions is received.

The ADC sampling frequency can be adjusted by changing the parameter `sampling_freq`.

Note: The number of samples per acquisition is currently fixed to 400 samples and can not be changed via the GUI.

Parameter name	Display name (in GUI)	Description/Note
Acquisition settings		
num_acqs	Number of acquisitions	N acquisitions to collect
meas_period	Measuring period [us]	Time between acquisitions
sampling_freq	Sampling frequency [Hz]	ADC Sampling frequency
num_samples	Number of samples	K samples per acquisition
rx_gain	RX gain [dB]	Gain of the MSP430 PGA
Excitation settings		
pulse_freq	Pulse frequency [Hz]	Frequency of the pulses
num_pulses	Number of pulses	L pulses to generate
Advanced settings *(special time events)*		
start_hvmuxrx	HV-MUX RX start time [us]	MUX switches from TX to RX
dcdc_turnon	DC-DC turn on time [us]	DC-DC converter is enabled
start_ppg	PPG start time [us]	Pulse Generation starts
turnon_adc	ADC turn on time [us]	ADC is turned on
start_pgainbias	PGA in bias start time [us]	Input biasing is enabled
start_adcsampl	ADC sampling start time [us]	ADC starts sampling
restart_capt	Capture restart time [us]	(reserved, not used)
capt_timeout	Capture timeout time [us]	(reserved, not used)

Table 4.2: Ultrasound subsystem configuration parameters and their descriptions.

Excitation settings
The second group of settings determines the excitation pattern for the ultrasound transducer. The WULPUS system produces unipolar excitation pulses with an amplitude of $15V$ (from $0V$ to $+15V$). The frequency of the pulses along with the number of pulses can be changed via GUI.

Note: The duty cycle of the pulses is currently fixed to 50% in the API but can potentially be adjusted at runtime.

☐ Show an example Excitation pulse

Advanced settings
The third group is responsible for setting the precise time marks of the acquisition's events. These timings have to fulfill certain requirements for a successful acquisiton. Since the understanding of these timings is important to get the full potential out of these settings, the principle of how they work is explained further.

A short overview of the timings during one acquisition is displayed in figure 4.4. Preparation for the acquisition starts with the event L_A responsible for turning on the high-voltage DC-DC converter. This converter supplies the $+15V$ power domain used to generate excitation pulses. Next, during the L_B event the MSP430 MCU wakes-up, prepares the ultrasound subsytem for the acquisition and activates the universal ultrasound

