Dual Accelerometer Vector Sensor

User Manual

EMSO-PT 22157 (01/SAICT/2016)

Document Information

Work requested by	CINTAL Universidade do Algarve, FCT - Campus de Gambelas 8005-139 Faro, Portugal Tel: +351-289864258 cintal@ualg.pt www.cintal.ualg.pt
Laboratory performing the work	Signal processing Laboratorry (SiPLAB) Universidade do Algarve, FCT - Campus de Gambelas 8005-139 Faro, Portugal Tel: +351-289800949 info@siplab.fct.ualg.pt www.siplab.fct.ualg.pt/
Project	EMSO
Document Title	DAVS User Manual
Authors	N. Pinto, S. Jesus
Authors Contact	nfpinto@ualg.pt ; sjesus@ualg.pt
Date	March 11th 2020
Reference	CINTAL 2/20
Number of Pages	81 (eighty one)
Abstract	This document consists of a user manual for the DAVS equipment. It describes how to prepare an experiment, deploy and retrieve recorded data from DAVS.
Clearance level	UNCLASSIFIED
Distribution List	SiPLAB (1), CINTAL (1)
Total Number of Copies	2 (two)

Document revisions

Revision	Date	Description	Reference
01	20/01/2020	First draft version	
02	29/01/2020	Changed chapters order Some images and figures redone	
03	10/03/2020	Changed chapters structure New chapters introduced	

Content

List of Figures

List of Tables

Chapter 1 Introduction

The Dual Accelerometer Vector Sensor (DAVS) is a measuring equipment developed during WiMUST[1] project to perform geophysical acoustic surveys at sea. The device is a vector sensor for underwater sensing applications, in a compact size capable of being mounted on an autonomous underwater vehicle (AUV). This small size allows for an easy integration with AUVs and facilitates the vehicle manoeuvrability, compared with a classic towed array [1].

Figure 1 - DAVS system

The DAVS acoustic sensor consists of two closely spaced tri-axial accelerometers one on each side of a hydrophone, all in the same casing and aligned along an axis. These sensors work as a vector sensor and can obtain information about the directivity of the acoustic signals. Compared to traditional sensor arrays, the DAVS sensor configuration is able to easily distinguish sea-bottom reflections from direct and surface reflected paths, allowing for a simpler data processing than that needed for a towed array [1]. It´s acoustic sensors are molded together to form a solid and compact directional array. Additionally, it has inertial sensors, which allows to obtain information about the device orientation. For the cabled mode will be used the DAVS Base Station system as a telemetry and powering external system.

The DAVS system is a complete acquisition system, with 7 recording channels corresponding to the acoustic sensors (accelerometer #1 X axis, X axis, Z axis, hydrophone, accelerometer #2 X axis, Y axis, Z axis). The recorded data from that 7 channels is stored into a WAVE format file, without any compression.

Any deployment follows a workflow with 3 main phases, covered in the following chapters:

1) Preparation and configuration of the device;
2) The underwater deployment or the experiment;
3) Retrieving and processing of the recorded data.

The DAVS can operate in two different modes: an autonomous mode and a cabled mode. The autonomous mode allows for a standalone deployment where acquisition, data storage and powering are internal to the DAVS. The cabled mode allows the acquired data to be visualized and stored by any external host device, the powering of the system and the adjustment of some DAVS settings on the fly. In this manual we will use the external DAVS Base Station system, developed by Marsensing. Each mode has different preparation methods and running details, so for each phase of the workflow there's also a distinction between them.

The manual is structured in the following way:

- 2nd chapter talks about the basic preparations before deployment with DAVS. It´s explains what you need to do and prepare before leave to deployment site.
- 3rd chapter shows how to proceed in the deployment site and some care to be taken when running the experiment.
- 4th chapter details the after deployment, namely the data retrieve process and some simple data processing examples, using two common software's (**Audacity** and **Matlab**). It has also some information about the long-term storage and maintenance operations.

[1] WiMUST project, funded under H2020-EU.2.1.1.5, agreement ID:645141

- The appendix section, where some additional information's not needed for deployment are done. There is a more detailed description of the devices, the **Matlab** source code, some basic information about the Fast Fourier Transform algorithm and the structure of some files generated by DAVS.

This manual serves as a reference for DAVS usage from a user perspective, containing only general information about the system. It will explain some of the hardware without getting into much detail, for anyone that will need to work with DAVS understand what each component is. More technical information's are described in the device hardware report [2].

Chapter 2 DAVS Operation: Before Deployment

This chapter will focus on the preparation of the device, for any of the deployment modes. It contains all the important information to plan and prepare an underwater deployment.

2.1 DAVS system introduction

DAVS system is a complete acquisition platform that record data from acoustic sensors plus some meta data information about the device status. It´s three main parts are the end cap that close the, the main cylinder that contains all the electronic boards inside and the nose where the acoustic sensing elements are molded. These are identified in Figure 2.

The end cap is a simple cap made with Delrin, which seals the main cylinders and contain an underwater connector that allows a cable connection. The main cylinder is also made from Delrin and contains all the electronic systems (electronic control boards and battery packs for autonomous mode) of the system. DAVS nose is where the sensing elements are molded and is attached to a Delrin cap that screw to main cylinder. All the electronic boards are attached to nose. Inside DAVS nose there are two accelerometers placed between a hydrophone. Each accelerometer is designed by a number (#49 or #50), referring to accelerometer last two digits of serial number. This identification is useful to know the relative position of accelerometers into DAVS nose, since DAVS is a vector sensor where the orientation is important.

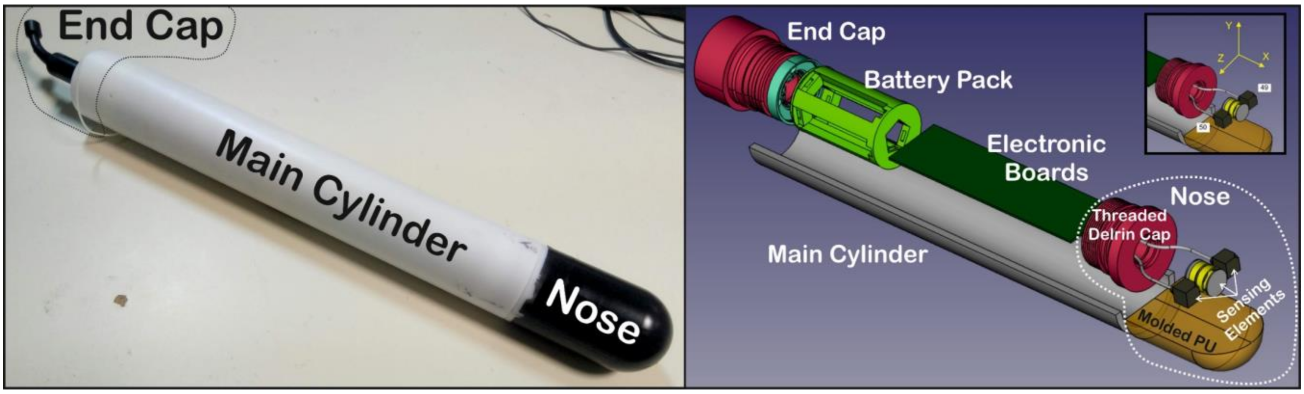


Figure 2 - DAVS main components

The two accelerometers and the hydrophone generate 7 channels of acoustic data which depending on the deployment mode can be streamed (autonomous mode) or stored into an internal SD card (autonomous mode). When streamed to DAVS base station, this device will store the acquired data internally. In any of the two cases, the acoustic data is stored into a WAVE file of 8 channels. Table 1 resume the corresponding WAVE files channels to DAVS sensing element.

Transducer	Accelerometer #49			Hydrophone	Accelerometer #50			NOT USED
	X axis	*Y axis*	*Z axis*		*X axis*	*Y axis*	*Z axis*	
WAVE channel	1	2	3	4	5	6	7	8

Table 1 -WAVE file channels

The DAVS has also an inertial measurement unit which generate roll, pitch and heading information. These are stored in two ways, depending on the deployment mode. When using in autonomous mode these inertial data is stored into WAVE file header, while when in cabled mode this data is merged with some other system information and stored into a comma separated values (CSV) file. During Autonomous mode, additional information about the system is stored into WAVE file header. Figure 3 resumes the stored data, where the acoustic sensor is always recorded to WAVE file, while system and inertial information can be stored into WAVE file header (autonomous mode) or into a separate CSV file (cabled mode with base station).

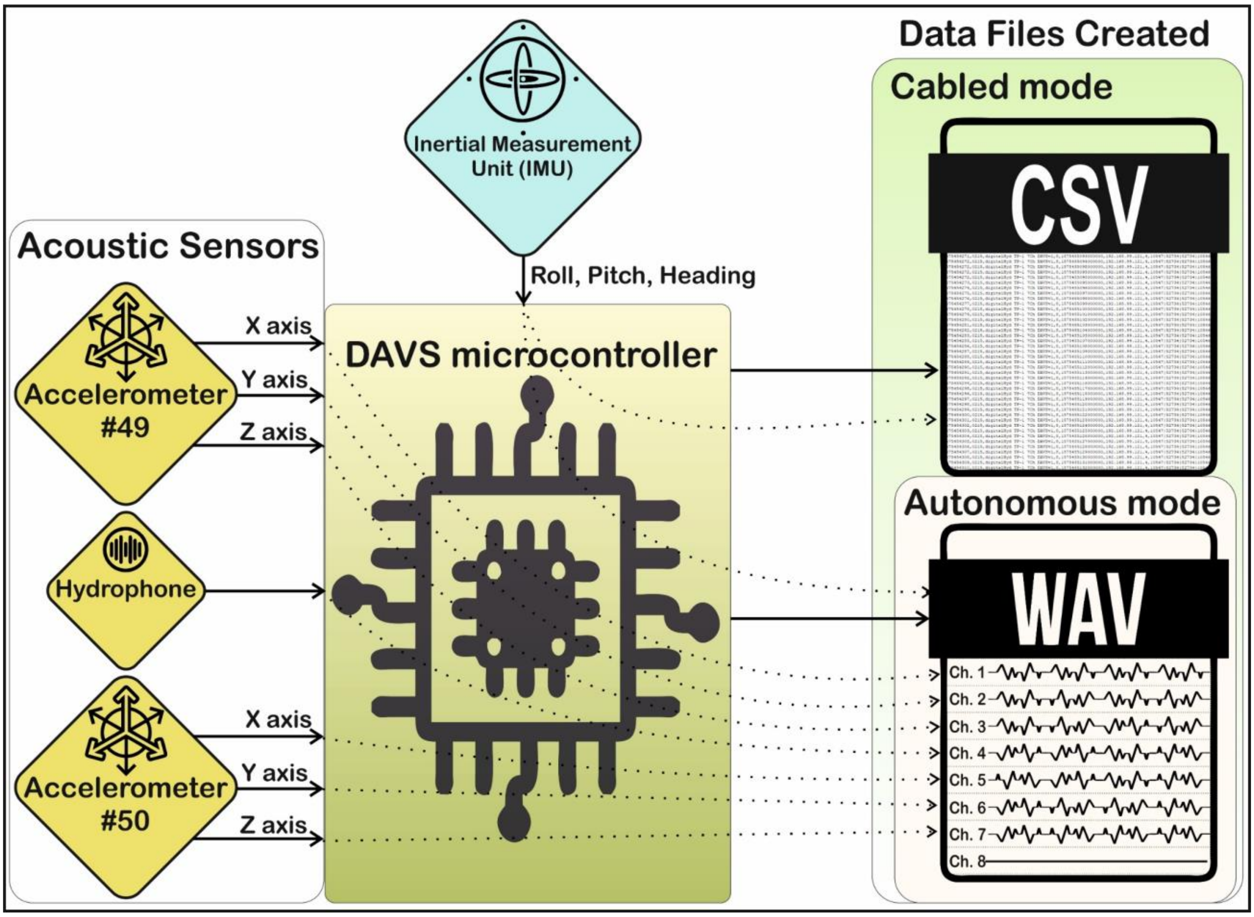


Figure 3 - Recorded data path and files created based on selected mode

In autonomous mode the created WAVE files are stored into a micro SD card, while when in cabled mode with DAVS base station, the WAVE and CSV files can be accessed and downloaded via web browser.

2.2 DAVS Storage and Transportation case

The DAVS system is stored inside a waterproof Peli Case (model 1605), which can also include several accessories as the batteries, battery packs and chargers for batteries (Figure 4). Depending on the

Figure 4 - DAVS Peli Case (left side image) and some optional accessories inside case (right side)

deploy mode, this case can carry all the needed material. Before leaving for deployment always check the case content for your deployment needs.

2.3 Autonomous mode

The DAVS autonomous mode selection is based on the detection of a SD card in slot A, containing a valid configuration file. If, when powering up the device, there's a SD card in slot A with a configuration file, the device will boot for autonomous mode. If no card is found, then DAVS will assume the cabled mode and begin streaming the data through Ethernet. When powered ON, DAVS will start by searching for the configuration file and then configures all the electronics accordingly.

2.3.1 SD card preparation

Prepare the micro SD card timely, formatting it in exFAT or FAT32 format. There are no restrictions on volume name. It's not necessary to format the card every time you use DAVS but pay attention to free space and the estimated deploy time. DAVS have 3 different sample rates available (10547, 52734 and 105469) which can be chosen, depending on the experiment needs. Even if it only uses 7 channels, the device records the 8 analog outputs of analog to digital converter (ADC). You can estimate the file size for some acquisition, based on the recording time $t_{recording}$ (seconds) and analog to digital converter (ADC) sample rate SR (Hz) setting, using the following formula (1) in MegaBytes:

$$Size_{estimation} = \frac{SR * 24 * 8 * t_{recording}}{8 * 2^{20}} \; [MegaBytes] \qquad (1)$$

Note that since each WAVE file created has a header information which occupy some extra bytes (512), this value is only an approximation. You can also estimate the recording time $t_{recording}$ (seconds) allowed by a SD card of size SD_{size} (GigaBytes):

$$t_{recording} = \frac{8 * 2^{30} * SD_{size}}{SR * 24 * 8} \; [seconds] \qquad (2)$$

For example, when sampling at 10547 Hz, you will be storing 10547 points of 24 bits for each channel per second. So, the eight channels will need 2025024 (10547*24*8) bits per second of recorded data. If you store 60 seconds of data, it will result in 121501440 bits (2025024*60) or 15187680 bytes (121501440/8). This is near 15 MB per minute of record. Using a sample rate of 52734 will result in file with near 73 MB per minute. Each free GB will give you approximately 4241 seconds (70 minutes) at a sample rate of 10547 Hz or 848 seconds (14 minutes) at 52734 Hz, so always consider the expected deployment time, sample rate chosen and free card space. Also remember that higher sample rates (as 105469 Hz) will require a card with a high write speed, namely an U3 class card. Some slow speed classes may result in data losses or card corruption.

After formatting, create a configuration file named "*CONFIG.TXT*" and place it in the root of SD card. The settings you choose will maintain until you change any parameter in "*CONFIG.TXT*" file. Each line of this text file will represent some DAVS configuration parameter:

- Line 1) Length of the recorded WAVE file, in seconds. The minimum value is 60 seconds. To keep files at a reasonable size, they are stored with a fixed length, usually a few minutes (1, 2 or 3). Processing large files can be hard, so it's a good practice to create individual files with only a few dozen MB. For example, for a record file with 60 seconds of the seven sensors data, this line should contain the number "*60*" (without quotes);
- Line 2) Hydrophone Programmable Gain Amplifier (PGA) index, from 1 to 7 representing the gains from 1x to 64x (see Table 2). These gains are explained in appendix A.1.4 ;
- Line 3) Accelerometers PGA Gain Index, from 1 to 7 representing the gains from 1x to 64x (Table 2);

Line 4) Sampling rate index, from 0 to 3 corresponding to DAVS sampling rate (see Table 3). Check appendix A.1.4 for a detailed explanation.

PGA Gain	**1x**	**2x**	**4x**	**8x**	**16x**	**32x**	**64x**
PGA File Index	1	2	3	4	5	6	7

Table 2 - PGA gain index for configuration file (for line 2 and 3)

Sampling Rate (Hz)	**10547**	**52734**	**52734**	**105469**
Sampling File Index	0	1	2	3

Table 3 - Sampling rate index for configuration file (for line 4)

2.3.2 SD Card preparation example (Windows OS)

In Windows environment, insert the SD card into computer using any micro SD adapter. By selecting the card in windows explorer, right click it and choose format as in (1) of Figure 5. Then select FAT32 file system, write a name and click 'begin' to start and 'OK' to confirm (2). After formatting card finished (3), open it and create a text file (extension ".txt") with notepad or similar (4). Write the desired settings in each line, saving the file as "*CONFIG.TXT*" into SD card root folder (5). In the example above, a configuration file was created with files of 60 recording seconds, with PGA gains of 1 for hydrophone and accelerometers and an ADC sample rate of 52734 Hz.