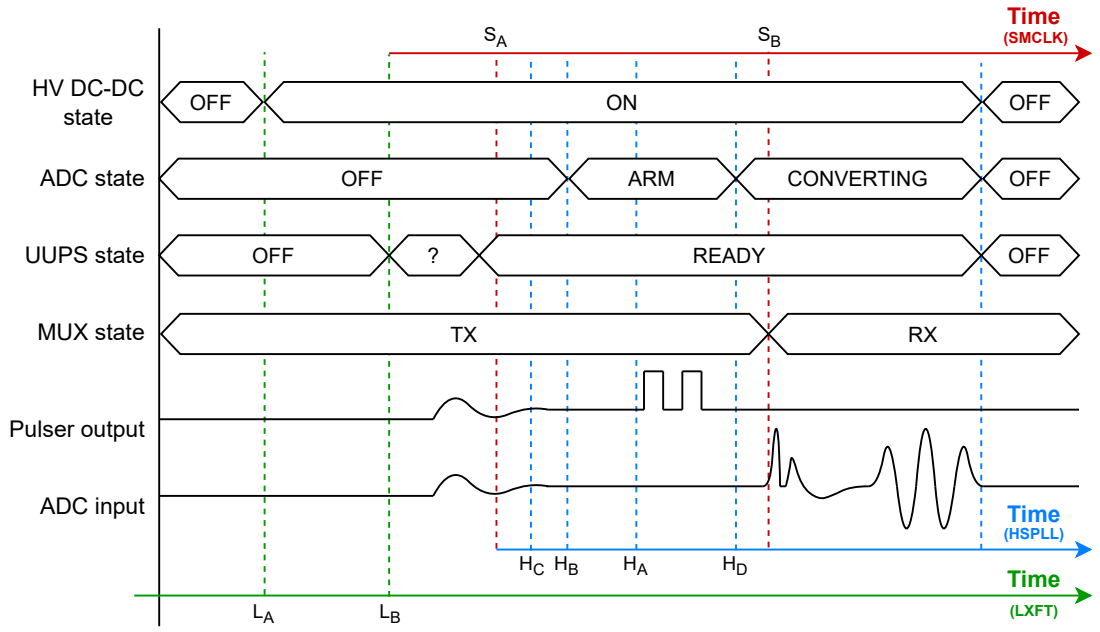


Figure 4.4: Timing diagram for the internal events of the MSP430 MCU during a single ultrasound acquisition. Colors relate internal events to the clock domain: blue color denotes the events whose time stamps are coupled with the high-speed PLL (HSPLL) clock domain of the MSP430, red color corresponds to sub-system master clock (SMCLK), green stands for low frequency crystal (LXFT).

power supply (UUPS) module. This module requires some time to get ready. Therefore, the firmware also initializes a timer (clocked by the SMCLK domain), sets a delay equal to approx. 9 us and starts the timer. When the timer elapses, an event S_A happens. Up until this moment, the UUPS module is typically in the READY state (but if not, we wait further until it is ready). Next, we configure the SMCLK-based timer with the timestamp S_B which will be later responsible for switching the HV multiplexer from the transmit to receive state. Then, we immediately trigger an ultrasound acquisition. Later steps are automatically performed by the dedicated hardware acquisition sequencer (ASQ) of the MSP430 without involving the CPU at all. This means that the hardware automatically applies an RX bias (H_C), turns on the sigma-delta high-speed ADC (H_B), triggers the programmable pulse generator (H_A) and starts sampling the input signal (H_D). The only exception is the event S_B when the CPU wakes up from the timer event to switch the HV multiplexer to the receive state. Finally, the acquisition finishes when all the required samples are acquired, and all the US-related peripheral modules are turned off until the L_A and L_B events happen again in the next acquisition.

A user can flexibly configure the timings of the internal events. The mapping of the internal parameters to the Python API is summarized in the Tab. 4.3

Event	Clock	Parameter in GUI	Function
L_A	LXFT	DC-DC turn on time	Turn on HV DC-DC
L_B	LXFT	Measurement Period	Turn on UUPS
S_A	SMCLK	-	Trigger Acquisition Sequencer
S_B	SMCLK	HV-MUX RX start time	Switch HV MUX to receive
H_A	HSPLL	PPG start time	Trigger pulse generator
H_B	HSPLL	ADC turn on time	Turn on sigma-delta ADC
H_C	HSPLL	PGA in bias start time	Apply receive bias
H_D	HSPLL	ADC sampling start time	Start sampling the input signal

Table 4.3: The six timing points and their corresponding parameters

Note: S_A event is fixed in firmware with respect to the L_B so that $S_A - L_B = 9\ us$.

Note: L_B is a central event of the whole acquisition, since it triggers SMCLK-based timer/events which is responsible for triggering the acquisition sequencer operating in HSPLL domain (see Fig. 4.4).

Note: While changing the Measurement Period (L_B event), a User should also adapt the DC-DC turn on time (L_A event) so that the HV DC-DC converter has enough time to start up and generate the +15V power domain.

While adapting the advanced settings, a user must follow the recommendations below (otherwise, the system will not work):

- The DC-DC converter has to be turned on and settled before starting pulsing.
- The pulsing has to be finished before the MUX switches to RX.
- The PGA input (receive) bias has to be settled before the ADC is triggered.
- The ADC has to be turned on and ready before the ADC is triggered.
- The MUX has to be settled to RX before early enough to capture early echo-signals.
- The previous acquisition must be finished before the next acquisition starts.

Since WULPUS probe uses multiple clock domains originating from different sources (see schematics for details), some variations between the different WULPUS probes may occur such as slight misalignment of the HV MUX artifact caused by the switching event S_B (see Fig. 4.4). For the best results, we recommend starting from the default settings and tuning the **HV MUX RX start time** parameter to get the best results. The default timings are displayed in the Fig. 4.5.

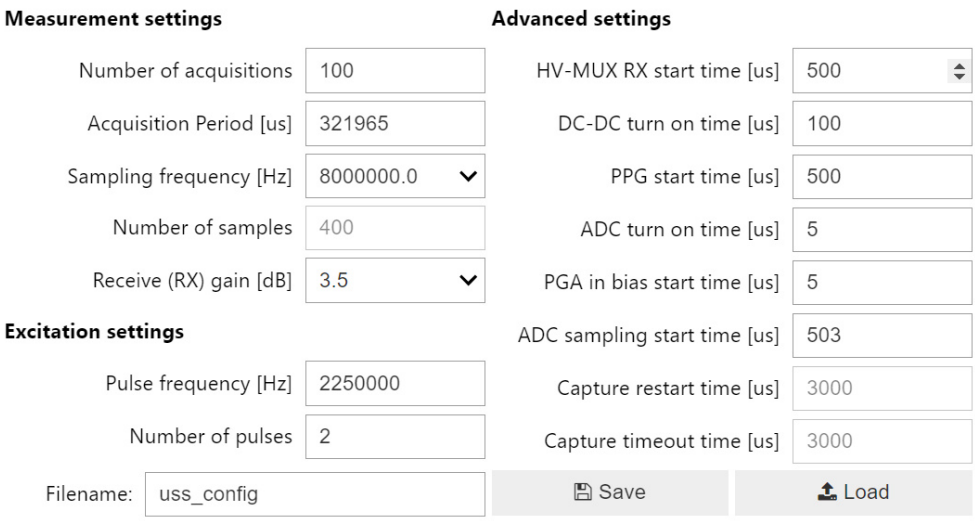


Figure 4.5: Default settings of the WULPUS GUI.

Chapter 5

Example experiments

This chapter describes simple example experiments a user can replicate with a WULPUS probe and with minimal equipment.

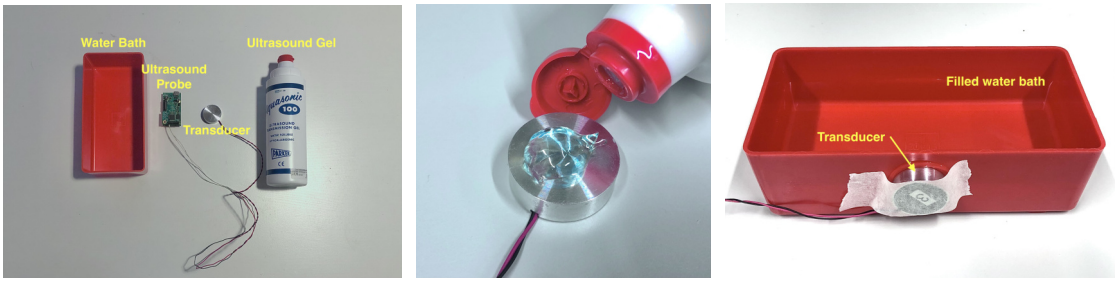
5.1 Water bath

This simple example experiment shows how to acquire data in a water bath setup with simple metal reflectors. To replicate the results, you will need the following materials (key items are visualized in Fig. 5.1a):

- Plastic water bath.
- WULPUS probe (+ micro-USB cable).
- 1 MHz ultrasound transducer (e.g. H2KLPY11000600 from Unictron Technologies).
- WULPUS transducer connector with pre-crimped wires.
- Ultrasound Gel.
- Sticky tape.

We first solder the transducer's wires to the pre-crimped cables, assemble them with the mated mechanical connector (please wire the transducer to channel 8), and then insert the connector into the WULPUS probe. Later, we apply the gel on the transducer as shown in figure 5.1b and attach the transducer to the water bath using sticky tape as shown in Fig. 5.1c.

Next, we plug the USB dongle into the PC. As always, a glowing green LED on the WULPUS probe indicates an established connection, and we are ready to acquire the ultrasound echo data.



(a) Experiment materials (b) Coupling gel on transducer (c) Experiment setup

Figure 5.1: Water bath experiment.

Lastly, we activate the **wulpus_env** as described in section 3.2 and open the GUI in **wulpus_gui.ipynb** notebook located in **sw** folder. Since the default configuration does not match the 1-MHz transducer, we implement the custom settings shown in Fig. 5.2. Particularly, we reduce the ADC sampling rate to 4 MHz, change excitation frequency to 1 MHz and program 10 pulses per single excitation.