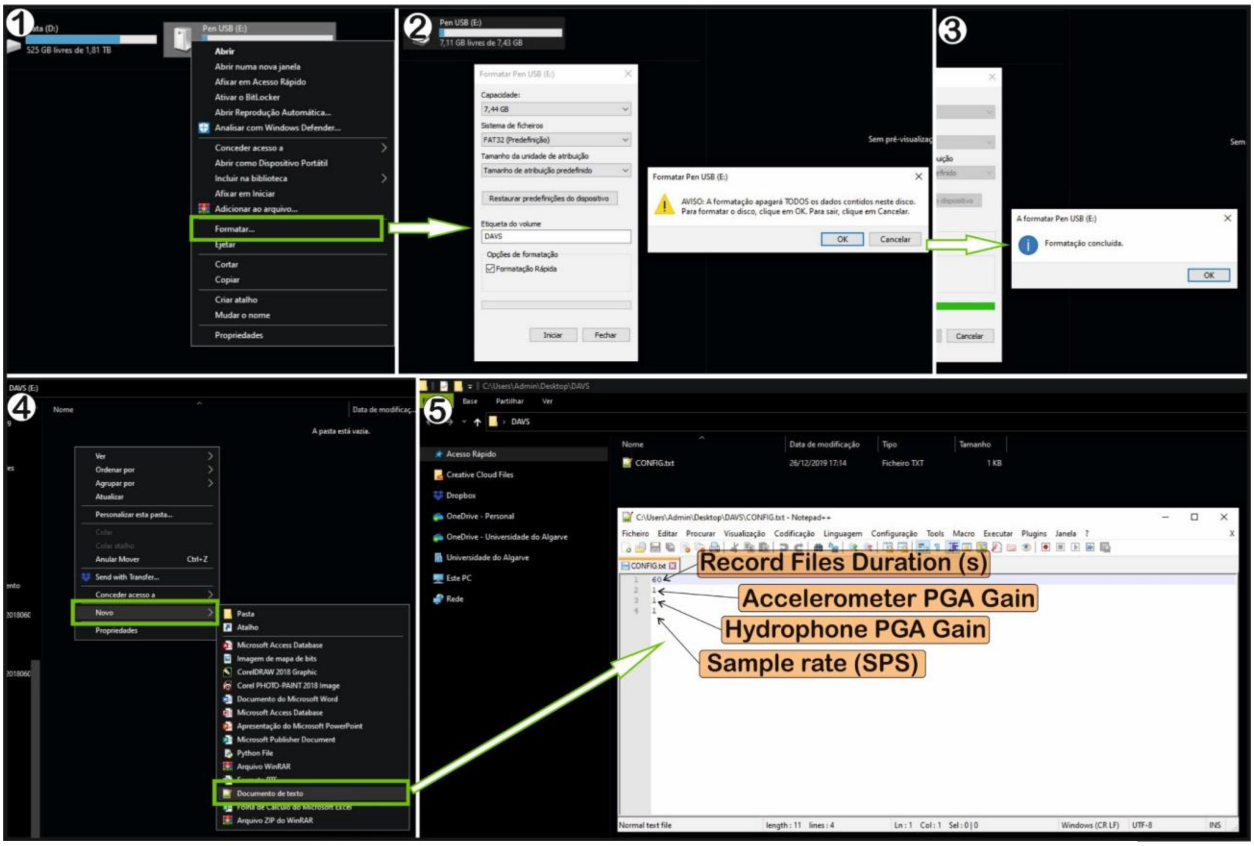


Figure 5 - Example of DAVS configuration file in Windows and Notepad++

2.3.3 18650 Li-Ion Batteries preparation

When deploying in autonomous mode, DAVS use five rechargeable Li-Ion 18650 batteries as power source. These batteries should be charged in an appropriate charger for Li-Ion, which manages the charging process. When picking the batteries, it´s possible to check for the charge state of a battery using a multimeter. Just check if the battery presents a voltage around 4.1 Volt as in Figure 6. A good charger will also have the option to check the charge state of a battery. If any battery has less than that voltage, probably isn't fully charged.

Note that a discharged battery can take several hours to recharge, depending on the charger and charging current, so prepare them in advance. Always choose quality branded batteries with a high rated capacity (> 3000 mAh) for longer runtimes.

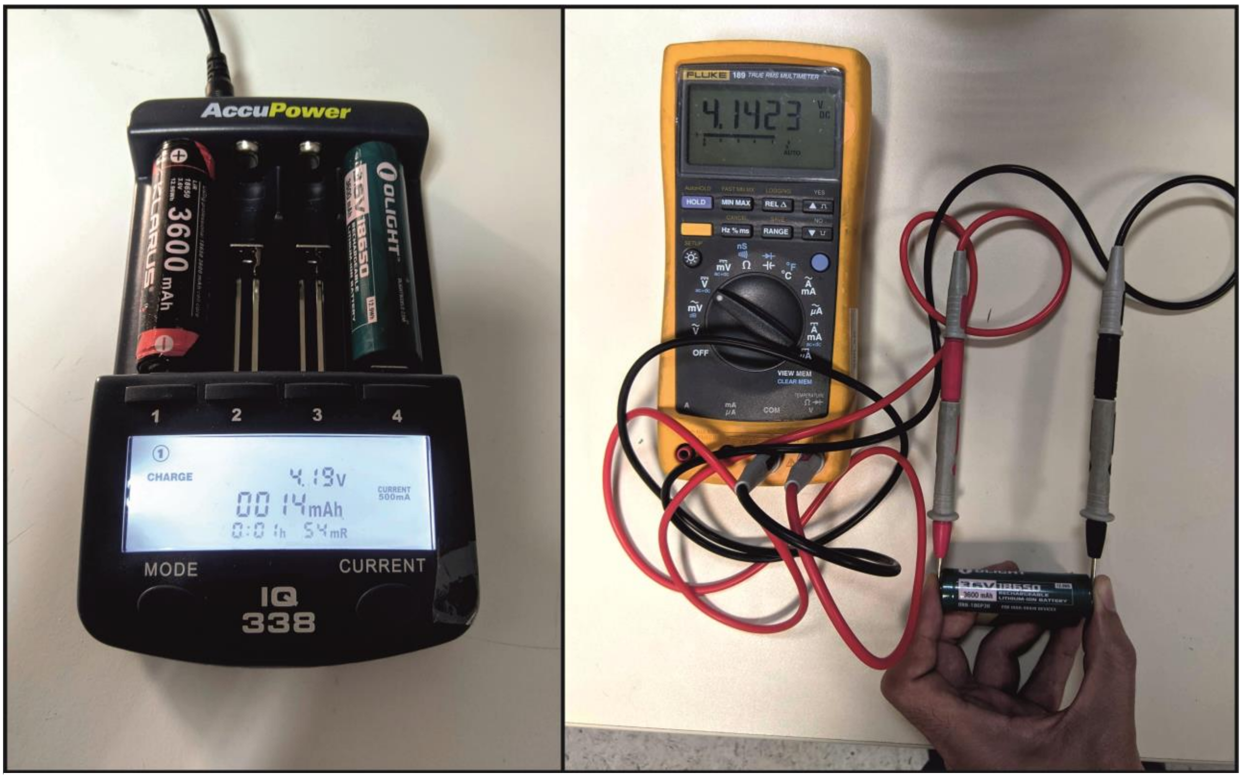


Figure 6 - LiIon batteries charger (left) and LiIon voltage measure with a multimeter (right)

2.3.4 Accessing DAVS Battery compartment

Open the DAVS battery compartment, removing the locking wire and pulling the cap out gently (check Figure 7). Take care of the Ethernet and XT30 power cables inside DAVS which could be connected to DAVS mainboard. XT30 could be manually unplugged, but Ethernet need a screwdriver (or anything with a long rod) to help unplugging it from the yellow main board support.

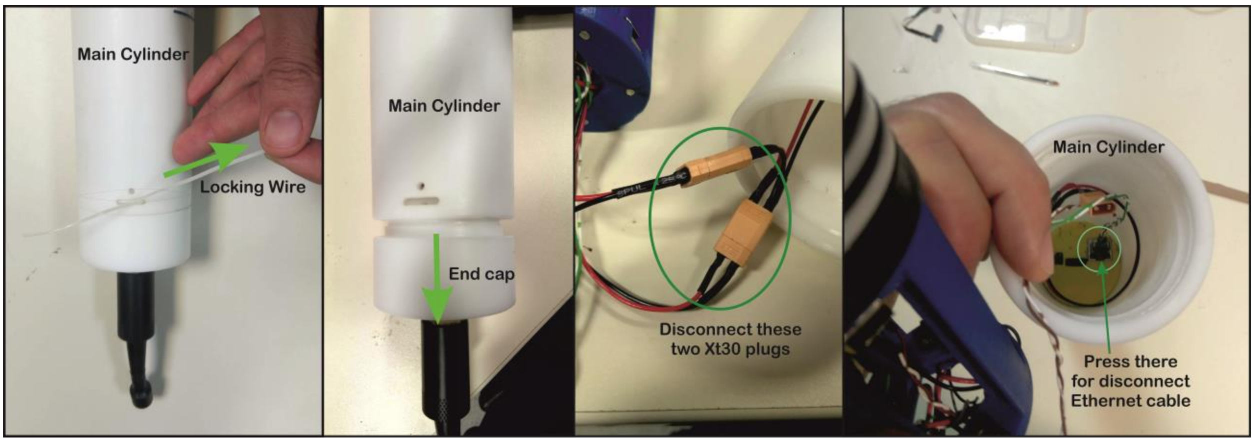


Figure 7 - Removing locking wire, end cap, XT30 connectors and Ethernet connector from main cylinder

2.3.5 Accessing DAVS SD card slots

If you need to insert a SD card, remove the DAVS nose by unscrewing it gently. Try to press the nose in the white Delrin zone only (marked in dashed red at Figure 8 above). Only unscrew the nose from the cylinder when the SD card is prepared, to minimize the exposure time to any dust or particles that may damage the sealing o-rings.

Figure 8 - Remove nose from main cylinder

All electronic boards are attached to the nose, so remove it from cylinder carefully (1 in Figure 9). The SD slots are in the plastic yellow support and you need to use on slot A a card with a configuration file. Can also be used slot B for extending memory capacity, with a valid card on slot A(2). After inserting the SD card immediately close the DAVS, reversing the process of Figure 8. Slide the cylinder through the electronic boards carefully (3), then screw it in the direction marked by (4) and until it aligns the cylinder X axis mark with the nose X mark (5). The X mark follows the mold marks, as the dashed green alignment mark shows.

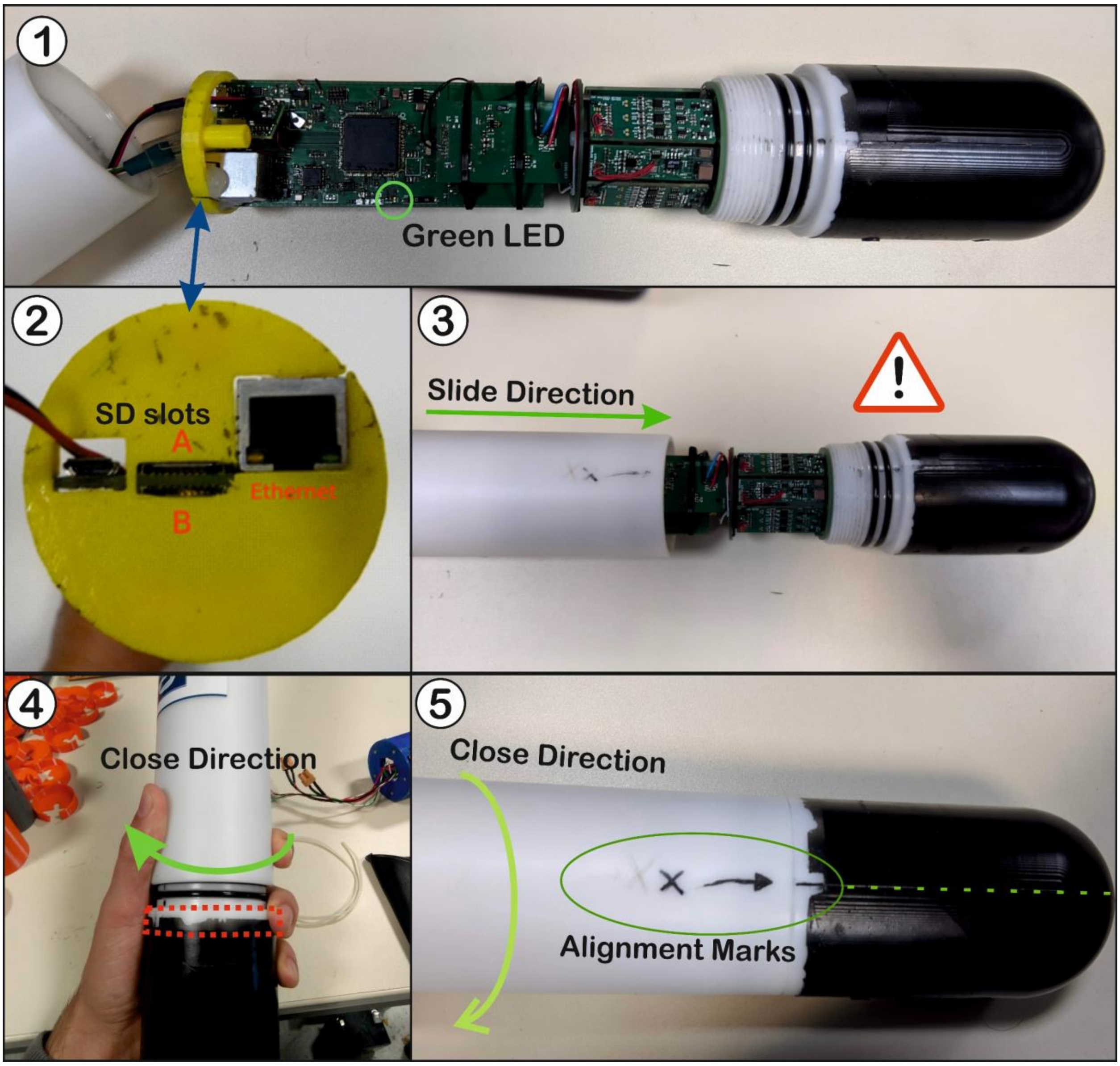


Figure 9 - Opened DAVS (1) with marked yellow support location and SD card slots (left side) and X axis alignment marks (2). Closing procedures (3 and 4) and aligned position (5)

2.3.6 Attaching Battery pack to DAVS end cap

DAVS Battery pack is a blue cylinder which use five Li-Ion 18650 batteries and can be screwed to DAVS end cap. If the battery pack was disassembled from end cap (Figure 10), it's necessary to pass the energy

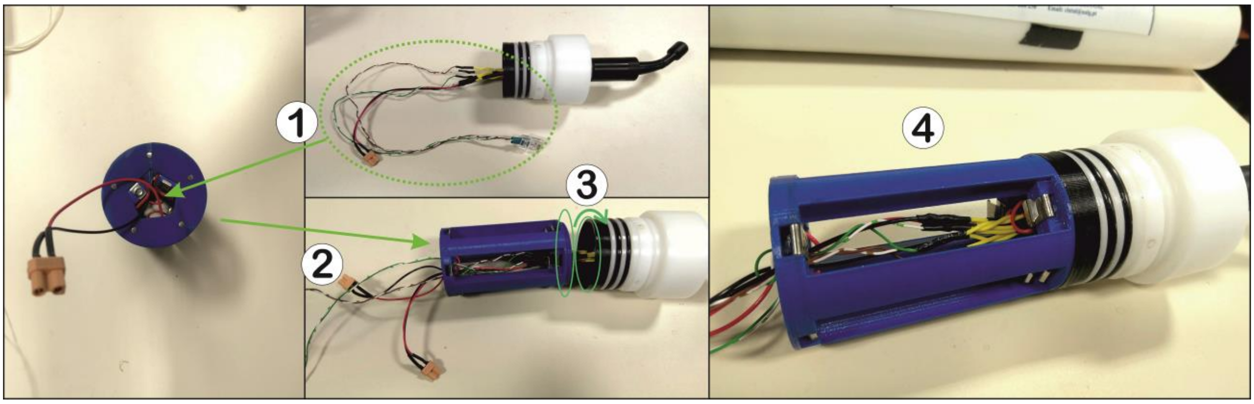


Figure 10 - Mounting of battery pack in End Cap

and communication cables through the centre of the battery pack (1) before inserting the batteries. With some care, pass the cables as shown in Figure 10 (2) paying attention to the pack threaded end that should be in the end cap side. Then screw the battery pack to the black part of end cap and just slightly tighten it (3). It's ready to insert the 18650 batteries (4).

2.3.7 Installing DAVS Batteries

Before installing the batteries, make sure they are recharged by checking the voltage individually. Each one should have around 4.1 Volt, as seen in chapter 2.3.3. A discharged pack will reduce the device autonomy. Don´t mix charged and discharged batteries in the same pack.

Insert five batteries in the pack, respecting the polarity as in Figure 11. The polarity isn't marked in the pack, so note the negative side as a metal spring plate. The battery pack voltage should also be checked with a multimeter, to guarantee that the voltage is near 20.5 Volt. An autonomy of 20 hours is expected when using five 18650 batteries with 3100 mAh of capacity.