Measurement settings

Number of acquisitions	100
Acquisition Period [us]	228885
Sampling frequency [Hz]	4000000.0
Number of samples	400
Receive (RX) gain [dB]	6.8

Excitation settings

Pulse frequency [Hz]	1000000
Number of pulses	11

Filename: uss_config

Advanced settings

HV-MUX RX start time [us]	500
DC-DC turn on time [us]	100
PPG start time [us]	500
ADC turn on time [us]	5
PGA in bias start time [us]	5
ADC sampling start time [us]	503
Capture restart time [us]	3000
Capture timeout time [us]	3000

Save Load

Figure 5.2: WULPUS config for water bath experiment. ADC sampling frequency, excitation waveform and receive gain are modified to match the 1-MHz transducer.

Since we use a single-channel transducer, we program only one TX and RX configuration of WULPUS probe shown in figure 5.3. In our example, the transducer is connected to channel number 7 (remember, in the schematic this is channel 8, see Section 4.1.2).

We then place a metal reflector into the water bath as demonstrated in figure 5.4 and

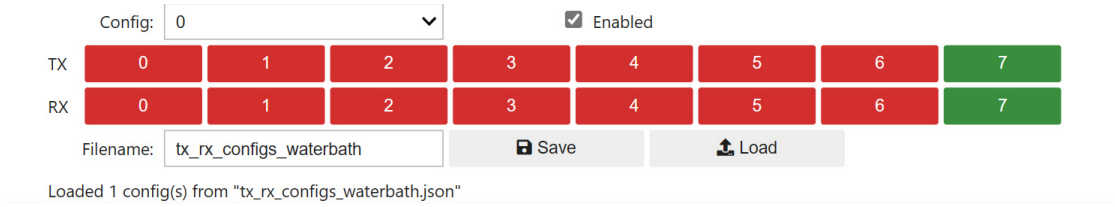


Figure 5.3: RX/TX configuration for water bath experiment. A single WULPUS channel (id=7) is activated for both transmit and receive.

start the acquisition.

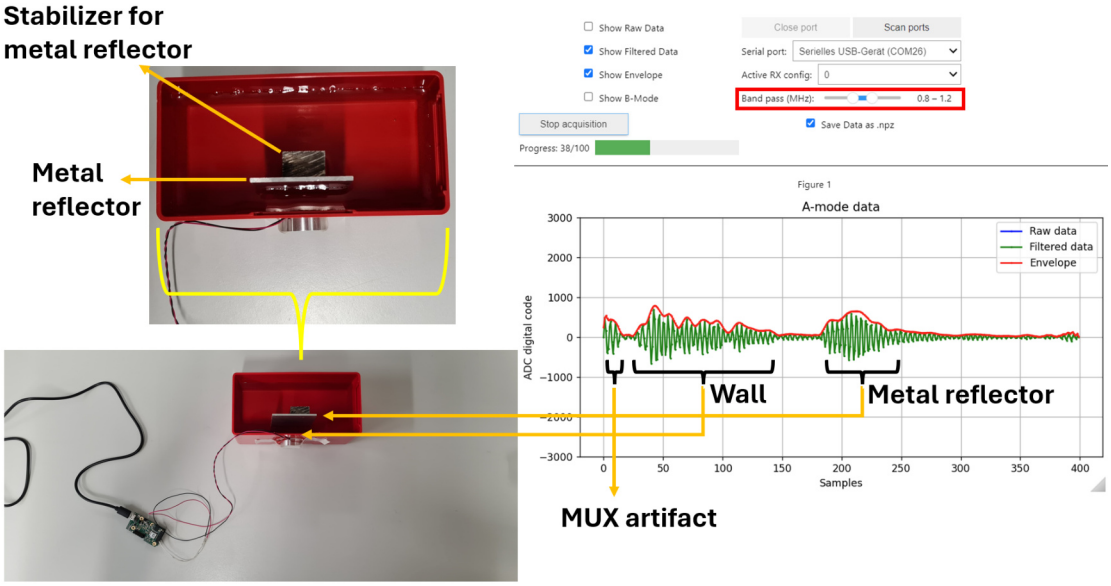


Figure 5.4: Water well experiment. The metal plate reflector is placed approx. 1 cm away from the water bath's wall. The received signal reveals multiple parasitic reflections from the wall and target reflections from the metal plate

The result (we display the filtered data and envelope) is shown in figure 5.4. The red box denotes the frequency band the applied band-pass filter allows to pass through. As we employ a 1-MHz low-bandwidth transducer, we set the low and high cut-off frequencies to 0.8 MHz and 1.2 MHz respectively. The first two reflections correspond to the MUX artifact (see section 3.4 for further information) and the initial reflection when the acoustic wave enters the wall and the bath. The third reflection, located at sample ∼220 of the received waveform, corresponds to the reflection from the metal plate. By varying the position of the metal reflector, this peak changes position accordingly.

Chapter 6

Troubleshooting

If the system does not work (no BLE connection between WULPUS probe and Dongle confirmed by green LED), start debugging the individual components. A high-level overview of the debug action tree is shown in Fig. 6.1. The next subsections guide the user through this tree in detail.

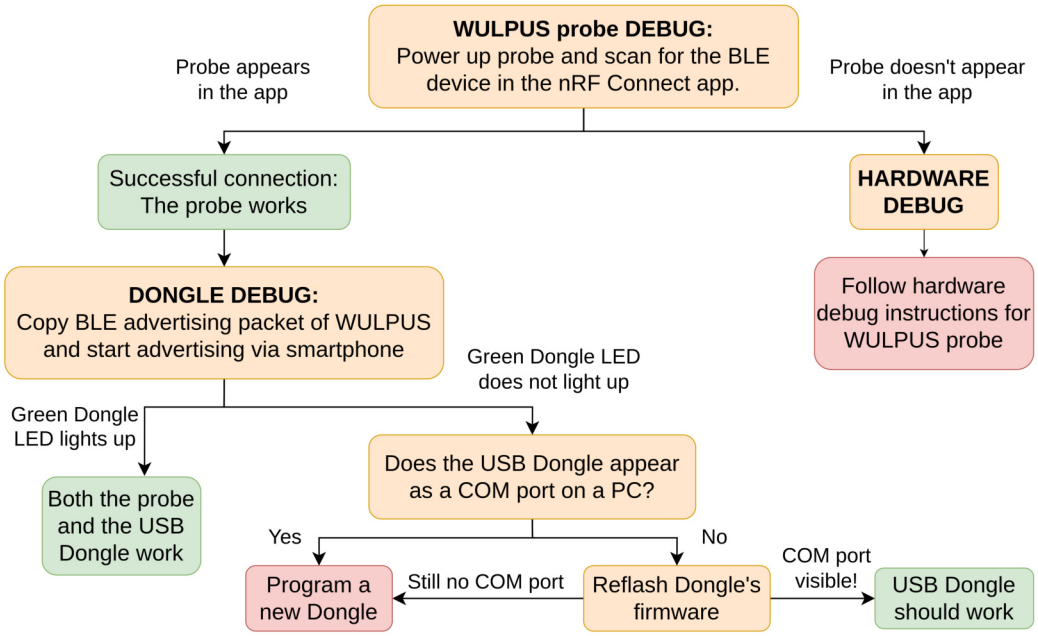


Figure 6.1: WULPUS system debug action tree.

6.1 BLE

BLE connection is the first indicator of the system's malfunctioning. Start with debugging the WULPUS probe.