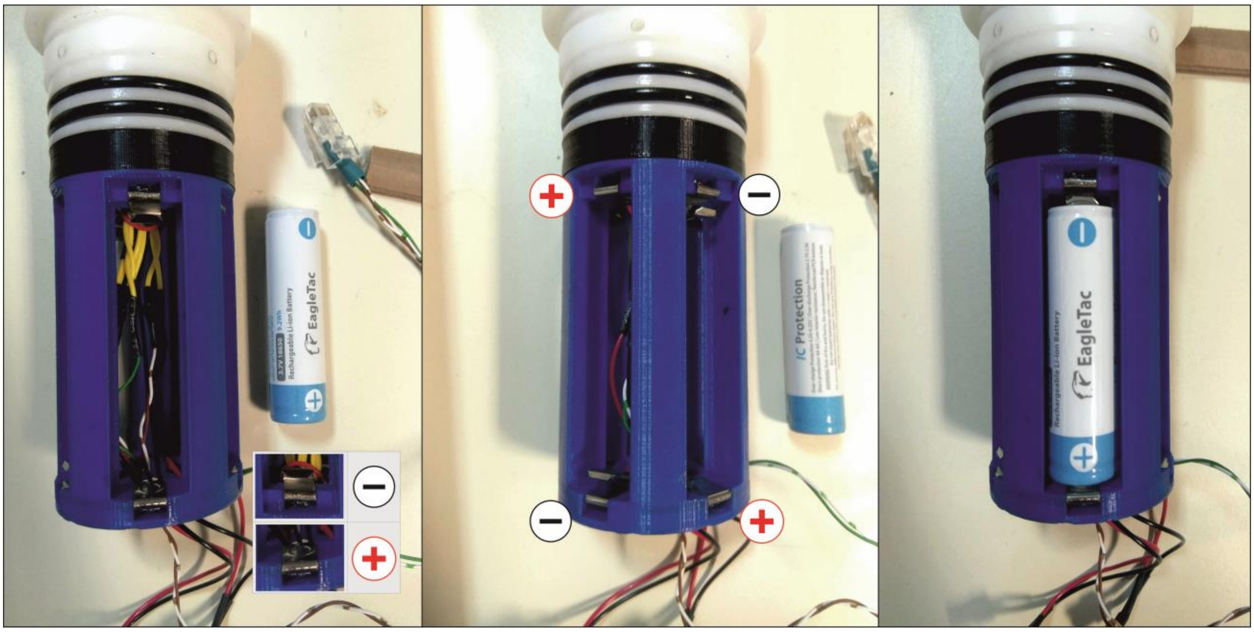


Figure 11 - Batteries position on battery pack

2.3.8 Closing DAVS system

Connect the two female XT30 energy connectors from the end cap and battery pack, to the two male connectors of the electronics (Figure 12). The Ethernet cable can be left unplugged. The DAVS system will start up immediately, trying to read the SD card. To see if there's no error reading SD card, check for LED light which should be blinking with 10 seconds duration (10 second ON then 10 seconds OFF). If blink duration is near 1 second, then there's some error reading SD file which should be verified. Please note

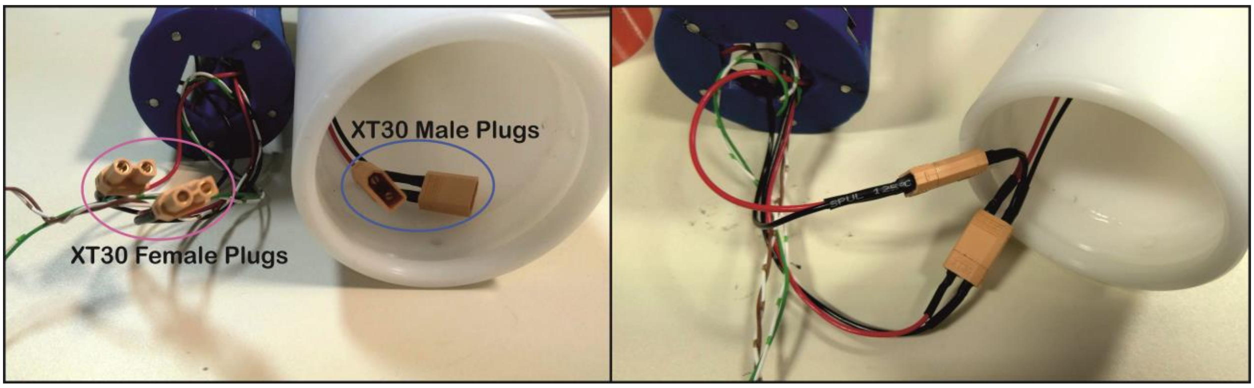


Figure 12 - XT30 power plugs connection

that a 10 second blink just indicate that DAVS is working. If there's any problem with configuration file, DAVS will not be recording but it doesn't has any LED indication for that.

Assemble the DAVS by inserting the cap with care in the cylinder. Just slide it in the direction shown at Figure 13 (1) and press a little in the end, there's no fixed position for this cap. Put the locking plastic wire, by rolling it (3) through the opening of the image (2). After closing it, the two ends of the locking wire will be sticking out, and the system is ready for deployment (4).

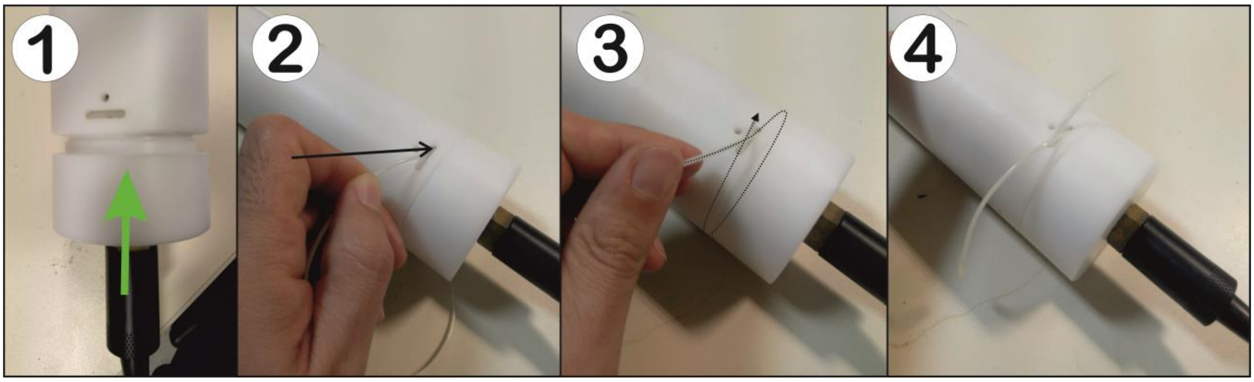


Figure 13 - Closing the end cap

2.3.9 Autonomous deployment material check list

Before leaving for deployment site check if you carry the basic material for autonomous operation. The following Table 4 resumes the more useful material to take.

Material	Notes
DAVS Peli case	
18650 batteries	5 units
18650 spare batteries	Depending on deployment time
18650 charger	
Zip Ties	Large size
Duct Tape	Large size
Fixation Supports	If there's any for the experiment
Cleaning cloth	To dry DAVS after retrieving it from the water

Table 4 - Autonomous mode material check list

2.4 Cabled Mode:

When using DAVS on cabled mode an external device should be used to provide energy and receive the acquired data. This connection is done through an Ethernet cable prepared for underwater use, allowing the real-time streaming of sensors data as well as powering the device. An external host computer must be used for view and retrieve streamed data as well as set acquisition parameters. Here we will use DAVS Base Station system described in appendix A.2, which already has the underwater cable and act as a Wi-Fi hot spot allowing the connection of any external host computer. Real time streamed data will be stored in this device memory. DAVS setting can be adjusted through a web interface of DAVS Base Station.

2.4.1 DAVS base station storage case

DAVS Base Station (Figure 14) is a companion system of DAVS developed by Marsensing, allowing an easy visualization of acquired signals and an interface to change acquisition settings on the fly. All the system components are inside a protective Peli case model 1200. It includes an underwater cable for direct connection to DAVS. When the system is working, it will act as a Wi-Fi hot spot allowing the connections of external devices and giving access to a web server. This web server will allow the visualization of streaming data and the download of recorded files, through a web interface. It's also possible to configure DAVS, adjusting some settings through this web interface.

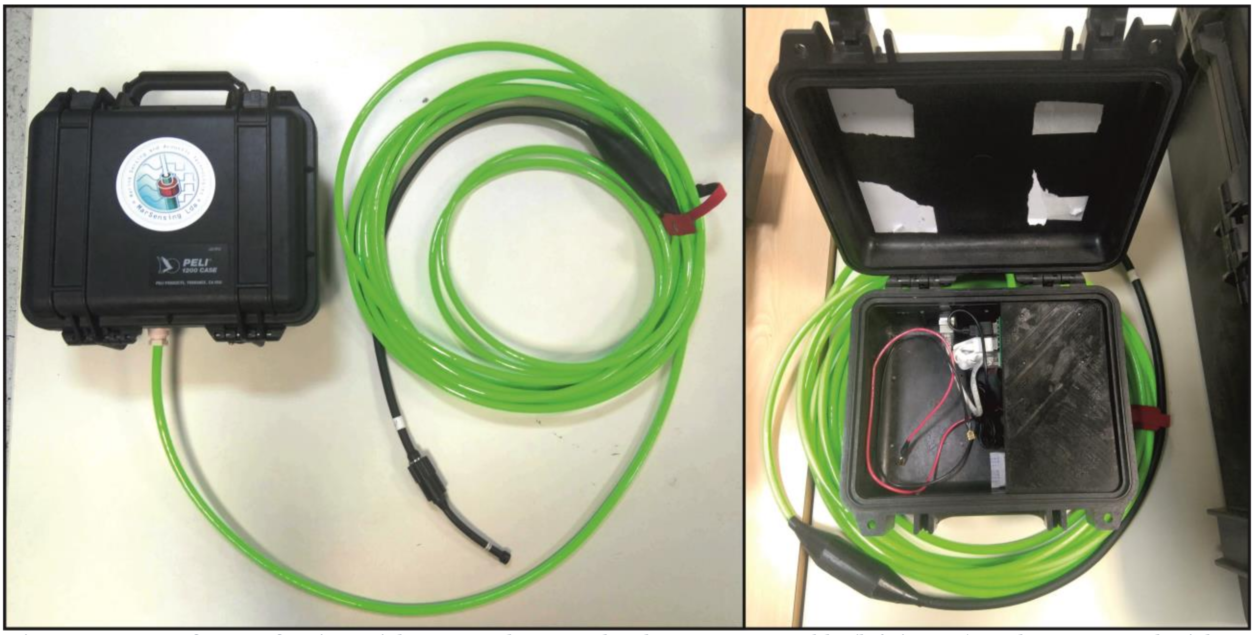


Figure 14 - DAVS Base Station with case and external Ethernet green cable (left image) and case opened without battery (right image)

2.4.2 Lead/Acid Batteries preparation

DAVS Base Station is powered by a 12 Volt DC lead acid battery type (Valve Regulated Lead Acid type, model UL9-12), with 2 male spade type connectors and a capacity of 9 Ah (Figure 15). It needs to be charged with a specific charger for lead acid batteries and can take several hours to recharge. So, prepare the batteries some days before the deployment day. For recharge just connect the charger cables to battery, respecting polarities (Figure 16). Depending on the charger, can be seen the state of charge. The charging times also depend on the charger current. The voltage can be measured with an multimeter and should be higher than 12 Volt.

Always prepare some spare batteries, depending on how many days are planned for the experiment.

Figure 15 - 12 Volt Lead Acid (VRLA) Battery for base station with marked polarities

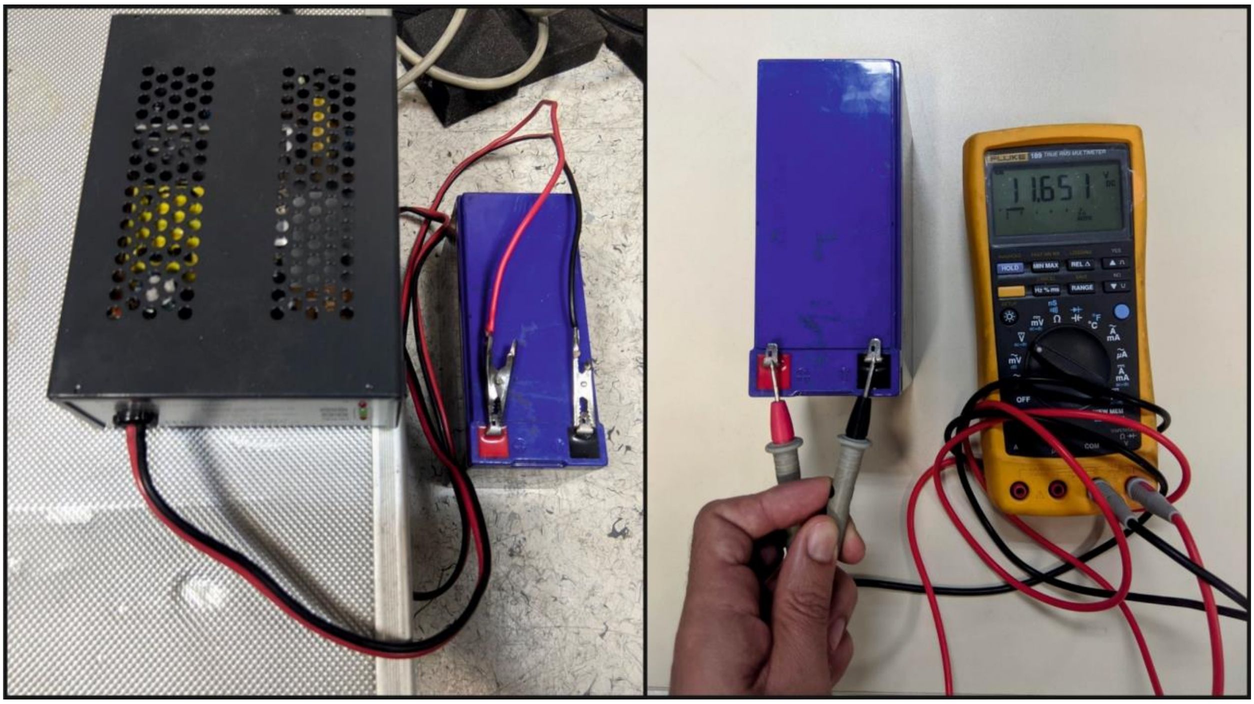


Figure 16 - Charging a lead acid battery (left side). Note the color of the charging cables respecting the battery polarity. Checking battery voltage with a multimeter (right side)

2.4.3 Accessing DAVS Batteries compartment

To access the batteries compartment, start by opening DAVS, removing the locking wire and pulling the end cap out gently as seen on Figure 17. Take attention to the Ethernet and XT30 power cables inside DAVS which could be connected to DAVS mainboard. XT30 can be disconnected by hand, but Ethernet need a screwdriver to help unplug it from the yellow main board support (check). You will need to press the Ethernet locking pin with the help of a screwdriver and pull it by the wires. Or you can disconnect it by hand if need to disassemble DAVS nose for SD card removal.

For the cabled operation mode, we need to make sure that no SD card is inserted in any DAVS slots and that Ethernet pigtail and XT30 power cable are connected to main board. If there’s any SD card inserted, it will prevent DAVS from booting in cabled mode and stream data, so we need to remove it. If none SD card is inserted and the Ethernet and energy cables are connected, you can skip the next steps to chapter 2.4.6.

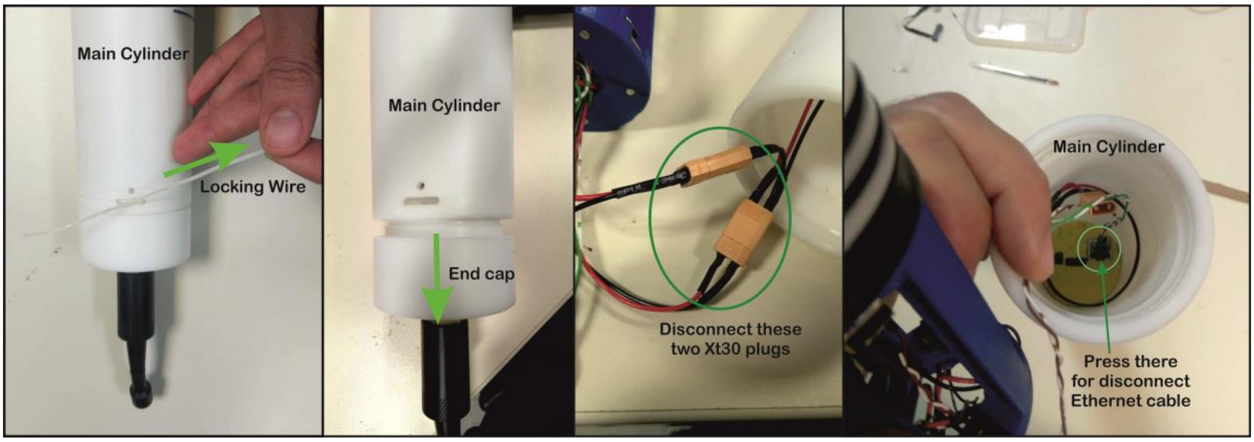


Figure 17 - Removing locking wire, end cap, XT30 connectors and Ethernet connector from main cylinder

2.4.4 Removing the battery pack from DAVS

In cabled mode it's possible to use DAVS with or without battery pack inserted. If you don't want to remove the battery pack, just make sure it hasn't any batteries inserted. If you want to remove it follow the sequence from Figure 18 and with all cables disconnected from mainboard unscrew the blue pack gently from white end cap (1) in green arrow direction, then separate it from the end cap cables with care (2). Don't force the cables, if they're stuck use your fingers to lose them.

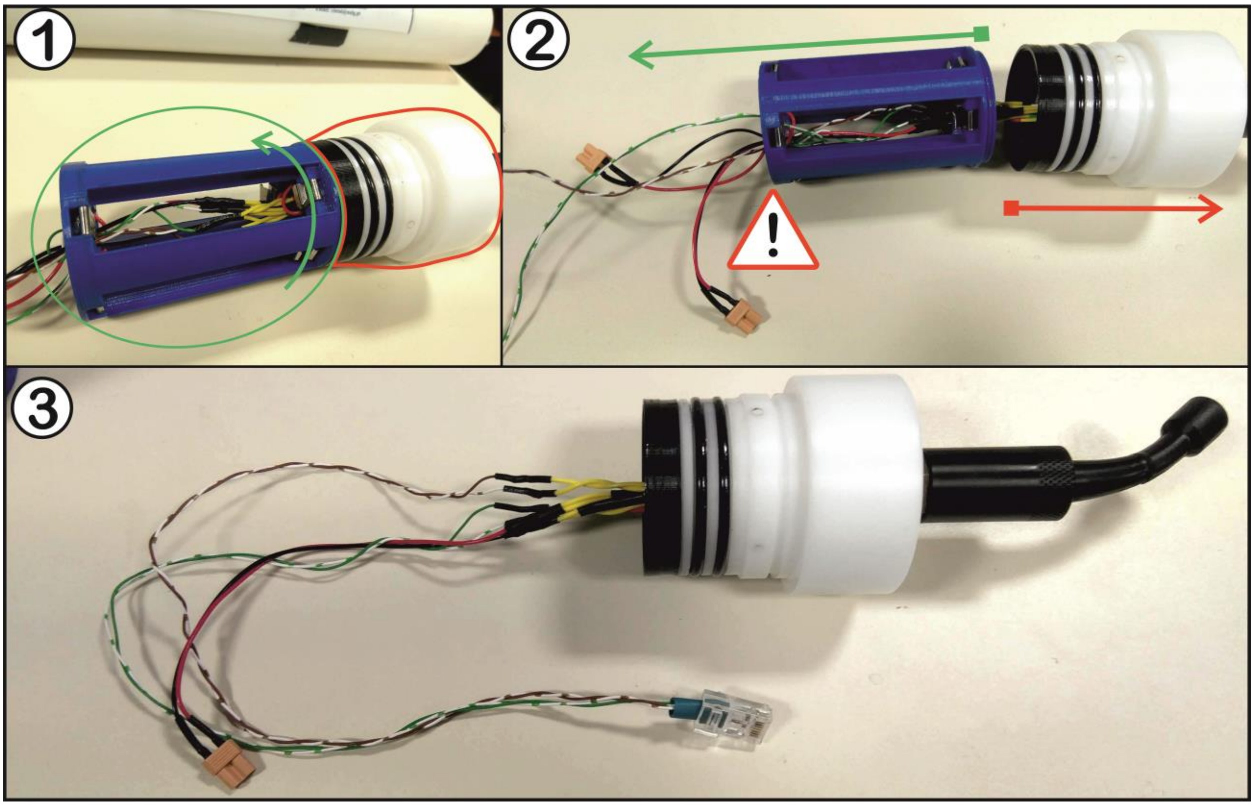


Figure 18 - Removing of battery pack from DAVS end cap

2.4.5 Removing any SD card from DAVS slots

If need to remove any SD card from mainboard you will need to disassemble DAVS nose, starting to unscrewing it gently. Try to press the nose only in the white Delrin zone (marked in dashed red at Figure 19 above) and rotate the main cylinder body in the green arrow direction. Try to minimize the exposure time to any dust or particles that may damage the sealing o-rings.

Figure 19 - Remove nose from main cylinder

All the electronics are attached to the nose, so remove it from cylinder with care. To remove a card, just pull it gently from the plastic yellow support (Figure 20 image 1). After removing the SD card immediately close the DAVS, first connecting the Ethernet cable (2) then sliding the white cylinder with care (3). Screw it in the direction marked in (4) till the cylinder and nose marks align (5). Please note that when plugging the Ethernet cable, the end cap (and battery pack, if attached) will be physically connected to the nose, so screw the cylinder with care and never force it if you find any resistance. While screwing, check if the Ethernet wires are not being twisted and rotate the end cap to prevent twisting, if necessary. Check if something is stuck and try to clear it before continuing to screw the cylinder. Rotate it till the alignment marks from cylinder and nose overlap.

2.4.6 Closing DAVS

If the Ethernet cable is disconnected try to reconnect the Ethernet plug, just by pressing it until hear the locking pin click. You will need to manually align it with the DAVS mainboard socket and then press it with the help of any long object like a screwdriver or a pen. Then reconnect the XT30 power plug and close the cylinder, sliding the end cap gently in green arrow direction of Figure 21.

To lock the end cap, press a little in the cap to take it to the correct position (1 of Figure 22). There's no fixed position for this cap. Put the locking plastic wire, by rolling it (3) through the opening of the image (2). After closing it, the two ends of the locking wire will be sticking out, and the system is ready for deployment (4).

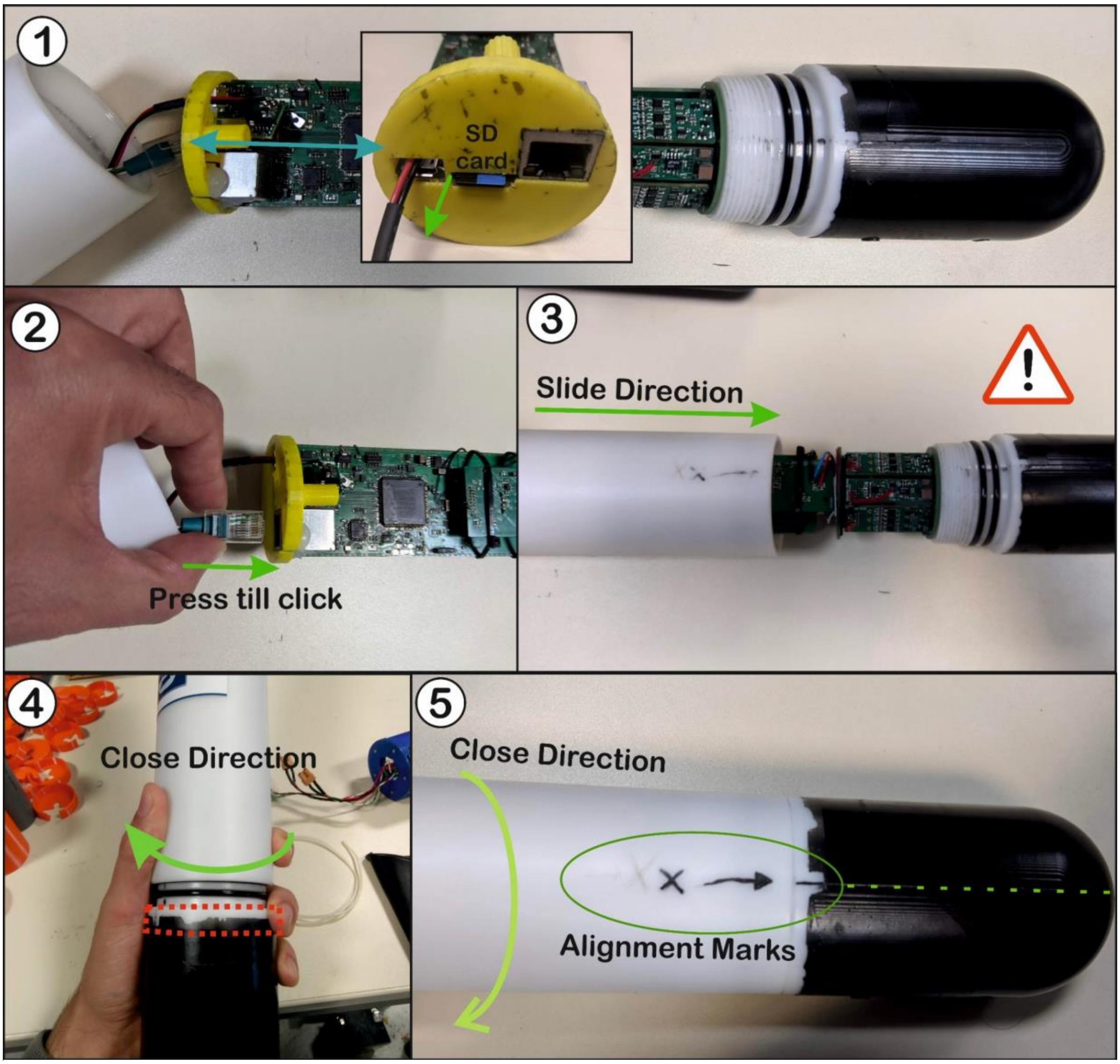


Figure 20 - DAVS SD card slots (1), Ethernet cable connection (2), closing DAVS procedure (3) and (4) and alignment of cylinder and nose (5)

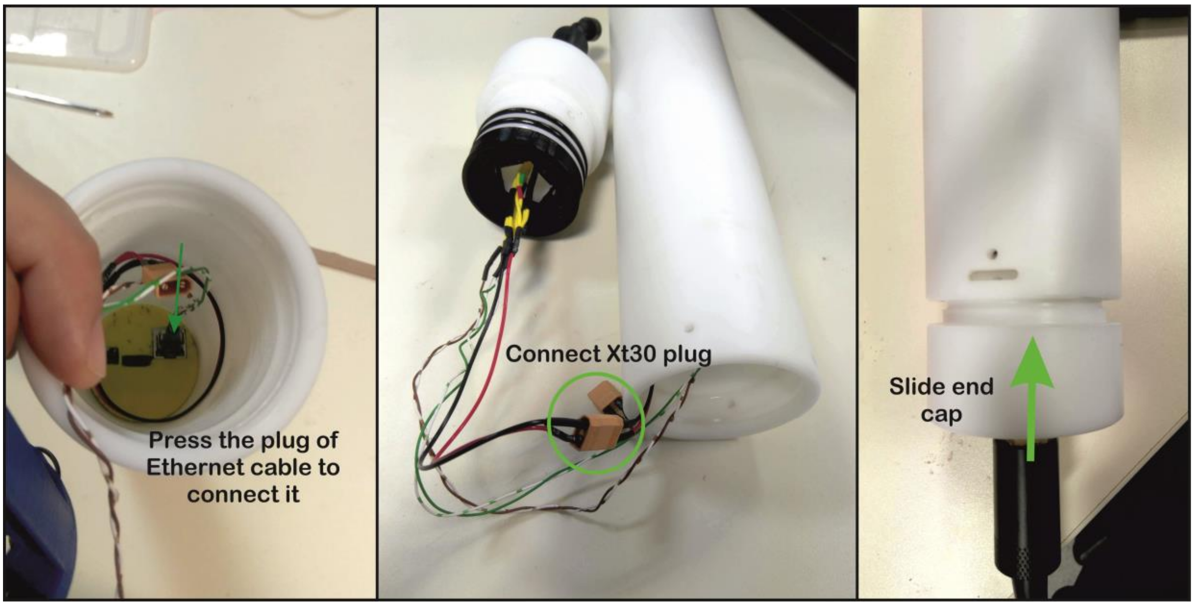


Figure 21 - Connecting Ethernet, power plug and closing cap

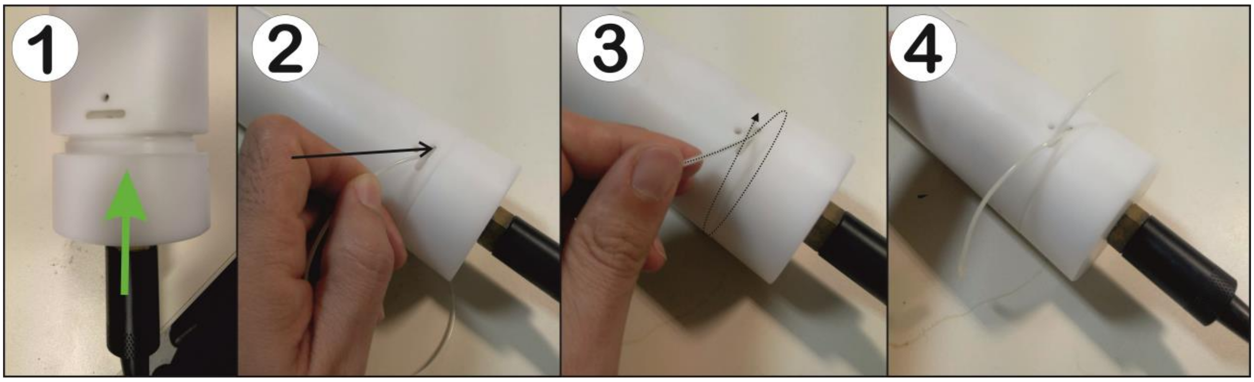


Figure 22 - Closing the end cap

2.4.7 Cabled mode deployment material check list

Before leaving for deployment site check if you carry the basic material for cabled operation. The Table 5 - Cabled mode material check listresumes the more useful material to take.

Material	Notes
DAVS Peli case	
DAVS Base Station case	
Lead Acid Battery	1 unit
Lead Acid charger	Depending on deployment time
Zip Ties	Large size
Duct Tape	Large size
Fixation Supports	If there's any for the experiment
Cleaning cloth	To dry DAVS after retrieving it from the water

Table 5 - Cabled mode material check list

Chapter 3 DAVS Operation: During deployment

This chapter will focus on the onsite deployment of DAVS device, for any of the operating modes. It contains information to prepare a deployment as the attachment to any structure and some verifications that can be done.

3.1 Autonomous mode

The deployment in autonomous mode allows the attachment of DAVS to any AUV or to any fixed structure, leaving the device working standalone. This method and the chose settings should be well planned, since it´s not possible to see what is going on during the acquisition.

3.1.1 Orienting DAVS and Attaching to a support structure

Since DAVS system is a vector sensor, it captures data which has directional information as the accelerometer readings. So, it´s important to decide an orientation for the deployment and respect that orientation when attaching DAVS to any structure. Also, DAVS have an inertial sensor which records the position of the device, useful when deploy attached to an AUV for example positioning. In Figure 23 can be seen the acoustic sensors axis orientations as well as the inertial sensor axis, which are not the same.

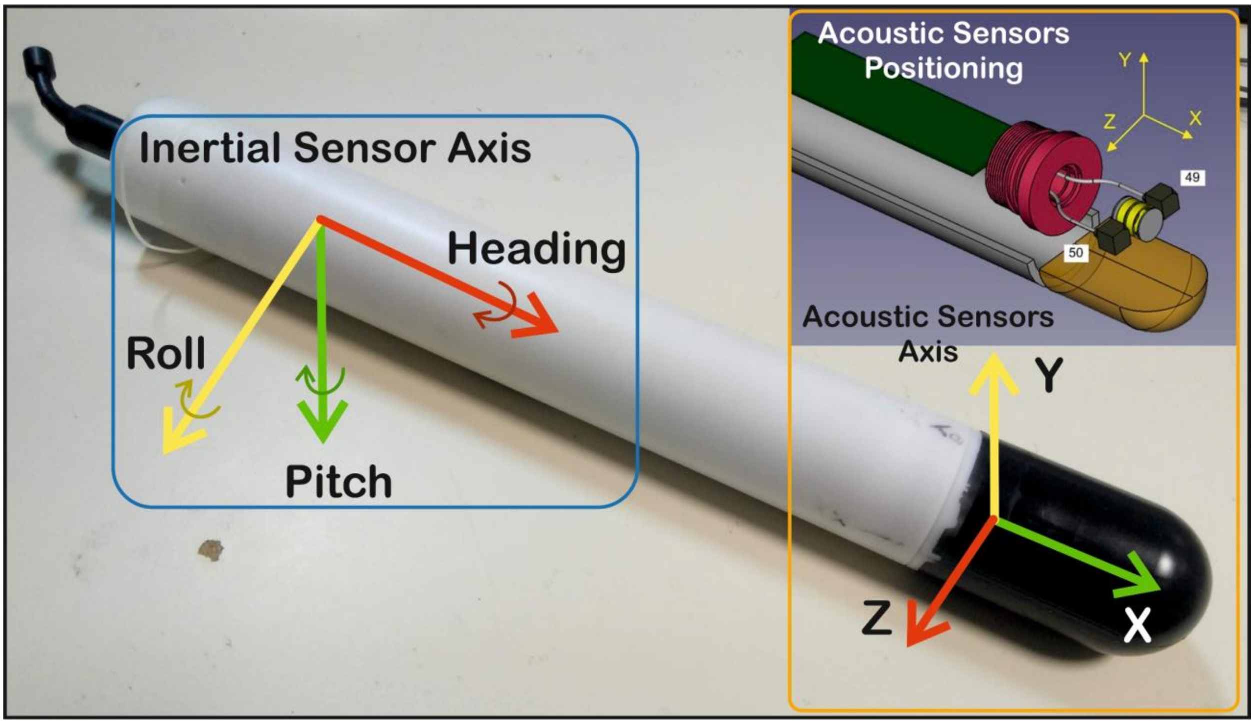


Figure 23 - DAVS orientation axis

Before attachment DAVS should be rotated along each axis to calibrate the inertial sensor. Just rotate twice along each axis to guarantee that the calibration is done. Also note that the inertial sensor is prepared for a vertical fixation. A horizontal attachment will generate wrong compass values.

The attachment of DAVS to the development structure can be made using any specific supports or using a simple duct tape or zip ties. The duct tape solution also prevents the rotation of DAVS, so should be used in conjunction to Zip Ties. In Figure 24 (1) and (2) can be seen some specific Delrin clamps made for the exact sizes of DAVS and the AUV. In (3) a duct tape solution is used, inserting some foam pieces between the AUV structure and DAVS. This foam is useful to reduce some of the mechanical noise that could occur with direct contact between the rigid surfaces. Any adopted solution should take care of tighten enough the device, preventing it from slipping but not tighten it to the point of deforming the case, which can result in water leakage.