6.1.1 WULPUS Probe

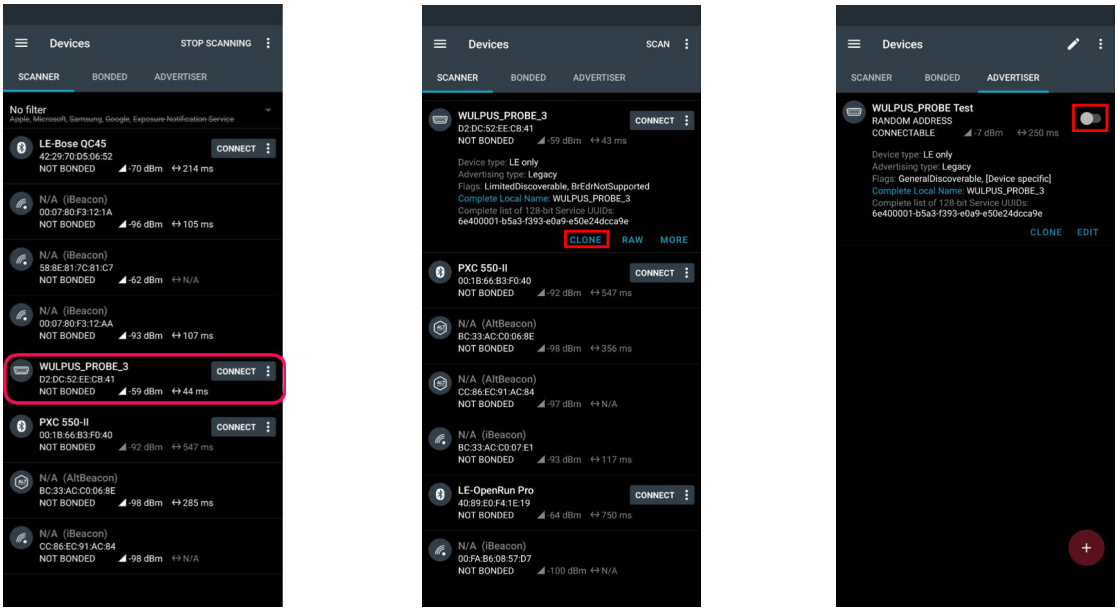
To test the BLE connectivity of the WULPUS probe, go through the following steps:

1. (optional, recommended) Move to an environment free from BLE devices.
2. Power up the WULPUS probe from USB (but do not connect the USB Dongle).
    - If you power up from the Power supply (through the battery connector), configure the power supply to 3.7 - 4.1 V output with a current limit of at least 150 mA.
    - After the power-up, the power consumption of the configured and unpaired WULPUS probe should be around 13-15 mA.
3. Take an Android smartphone and download nRF Connect application.
4. Open the App and scan for BLE devices:
5. Find your device named **WULPUS_PROBE_X** (X is a unique number) as in figure 6.2a. If your device does not appear, refer to section 6.2.1.
6. If your device appears, click on the dedicated button to connect (see Fig. 6.2a). If the connection is successful, the WULPUS probe works.
7. Disconnect from the probe, scan again, find the probe, and click on the probe's name (without connecting). You should see the details of the device (see Fig. 6.2b).
8. Click on the **Clone button** (see Fig. 6.2b) to clone the advertising packet. We will need it to debug the Dongle.

6.1.2 USB Dongle

At the second step, the USB Dongle is tested. Follow the steps below:

1. After debugging the WULPUS probe, in the nRF App switch to the Advertiser Window.



(a) nRF Connect scan for BLE devices

(b) Advertising packet of WULPUS

(c) Advertising configuration

Figure 6.2: Troubleshooting with nRF Connect app.

2. You should see a configuration similar to the one in figure 6.2c: P.S. You can rename this configuration (WULPUS_PROBE_3 -> WULPUS_PROBE Test) for convenience. If you don't have such a configuration, click on + and manually create it by adding all the fields (e.g. Device Type, UUIDs, etc). Change the **Complete Local Name** to your probe's name by clicking on the pen symbol in the top right area of the screen.

   > **Note:** Don't forget to encode the correct id **X** of the WULPUS probe (i.e. **WULPUS_PROBE_X**) in your **Complete Local Name**!

3. Plug the USB Dongle into a PC.
4. Enable advertiser configuration in the app by tapping the grey icon (in red box) on the top right of the Advertiser window (see Fig. 6.2c).
5. If the green Dongle LED lights up, the Dongle is working.
6. If the LED does not light up, check if you can find **a corresponding COM port** in the Windows Device Manager.

> **Note:** For this, unplug the dongle and check the list of COM ports. Then plug the dongle in and check if a new COM port appeared.

7. If you can't find the device or something else is not working, reflash the Dongle firmware.

   > **Note:** Don't forget to change WULPUS name by inserting the correct number in the firmware! For example, **WULPUS_PROBE_X** where **X=3** is the probe's id.