In Figure 24(3) there is also the DAVS axis represented, which are of extreme importance to respect. Since the DAVS has a directional response, the attached position should be the one which permit the best results for the acquisition scenario. When attaching DAVS the acoustic axis position should be considered and recorded for later data processing.

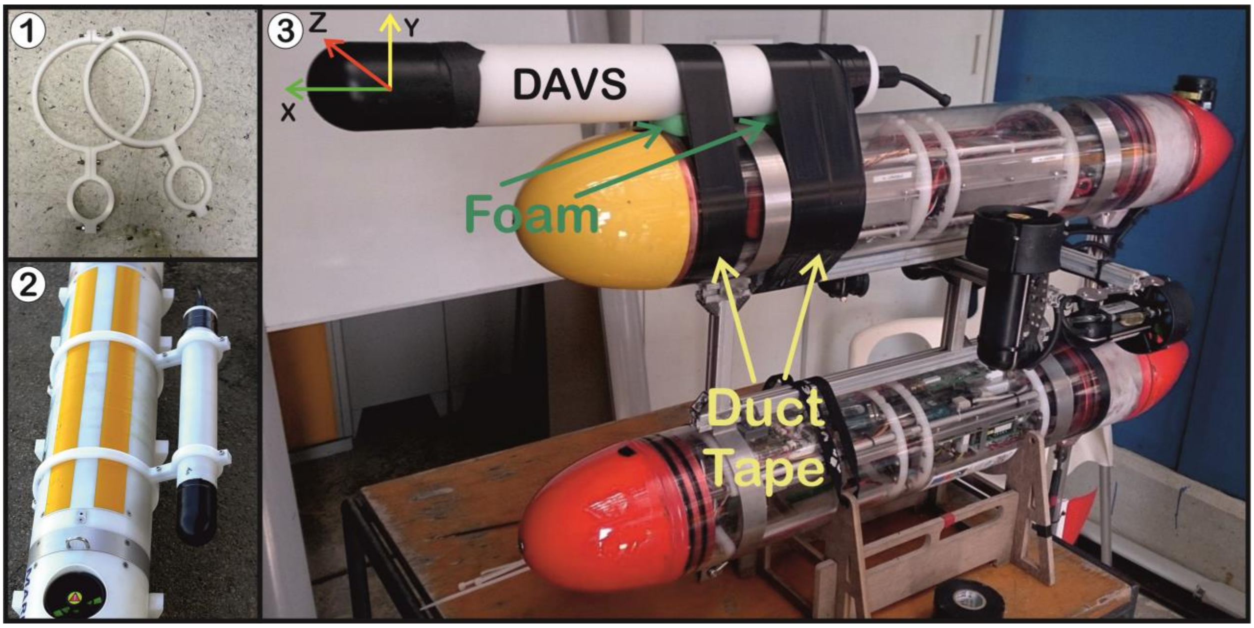


Figure 24 - DAVS Delrin supports (1) made for a specific AUV attachment (2). DAVS attached to a AUV using duct tape (3). Note the orientation of the device by the axis representation.

3.1.2 Data quality assurance check

When deploying DAVS in autonomous mode, the recorded data remains inaccessible before the recovery. So, it´s not easy to check if the recorded data are good for the purpose of the experiment. The gain settings of the programmable amplifiers should be set to an appropriate level for the type of signals expected by the experiment. An experiment for record underwater ambient noise for example, will need a gain high enough to get a good signal to noise ratio. However, if there´s a possibility to exist loud sources in the deployment site, this high gain can lead to saturation and clipping of the captured signal. So, these settings should consider the expected sound levels for that acquisition.

Depending on the experiment, some checks can be made to guarantee a good quality of the recorded data. Always check for LED blinking for 10 seconds (Figure 25), any other time or no LED light indicates some problem which should be checked. Please note that during daylight it can be difficult to see the LED light through the Delrin cylinder, so choose a dark location.

If it's possible to retrieve DAVS during the experiment (for batteries change), or immediately after recovering it, pick the SD card with the recorded data and check it in a computer, taking attention to:

- Visually check the recorded data for data presence. If no data is present in the WAVE file or no WAVE files exist, then there was some problem with DAVS system start up. Check that SD card was correctly formatted and configured, by doing a simple procedure. With the DAVS nose opened and SD card inserted, plug the battery pack for some minutes. Then unplug battery pack and check again for data recorded. If, there's no WAVE files check for configuration file format. Any wrong format (with text in any line, for example) will simply not store any WAVE file.

- Check for the amplitude of the interest signals (dynamic range of the signals). If there´s some saturated zones, the amplifier gains should be reduced. If the interest signals are weak, the gains should be raised.
- Hear the recorded audio for each channel searching for clipping and saturated zones, mechanical noises from attachment or platform, unwanted anthropogenic noises or other environmental noises. If audio files are saturated, the amplifier gains should be reduced. If there's some mechanical noises from the mechanical attachment, the fixation method should be changed. If the deployment site has meaningful unwanted anthropogenic noises, probably it´s not a good place to deploy.

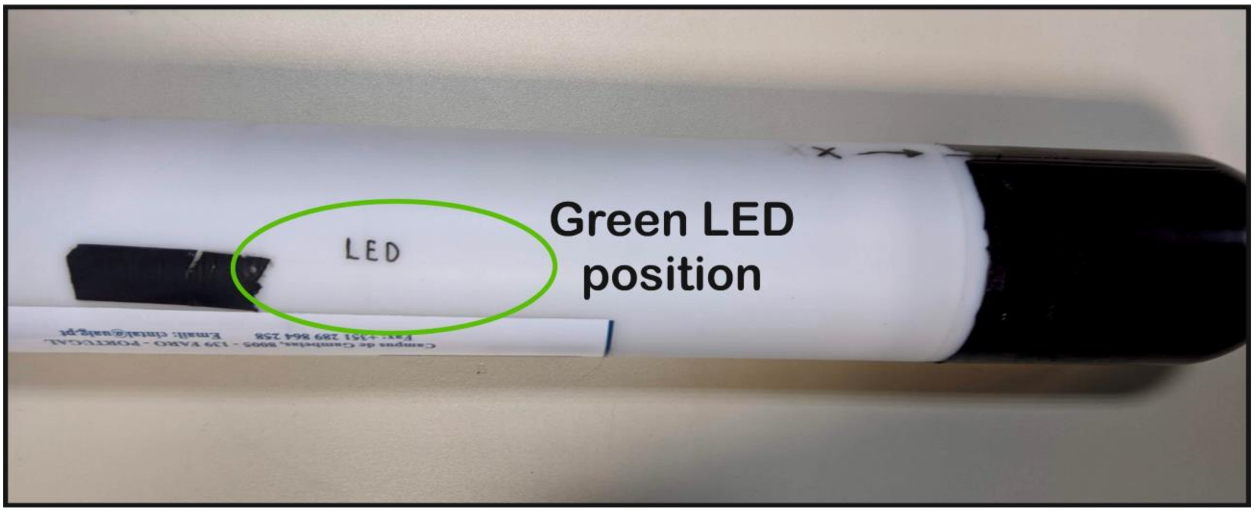


Figure 25 - Green LED position in Delrin cylinder

3.2 Cabled Mode

The DAVS cabled mode allow the connection to an external host computer and adjust some setting during the experiment.

3.2.1 Connecting DAVS Base Station external Ethernet cable

To connect the external Ethernet cable of DAVS Base Station first remove the dummy caps from both the DAVS and DAVS Base Station connectors (Figure 26). Start by unscrewing the DAVS locking sleeve in the direction of green arrow (1) and then pull the dummy caps apart gently, storing them in a safe place. Then connect the green cable terminal plug to DAVS connector (2) and screw the threaded connector sleeve (3).

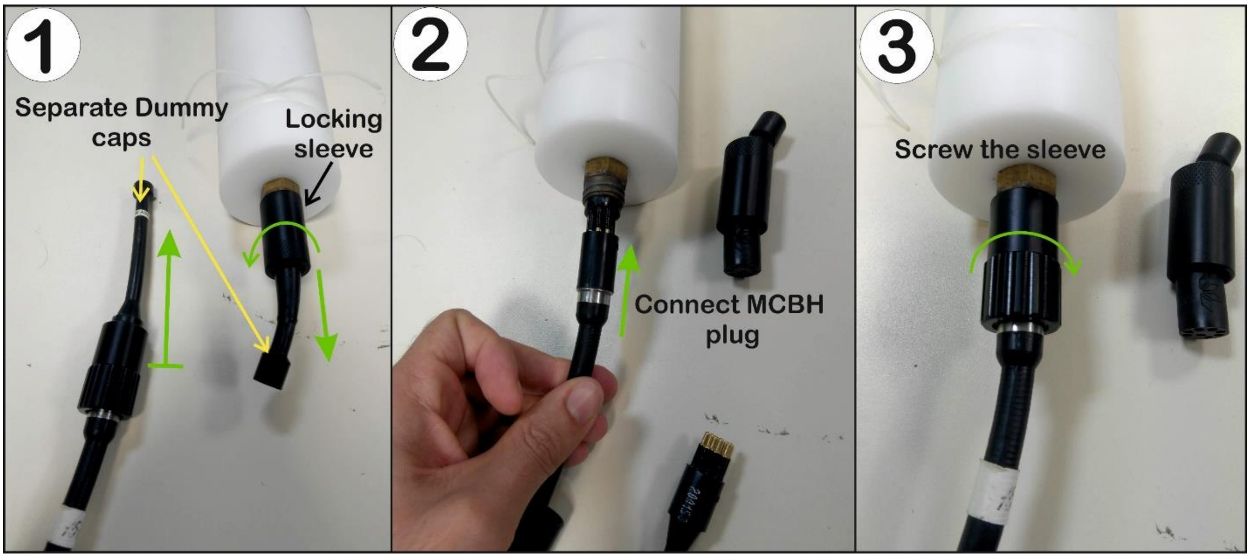


Figure 26 - Connection of DAVS Base Station cable to DAVS connector

3.2.2 Orienting DAVS and Attaching to a support structure

Since DAVS system is a vector sensor, it captures data which has directional information as the accelerometer readings. So, it´s important to decide an orientation for the deployment and respect that orientation when attaching DAVS to any structure. Also, DAVS have an inertial sensor which records the position of the device, useful when deploy attached to a boat that can change its positioning. In Figure 23 of chapter 3.1.1 can be seen the acoustic sensors axis orientations as well as the inertial sensor axis, which are not the same. Before attachment to support structures, DAVS should be rotated along each axis to calibrate the inertial sensor. Just rotate twice along each axis to guarantee that the inertial sensors are calibrated. Also note that the inertial sensor is prepared for a vertical attachment, any horizontal attachment will give wrong readings.

The attachment of DAVS to the development structure can be made using any specific supports or using a simple duct tape or zip ties. The duct tape solution prevents the rotation of DAVS, so should be used in conjunction to Zip Ties. When attaching to any platform, respect the defined orientation of DAVS. In Figure 27 can be seen three different attachments to support structures, all using duct tape and zip ties. The rightest image has marked the DAVS acoustic sensor axis, which should be recorded for the later signals processing.

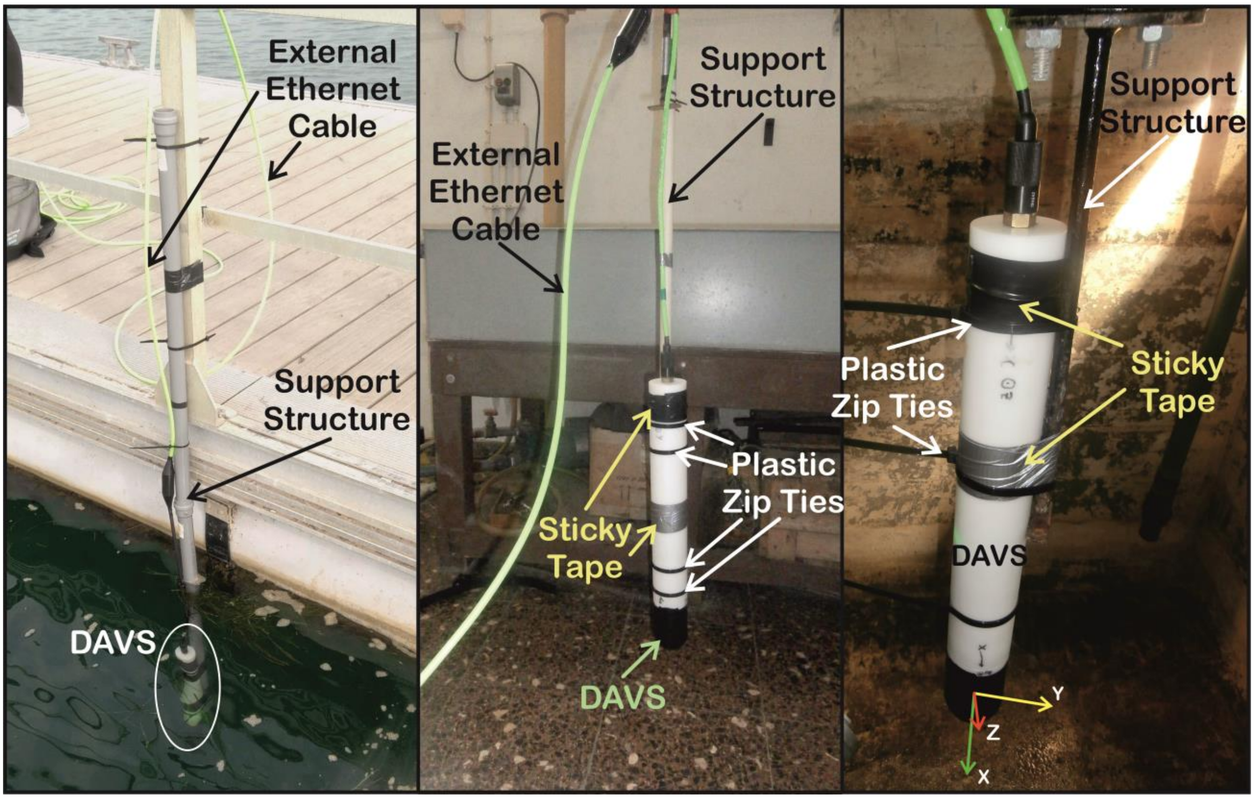


Figure 27 - Three different DAVS attachments using duct tape and zip ties. Please note the axis on the right photo

3.2.3 Connecting Marsening system VRLA battery

Insert a charged lead acid battery into the DAVS Base Station case battery area and arrange the power cables inside (Figure 28). A charged battery has a voltage around 13 Volt and can be checked with a multimeter. Connect the battery plugs, respecting the colors and starting from the negative pole (black). Then connect the positive pole (red) quickly, to minimize voltage spikes in the system. Make sure the connectors are well settled. You will see a red LED light blinking, showing that the system is booting up. The case should be closed to prevent water or moisture to get inside. Wait for up to 2 minutes for the system to completely start up. After that time the led should be always ON.

Figure 28 - Battery installation in DAVS Base Station case (left side) and power cables connection (right side)

3.2.4 Connecting to DAVS Base Station Hot Spot (Windows 10 OS)

When the Base Station system is fully working, it will act as a Wi-Fi hot spot allowing external devices connection. You will need to connect to Wi-Fi hot spot using the following credentials:

- Wi-Fi Hot Spot Name: **SiPLAB_TP-1**
- Wi-Fi Hot Spot Password: **digitalHyd**

To do this in Windows 10 environment, search for that Wi-Fi network name, choose "*connect*" and enter the password. Click "*next*" and "*yes*" to complete the connection process (Figure 29). You will receive a message saying that the network doesn't have Internet access, and that's OK since you only want to connect to DAVS.

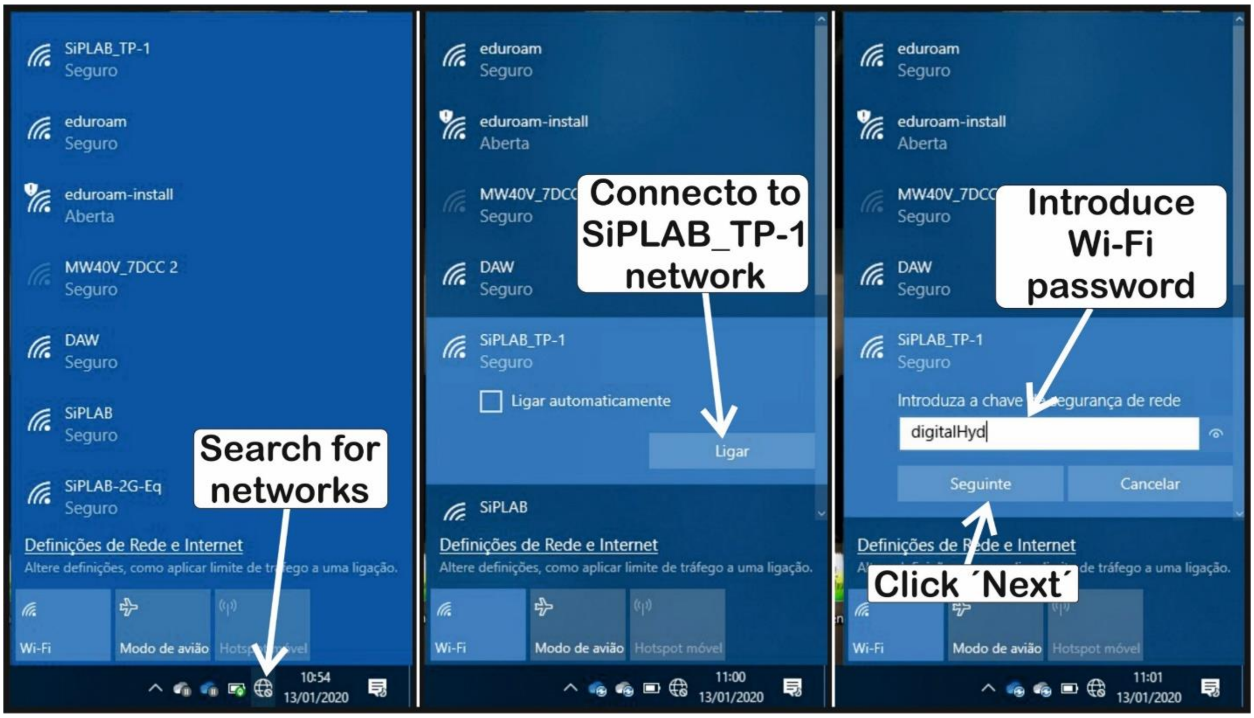


Figure 29 - Connecting to DAVS hot spot in a Windows 10 operative system

3.2.5 Connecting to DAVS Base Station Hot Spot (Android OS)

It's also possible to connect to DAVS Base Station using any mobile phone with Android or iOS operative system. Figure 30 shows an Android device connection example. The interface design can change, based on your Android version or device, but the steps are similar. Start by **disabling mobile** data, then search for a hot spot called "**SiPLAB_TP-1**" in Wi-Fi connections menu, click on it and insert the password "**digitalHyd**". To finish click "*connect*" and wait for the connection to stablish. In the end it should indicate that the device is connected to SiPLAB_TP-1 device but don't have Internet access.

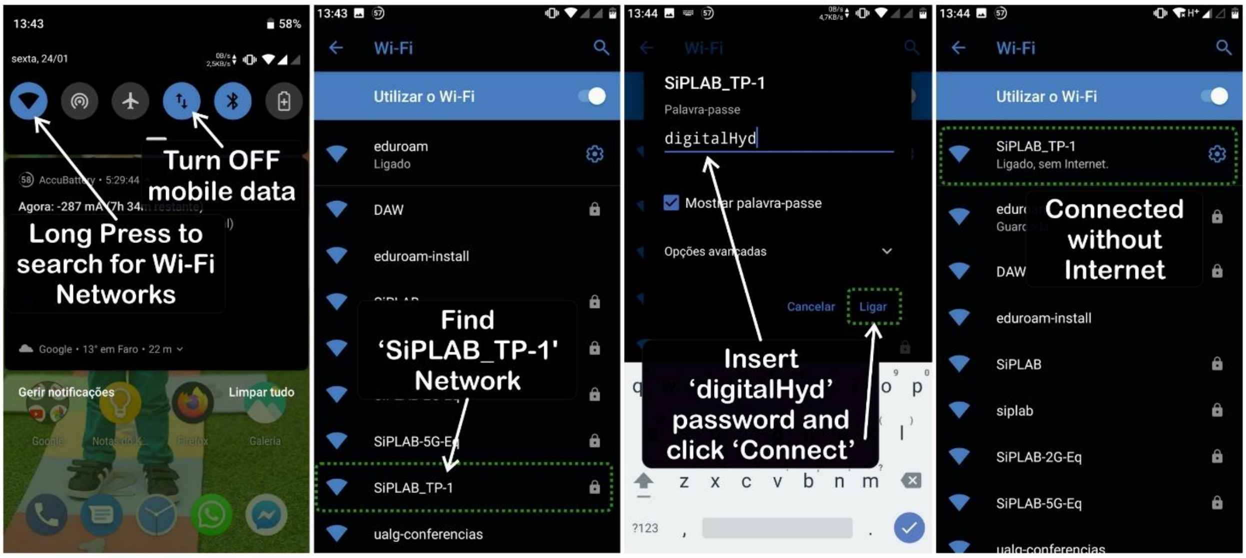


Figure 30 - Connecting to DAVS hot spot in an Android 9 mobile phone

3.2.6 DAVS Base Station real time web interface

After connecting to DAVS Base Station hot spot and access to web interface, open a browser in your computer and navigate to real time display interface at "[192.168.200.248:7681](192.168.200.248:7681)". There you will see the real time streaming data interface with a waterfall plot and the frequency spectrum inside in the left side of the screen. In the right side there are some visualization controls and DAVS settings, that can be changed in real time also.

The visualization window is composed of two components: a frequency spectrum graph and a waterfall plot (Figure 31). The frequency spectrum represents the real time amplitude (X axis) of the signal frequency components (Y axis). The waterfall plot is a dynamic spectrogram which shows how the signal frequencies amplitude changes along time. The amplitude is represented through a color that change based on the amplitude of the signal and follows the power scale color scheme. In Figure 31 an high intensity signal (around 180 dB) is shown in a red color and a weaker signal (near or above 60 dB) is shown in a dark blue color. Moving the mouse through the waterfall plot will identify the frequency of that point. A simplified description of the calculations performed to obtain these plots (called FFT) are given in appendix A.4.

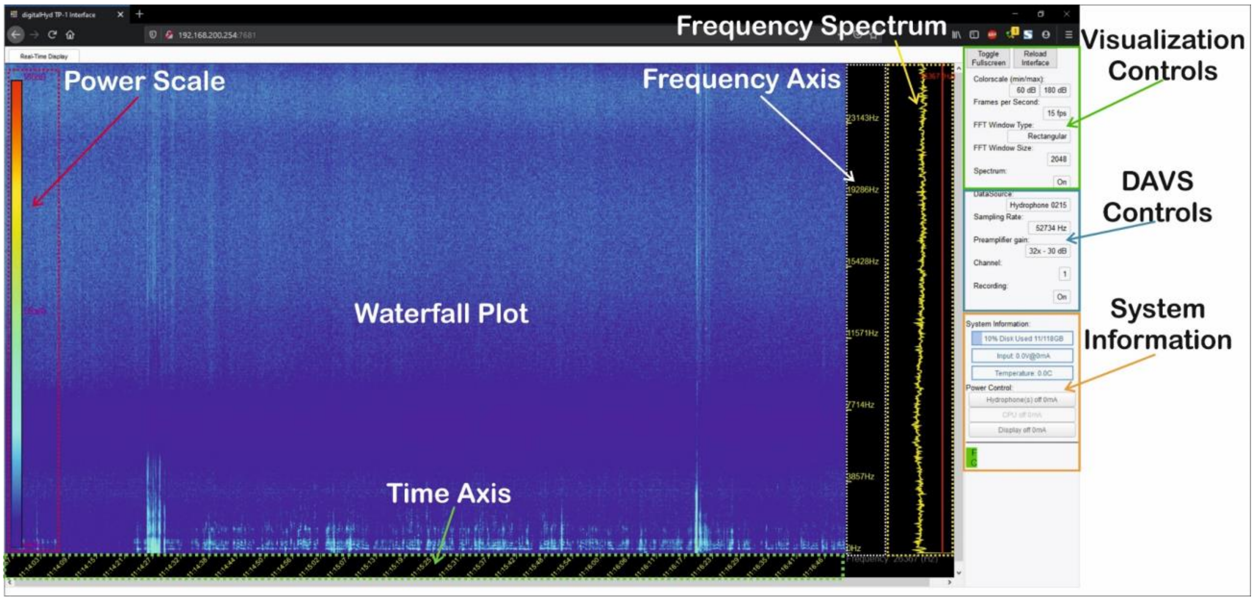


Figure 31 - DAVS Base Station web interface sections and control panels

In Figure 31 you can see the different sections of DAVS Base Station web interface. The right hand top panel shows the visualization controls that adjust how the signal is represented in the visualization window. It doesn't affect the settings of DAVS or saved WAVE files. The following settings are available:

- **Colorscale**, adjust which amplitude of the signal correspond to the minimum (dark blue color) and maximum (red color) of the waterfall color scale. The scale ranges from 20 to 200 dB. For example, choosing a minimum of 40 dB will represent a frequency component with an amplitude of 40 dB (or less) by a dark blue color in the spectrogram.
- **Frames per Second** change the refresh rate of the frequency spectrum and spectrogram. It ranges from 1 to 30 frames per second. A higher frame rate will update the frequency spectrum and spectrogram quicker.
- **FFT Window Type**, which permit to change the window type used by the FFT algorithm. There are Rectangular, Hamming, Hanning, Blackman, Nuttal and Blackman-Nuttal window types available (see more information on appendix A.4).
- **FFT Window Size**, which sets how many data points are used by the FFT algorithm. It has three options (2048, 4096 and 8192) and the higher the value the more frequency resolution you get. However, this is a trade off with time resolution, so getting higher frequency resolution take a lower time resolution.
- **Spectrum**, turns ON or OFF the visualization window.

DAVS controls panel changes how DAVS acquire signals and accordingly, the saved files settings. Changing any of these parameters (except the **Channel** one) will create immediately a new WAVE file with the new settings. Control section has the following options:

- **Datasource**, the name and ID of the device connected, usually "Hydrophone 0215".
- **Sampling rate**, the number of samples per second (Hz) used by DAVS. It has four options following the sample rate of ADC, being 10547, twice 52734 and 105469 Hz. The 52734 Hz has two options related to an ADC configuration that change the Signal to Noise ratio (check appendix A.1.4.3).
- **Preamplifier gain**, changes the gain of the PGA. The options available are 0x, 1x, 2x, 4x, 8x, 16x, 32x and 64x (check appendix A.1.4).
- **Channel,** selects the DAVS channel being visualized in the visualization panel. The available options are presented in Table 6. This setting only affects visualization window, since all the 8 DAVS channel will always be recorded to the WAVE file.
- **Recording,** to turn OFF or ON the recording of DAVS acquired data to a WAVE file.

Transducer	**Accelerometer #49**			**Hydrophone**	**Accelerometer #50**			**NOT USED**
	X axis	***Y axis***	***Z axis***		***X axis***	***Y axis***	***Z axis***	
Panel number	1	2	3	4	5	6	7	8

Table 6 - Channel options for DAVS control panel

System information panel isn't used with the actual DAVS hardware.

When using a mobile device, the procedure is the same, just open any browser and navigate to "*192.168.200.248:7681*" address. The page will have the same layout and functionality of the normal version, as seen on Figure 32.

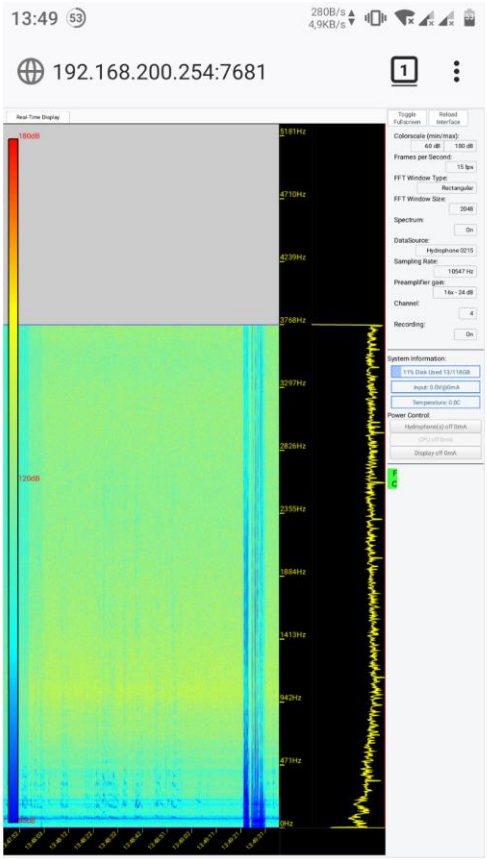


Figure 32 - DAVS Base Stationweb interface, using mobile Firefox (Android OS)

3.2.7 Data quality assurance check

When deploying in Cabled Mode, it´s easy to check if the acquired signals have some problems and correct them before the end of the experiment. Since the web interface allows the real time visualization of the acquired signal, the visual inspection of the waterfall plot will show the amplitudes of the signals. Check if there's any unexpected high level sounds or if the gains are too high or low for the expected signals. If there´s some saturated zones, the amplifier gains should be reduced. If the interest signals are weak, the gains should be raised. The gain settings of the programmable amplifiers should be set to an appropriate level for the type of signals expected by the experiment. Also check for transients that can be caused by impacts and could reveal some problems in the attachment of DAVS to supports.

Hearing the acquired audio can also help to find any problems. The simple way is to download a captured WAVE file and hear it in any software as explained in Chapter 4. Hear the recorded audio for each channel searching for clipping and saturated zones, mechanical noises from attachment or platform, unwanted anthropogenic noises or other environmental noises. If audio files are saturated, the amplifier gains should be reduced. If there's some mechanical noises from the mechanical attachment, the fixation method should be checked and changed. If the deployment site has meaningful unwanted anthropogenic noises, probably it´s not a good place to deploy.

3.3 Troubleshooting

Some common problems that may occur are described in Table 7 below.

Problem	Mode	Cause	Solution	Manual Chapter
DAVS don't start (green LED don't turn on or blink for 10 seconds)	Autonomous	Batteries are discharged	Check batteries and change them for charged ones	2.3.3 and 2.3.7
		XT30 connector not correctly plugged	Check XT30 connector, reconnecting it.	2.3.8
	Cabled	Underwater connector disconnected	Check underwater connector and reconnect it	3.2.1
Green LED blink for 1 second	Autonomous	SD card not formatted into FAT filesystem	Retrieve SD card and format it in FAT filesystem	2.3.1 and 2.3.2
		SD card into slot B	Change SD card from slot A to Slot B	2.3.5
Web interface get stucked or don't show any DAVS real time data	Cabled	Internal error	Reboot base station, by disconnect and reconnect lead battery	3.2.3
Can't connect to web server	Cabled	Base station system not connected (red LED turned OFF)	Check lead battery charge state and connection	2.4.2 and 3.2.3
		Red LED turned ON	If using a mobile device, disconnect "mobile data"	3.2.5
Find WiFi network but don´t connect	Cabled	Inserted a wrong password	Forget Wi-Fi connection and reassociate again. Double check password	3.2.4 or 3.2.5
DAVS don't recorded any WAVE file	Autonomous	Wrong "config.txt" file	Check if configuration values or "config.txt" file follow the correct format and values	2.3.1
DAVS compass information is wrong	Autonomous or cabled	Wrong attachment position	ReAttach DAVS in a vertical position	3.1.1 or 3.2.2

Table 7 - Troubleshooting for the most common problems

Chapter 4 DAVS Operation: After deployment

This chapter will focus on the after deployment procedures, for any of the operating modes. It contains information on how to retrieve and process data acquired by DAVS. There are also some considerations about maintain and store DAVS.

4.1 Data retrieve

Retrieve data from DAVS is obviously dependent on deployment method. The following chapters describe the procedures for the two modes.

4.1.1 Autonomous Mode

When in autonomous mode DAVS should be retrieved from water and dried before opening, to prevent water spilling to electronic boards. The opening of DAVS follows what is described in chapters 2.3.4 and 2.3.5. After picking up the SD card, connect it to any computer and retrieve the files by copying them to your computer.

Each time DAVS is plugged ON, it will create a folder named “DATAXXXX”, where the XXXX are a number from 0000 to 9999. When creating a new folder DAVS check for the last folder name existing in root folder of SD card, creating a new folder with the next number. For example, if when booting there are already some folders named “DATA0000” to “DATA0005”, DAVS will create a new folder called “DATA0006” and store WAVE files inside this one. The WAVE files have the same naming scheme, starting at “DATA0000.WAV” to “DATA9999.WAV”. In Figure 33 can be seen an example of the content of a SD card, with several folders following this naming scheme, and the content of a folder “DATA0012” where some WAVE files are.

It’s a good practice to keep the original files untouched, maintaining them without any editing or processing. So, duplicate the original files and use the duplicated files for processing.