8. If you are here and nothing above worked, take a new fresh Dongle and flash it.

6.2 Hardware

6.2.1 Hardware Debug

If the WULPUS probe does not appear in the nRF Connect app, follow the steps below:

1. Open and explore the schematics.
2. Power up the probe from the USB and probe the power domains on the connector P1, (it defines the USB-Battery power source selection).
3. If power from the USB does not arrive at the connector, power the probe from the Battery connector.
4. If power from the battery does not arrive at the P1 connector, power the probe from the lab supply through the middle pin of the connector.
5. Make sure all the power domains for nRF52 and MSP430 are powered up with the correct voltage levels.
6. Repeat the BLE tests.
7. If none of the above helps, reflash the firmware of the nRF and MSP.
8. If you are here, take another WULPUS probe.

Chapter 7

Errata

This chapter includes all the hardware, firmware and software bugs found *so far*.

If you find any bug not mentioned here, do not hesitate to contact us! You can find contact details on the title page or in the README of the repository.

7.1 Hardware Bugs

7.1.1 *EH-01* Current Measurement Resistor Sizing

HW Versions: $\leq$ **v1.1.0**

Problem

In certain cases, some heavier power spikes can cause some resistors to burn up, rendering the probe unusable. Resistors at the top of the power tree (**R1** at `VUSB`, **R5** at `VBAT`, **R6** and **R9** before the voltage conversion `VSYS_IN`) are particularly susceptible.

Cause

All current measurement resistors on the acquisition PCB (R1, R5, R6, R8, R9, R10, R11, R13) have been sized to **1 Ohm 1/8W**.

Solution

We recommend replacing at least **R1 and R5** with **0.1Ohm 1/4W** alternatives such as **ERJ-2BWFR100X** by Panasonic Electronic Components.

7.2 Firmware Bugs

-

7.3 Software Bugs

-

Chapter 8

Revision history

Date	Revision	Changes
18-Feb-2024	1	- Initial release.
20-Dec-2024	2	- Added instructions for producing silicone rubber package (Sect. 2.6). - Clarified the process of connecting a custom transducer (Sect. 3.1). - Fixed MSP and nRF programming instructions (Sect. 2.4,2.5 and 2.3). - Fixed incorrect DC-DC turn on time (Sect. 3.4, 4.2 and 5.1). - Clarified channel numeration (Sec. 4.1.2 and 5.1). - Added Errata (Sect. 7).

Bibliography

[1] S. Frey, S. Vostrikov, L. Benini, and A. Cossettini, "Wulpus: A wearable ultra low-power ultrasound probe for multi-day monitoring of carotid artery and muscle activity," in *2022 IEEE International Ultrasonics Symposium (IUS)*. IEEE, 2022, pp. 1–4.

[2] S. Vostrikov, M. Anderegg, C. Leitner, L. Benini, and A. Cossettini, "Hand gesture recognition via wearable ultra-low power ultrasound and gradient-boosted tree classifiers," in *2023 IEEE International Ultrasonics Symposium (IUS)*. IEEE, 2023, pp. 1–4.

[3] S. Vostrikov, M. Anderegg, L. Benini, and A. Cossettini, "Unsupervised feature extraction from raw data for gesture recognition with wearable ultra low-power ultrasound," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2024.

[4] G. Spacone, S. Vostrikov, V. Kartsch, S. Benatti, L. Benini, and A. Cossettini, "Tracking of wrist and hand kinematics with ultra low power wearable a-mode ultrasound," *IEEE Transactions on Biomedical Circuits and Systems*, 2024.

[5] S. Vostrikov, L. Benini, and A. Cossettini, "Complete cardiorespiratory monitoring via wearable ultra low power ultrasound," in *2023 IEEE International Ultrasonics Symposium (IUS)*. IEEE, 2023, pp. 1–4.

[6] N. Semiconductor, "nrf52840 dongle," https://www.nordicsemi.com/Products/Development-hardware/nRF52840-Dongle/GetStarted, accessed: 2024-02-18.

[7] S. Frey, V. Kartsch, C. Leitner, A. Cossettini, S. Vostrikov, S. Benatti, and L. Benini, "A wearable ultra-low-power semg-triggered ultrasound system for long-term muscle activity monitoring," in *2023 IEEE International Ultrasonics Symposium (IUS)*. IEEE, 2023, pp. 1–4.