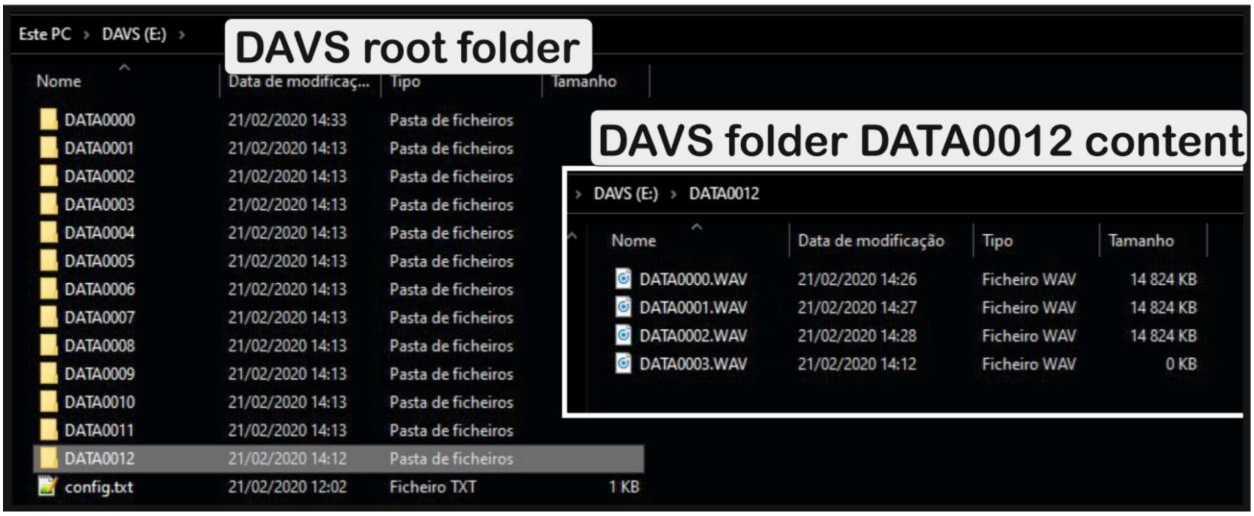


Figure 33 - Example of the content of a SD card and data folder, from a DAVS acquisition

4.1.2 Cabled mode

DAVS Base Station retrieves acoustic and system information from DAVS system, storing these information’s into 2 separate files. The acoustic data of the 7 acoustic channels is stored in a WAVE file format, with 120 seconds of duration. Every 120 seconds period a new file is created, being this time fixed in firmware. To retrieve the acquired acoustic data just navigate in a browser to “*192.168.200.254/data/*”, where a web interface allows to download the acquired data. The WAVE data files are separated into folders, following a path structure of “*192.168.200.254/data/YEAR/MONTH/DAY/*”. Just navigate to the folder matching the date of acquired signals and select the files you need to download them to the host device. WAVE data files follow a naming scheme containing DAVS ID and acquisition start time as ”*DATA_TP1_id_XXXhhmmss.wav*”. For example, one file recorded at 10:57:08 of day 13 January 2020 by device ID 0215, will be placed at folder path *“/data/2020/01/13/”* with a filename “*DATA_TP1_0215_012105708.WAV*”. Figure 34 shows an example folder, with the recorded files listed and a download window asking to open or save the file. In an Android mobile device, the listing page will have the same functionality.

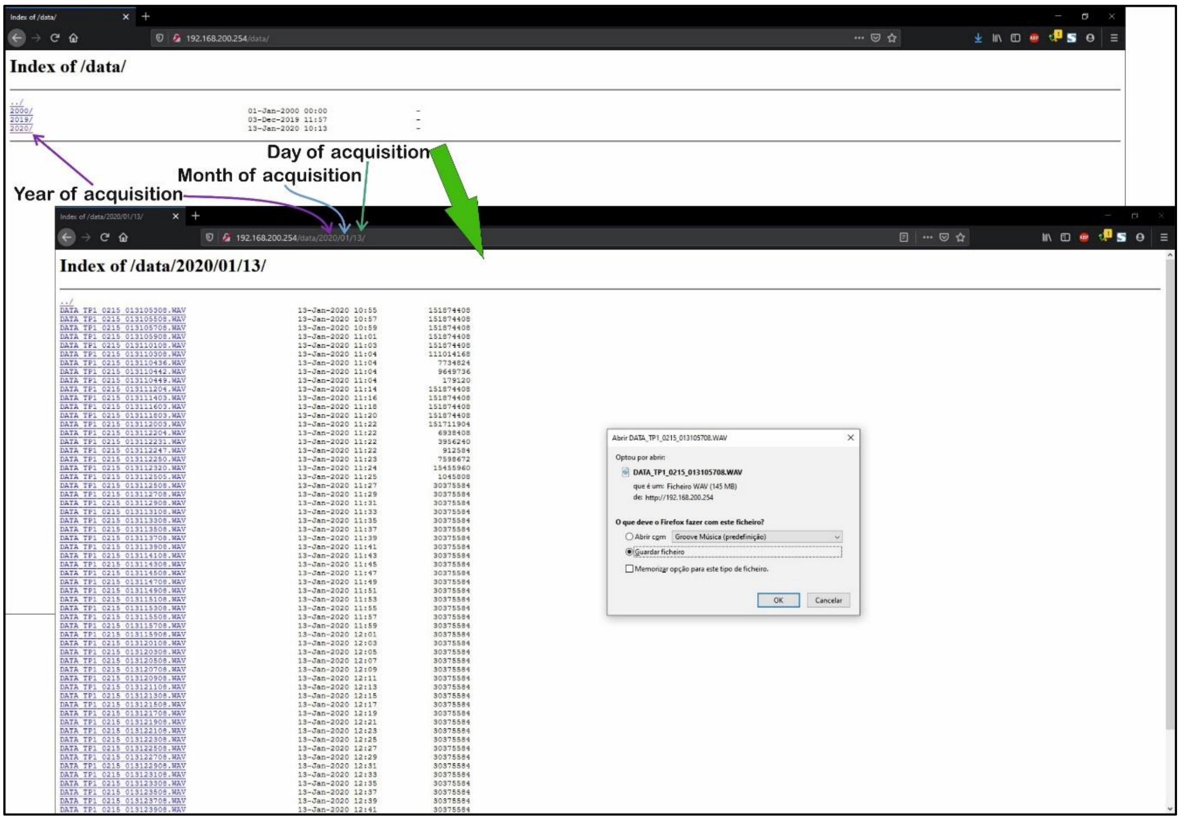
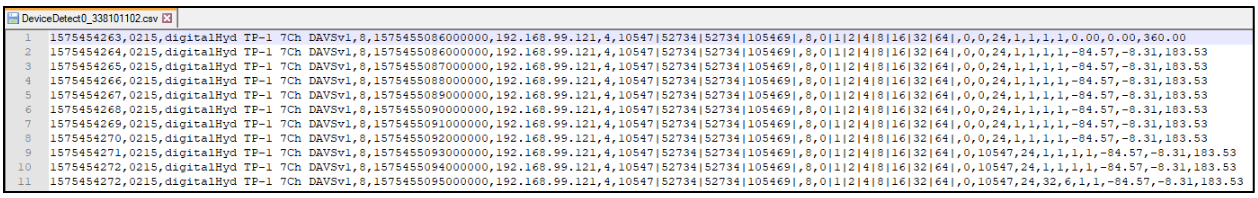


Figure 34 - Folder structure of data folder, from web interface (Firefox browser in Windows 10 OS)

Inside each day folder there are some “*DeviceDetect0_XXXhhmmss.csv”* files, where several information’s about DAVS are stored. In these files, all the data are stored using comma separated values (CSV) format. The file name reflects the creation time of the file in the “*hhmmss*” part, using Base Station internal time. Each file line represents one second of system activity or a change in settings done through web interface. There are 20 fields that represent some specific information about DAVS, as seen on Figure 35. These fields are described in more detail on appendix A.5.



```
DeviceDetect0_338101102.csv
1575454263,0215,digitalHyd TP-1 7Ch DAVSv1,8,1575455086000000,192.168.99.121,4,10547|52734|52734|105469|,8,0|1|2|4|8|16|32|64|,0,0,24,1,1,1,1,0.00,0.00,360.00
1575454264,0215,digitalHyd TP-1 7Ch DAVSv1,8,1575455086000000,192.168.99.121,4,10547|52734|52734|105469|,8,0|1|2|4|8|16|32|64|,0,0,24,1,1,1,1,-84.57,-8.31,183.53
1575454265,0215,digitalHyd TP-1 7Ch DAVSv1,8,1575455087000000,192.168.99.121,4,10547|52734|52734|105469|,8,0|1|2|4|8|16|32|64|,0,0,24,1,1,1,1,-84.57,-8.31,183.53
1575454266,0215,digitalHyd TP-1 7Ch DAVSv1,8,1575455088000000,192.168.99.121,4,10547|52734|52734|105469|,8,0|1|2|4|8|16|32|64|,0,0,24,1,1,1,1,-84.57,-8.31,183.53
1575454267,0215,digitalHyd TP-1 7Ch DAVSv1,8,1575455089000000,192.168.99.121,4,10547|52734|52734|105469|,8,0|1|2|4|8|16|32|64|,0,0,24,1,1,1,1,-84.57,-8.31,183.53
1575454268,0215,digitalHyd TP-1 7Ch DAVSv1,8,1575455090000000,192.168.99.121,4,10547|52734|52734|105469|,8,0|1|2|4|8|16|32|64|,0,0,24,1,1,1,1,-84.57,-8.31,183.53
1575454269,0215,digitalHyd TP-1 7Ch DAVSv1,8,1575455091000000,192.168.99.121,4,10547|52734|52734|105469|,8,0|1|2|4|8|16|32|64|,0,0,24,1,1,1,1,-84.57,-8.31,183.53
1575454270,0215,digitalHyd TP-1 7Ch DAVSv1,8,1575455092000000,192.168.99.121,4,10547|52734|52734|105469|,8,0|1|2|4|8|16|32|64|,0,0,24,1,1,1,1,-84.57,-8.31,183.53
1575454271,0215,digitalHyd TP-1 7Ch DAVSv1,8,1575455093000000,192.168.99.121,4,10547|52734|52734|105469|,8,0|1|2|4|8|16|32|64|,0,10547,24,1,1,1,1,-84.57,-8.31,183.53
1575454272,0215,digitalHyd TP-1 7Ch DAVSv1,8,1575455094000000,192.168.99.121,4,10547|52734|52734|105469|,8,0|1|2|4|8|16|32|64|,0,10547,24,1,1,1,1,-84.57,-8.31,183.53
1575454272,0215,digitalHyd TP-1 7Ch DAVSv1,8,1575455095000000,192.168.99.121,4,10547|52734|52734|105469|,8,0|1|2|4|8|16|32|64|,0,10547,24,32,6,1,1,-84.57,-8.31,183.53
```

Figure 35 - DeviceDetect file content, opened in a notepad application

4.2 Data visualization

In this chapter we will present some simple examples to visualize and process the acquired data, using 2 different software, **Audacity** and **Matlab**. We will not deepen any explanation about them, since it´s not the purpose of this chapter.

4.2.1 Audacity

A simple way to view and process the acquired acoustic data is to use the free **Audacity** audio editing software. It's available for Windows, Linux or Mac platforms and in our examples the Windows version has been used. The functionality should be similar in any other operating systems.

4.2.1.1 Opening and playing of a WAVE file

To open a DAVS WAVE file simply start Audacity and select *File>>Open* or "Ctrl+O" shortcut, selecting the desired file from storage (Figure 36).

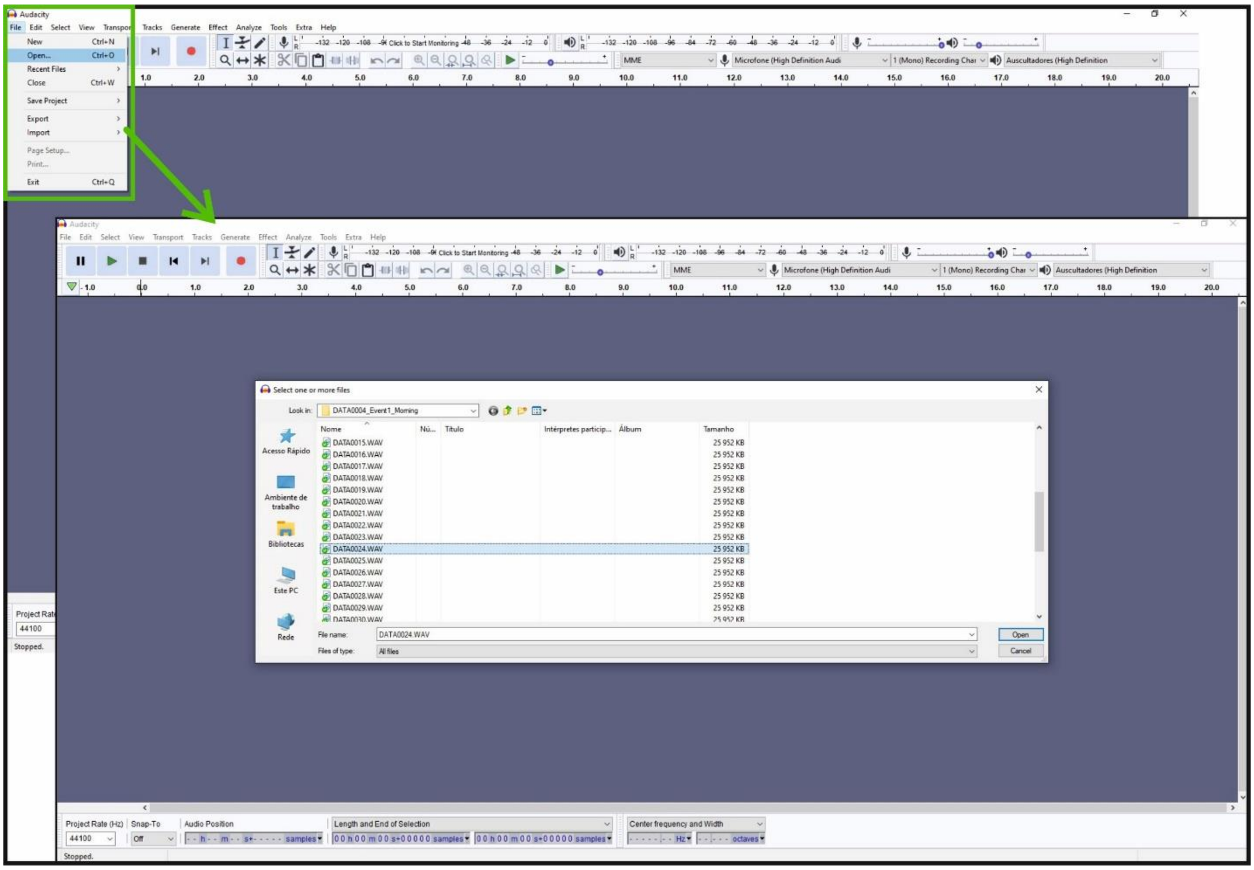


Figure 36 - Audacity window and open file dialog

When a WAVE file is loaded, it's possible to hear it and see the corresponding normalized waveform (range -1 to 1) of each recorded channel. To hear it, just press the *play button* or use the "*space bar*" shortcut. All the seven channels will play simultaneously. It's possible to mute or solo any channel in the channel menu, as well as adjust channel intensity. There are some additional options available in the channel options menu. In Figure 37 can be seen the channels position and the location of each menu.

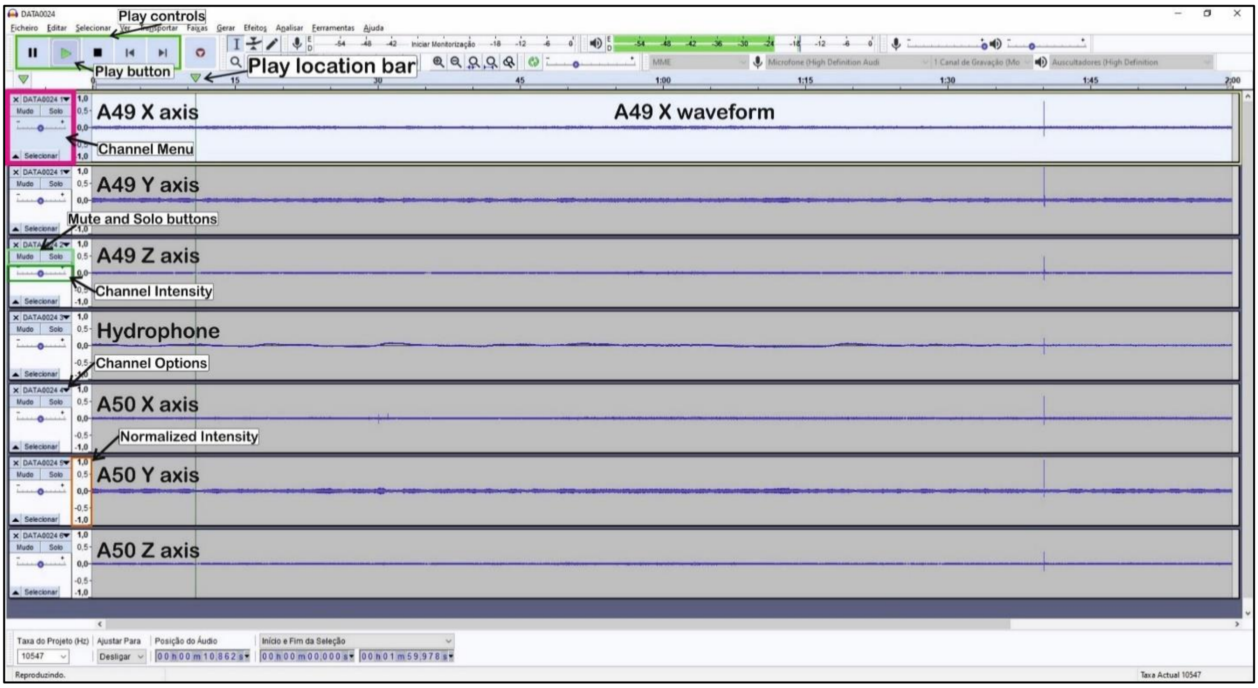


Figure 37 - Basic menus and recorded channels location after open a WAVE file

The waveforms represent the sampled points in time, as can be seen on Figure 38. If press "*CTRL+Mouse Scroll*" it´s possible to zoom in until see the individual samples. These are spaced of 1/(sampling rate) seconds. In the example below there are 10547 samples per second (SPS), which correspond to approximately 94.813 microseconds between consecutive samples. Note the amplitude of each sample which is normalized between -1 and 1.

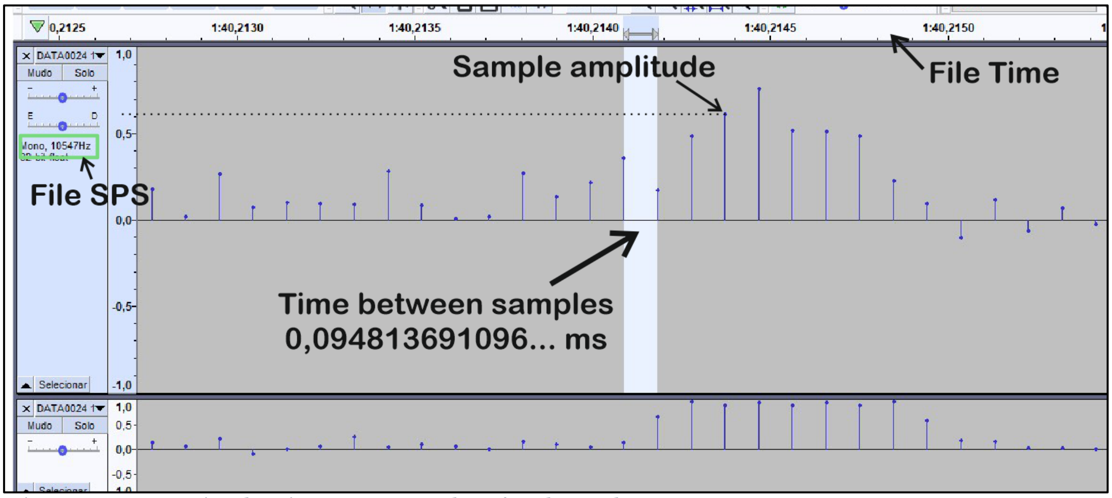


Figure 38 - Zoom in showing some samples of a channel

4.2.1.2 Viewing the spectrogram of a channel

To visually analyse a channel it's common to use a spectrogram, which represent the amplitude of each signal frequency component, in time domain. To change from a waveform to spectrogram view in Audacity, simply select "*spectrogram*" in the "*channel menu*" (1 in Figure 39). The amplitude is represented by a color, the frequency is plotted in Y axis and the time in X axis. To get a better view, drag the border of the channel window to a bigger size (2). There you can see the spectral behaviour of the signal, where the strongest signals get colored red and the weaker ones get a light blue or grey color (3). The X axis has the relative record time of the file, not the real capture time. For example, a 120 seconds record will have a time from 0 to 120 seconds only. The Y axis has the frequency scale in Hertz, which represent values from 0 to half the sampling rate (due to Nyquist Theorem). In Figure 39, the 10547 SPS will get a maximum frequency of 5273 Hz.

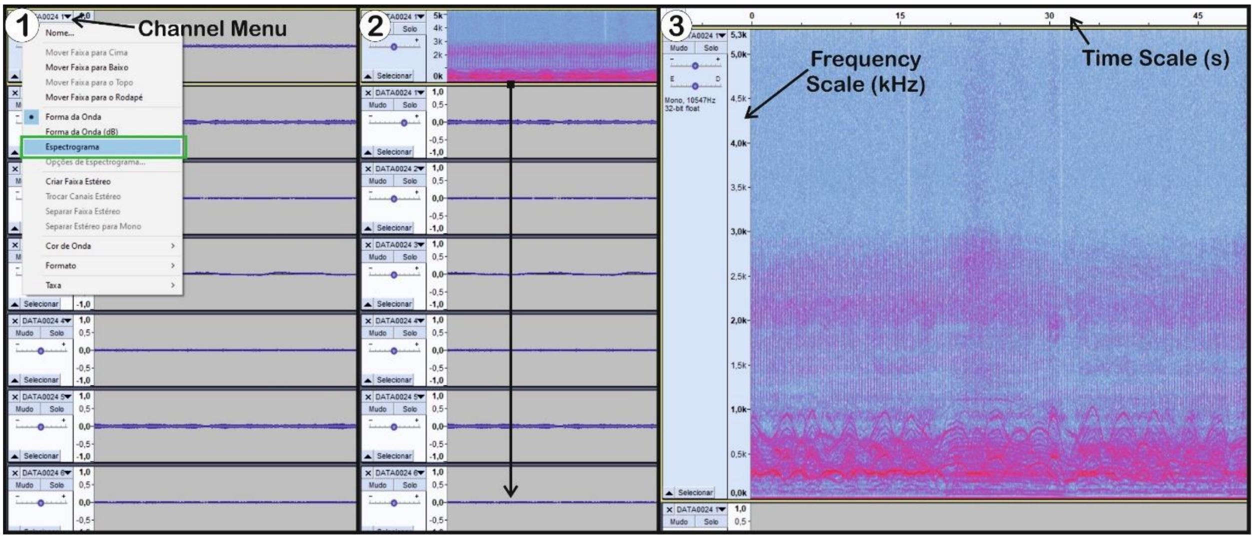


Figure 39 - Change from waveform view to spectrogram view

The spectrogram is generated by a mathematical operation called Fast Fourier Transform (FFT), which is outside the scope of this text. A short explanation was given in appendix A.4. The most common FFT settings are the window type and size, which can be configured in Audacity by clicking in the channel menu and "*spectrogram settings*". Beside that algorithm options, it's also possible to change the colors and scale, however we will leave the default options. Adjusting the best colors depends on the signal strength and can be done using "*gain*" and "*range*" options. Frequency and time resolution can be changed by adjusting the "*algorithm window size*". A lower value will get a high temporal resolution with a low frequency resolution. A high window value will result in a high frequency resolution but a lower temporal resolution. For more information on these settings please refer to Audacity spectrogram manual.

4.2.1.3 Frequency analysis of a file

You can analyse in detail the frequency of a specific zone of the spectrogram. This is also done by a Fast Fourier Transform. Just select some spectrogram area with the mouse, then go to "*Analyse"* menu and "*Plot Spectrum"* which will open a spectrum analyser window, with several option to configure. For more information about these options, please refer to Audacity plot spectrum manual. In the example of Figure 40 we select a zone between approximately 45.8 and 45.9 seconds, which will give us the spectrum from the selected time.

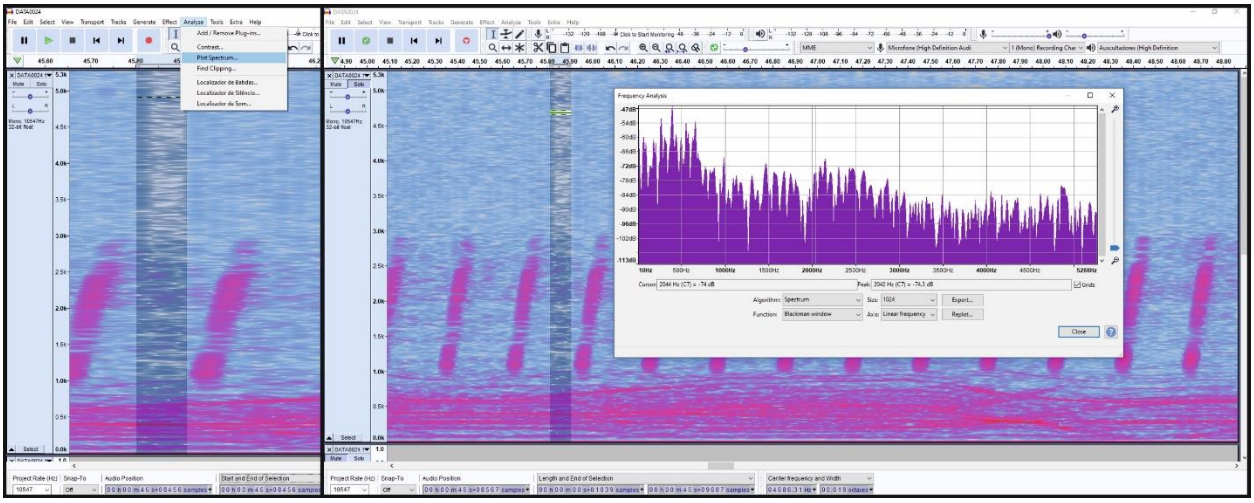


Figure 40 - Frequency analysis plot from a selected region of the spectrogram

Choosing the right parameters (size and window type) will help us to see where the most relevant frequencies are. In Figure 41 there are three big peaks marked in the frequency graph and the correspondent zone in the spectrogram. It can easily be seen that higher peaks in the frequency graph have a brighter red color in the spectrogram.

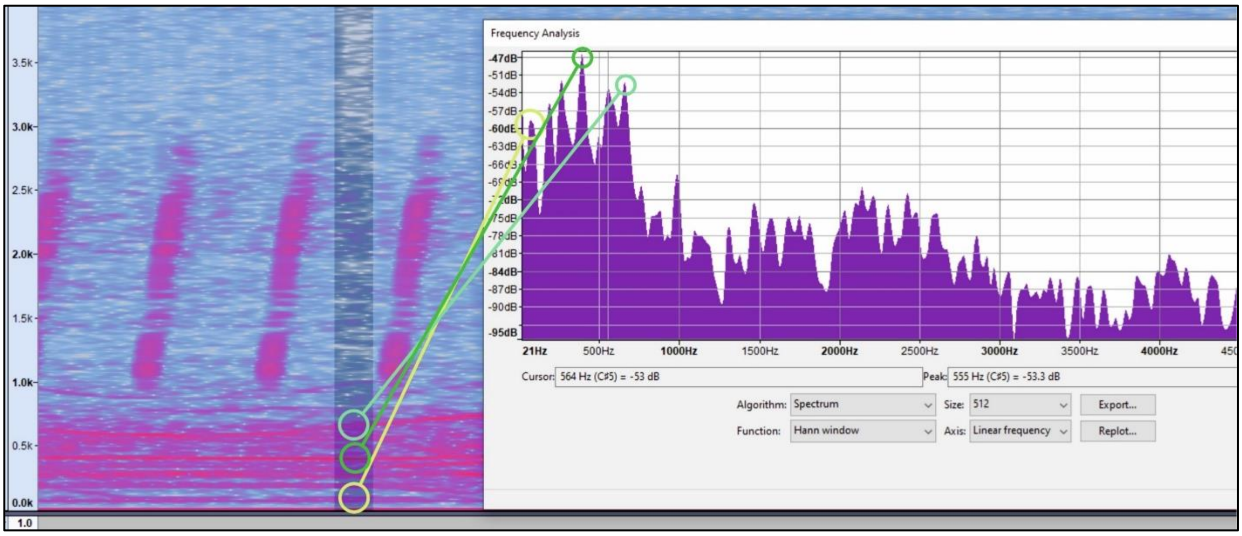


Figure 41 - Frequency plot correspondence in spectrogram

4.2.1.4 Applying a filter to the signal

It's also possible to apply some effects to the signal. For example, if we know that our signal of interest is between 1 and 3 kHz, it's possible to filter all the unwanted frequencies of the signal using the "*filter curve effect*". Just select the area where we want the effect to be applied, go to "*Effect*" menu and select "*Filter Curve*". Then you can design the filter behaviour, where unwanted frequencies can be attenuated, and desired frequencies can be amplified. In the graph windows you can add multiple points and set different zones with different gains. The 0 dB line is where the signal remains the original, the positive values means amplification and negative attenuation. In Figure 42 we design a band pass filter, where we remove all the frequencies below 1 kHz and above 3 kHz. For the unwanted ones we define a -30 dB attenuation. For the usable 1-3 kHz we define a small gain of 2 dB. After we apply it to the signal, we can see that almost all the signal outside our pass band is removed. The exception for the high intensity low frequency signals, which were strong enough to, even after reducing it 30 dB, maintain in the signal.

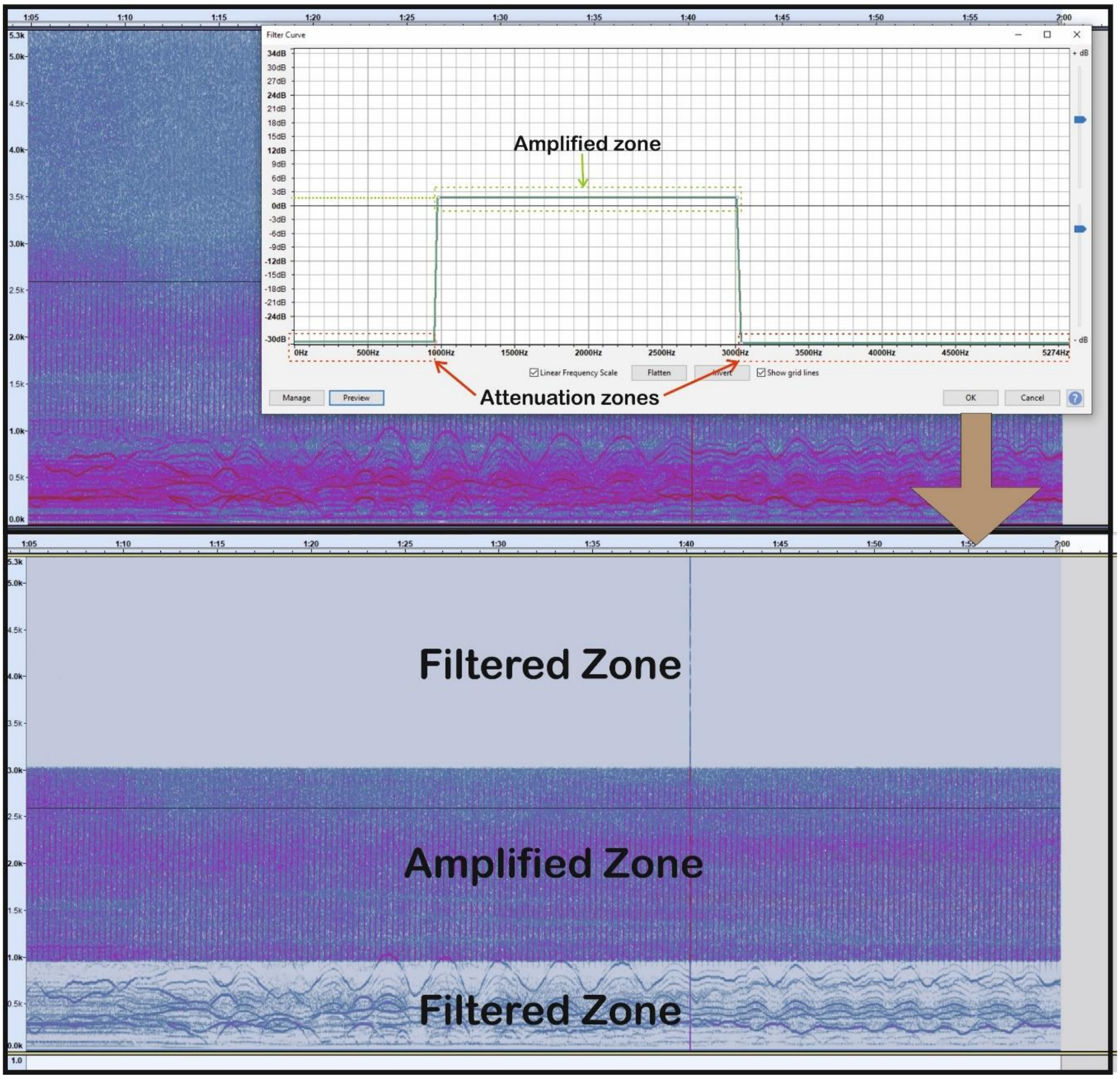


Figure 42 - Example of filtering the signal to remove unwanted frequencies

4.2.2 Matlab

Matlab is a commercial software, commonly used in the academic community. It has good tools for data processing and visualization and some toolboxes for specific applications like signal processing, audio processing or statistics. It's available for Windows, Linux or Mac platforms and in our examples the Windows version has been used. The functionality should be similar in any other operating systems. In our examples we will use two functions (*spectrogram* and *bandpass*) that are a part of "*Signal Processing Toolbox*" so make sure you have it installed. The complete code is on appendix A.3.

4.2.2.1 Opening and playing of a WAVE file

The usage of **Matlab** software gives us some good processing tools. Here we will show a simple example for plotting the waveforms and hearing a WAVE data file. We will not explain the **Matlab** options in detail, since there are so many and they are not needed for the explanation of these simple examples. Please refer to **Matlab** manual.

To open a file, define the path in the computer to the WAVE file and use the "*audioread*" function to separate the file in a matrix of size **m** samples and **n** channels (variable ***y***) and sampling rate (***Fs***).

```
% Wave File location
path_file = 'D:\example\of\path_to_data_file\DATA0024.WAV';
% Getting Sampling Frequency Fs and Y datapoints from WAVE file
[y, Fs] = audioread(path_file);
```

Then, to isolate each channel, just assign each column of the matrix ***y*** to a variable, representing the source of the data (***acc49_x, acc49_y, acc49_z, hyx, acc50_x, acc50_y, acc50_z***). Note that these values are normalized (between -1 and 1), and don't directly correspond to any physical quantity.

```
% Separate each channel from *.wav to a variable with source name
acc49_x = y(:,1);
acc49_y = y(:,2);
acc49_z = y(:,3);
hyd     = y(:,4);
acc50_x = y(:,5);
acc50_y = y(:,6);
acc50_z = y(:,7);
```

We have all the 7 channels separated, so we just need to create an array of time which will be common for each channel. To do so, just create an array starting at 0 till the size of the WAV file, representing the time of each sample data point (***t***). The simple way to do this, is to get the size in samples (***Nt***) of any variable using "*length*" function and divide it by sample rate (***Fs***).

```
% Create a time array [0, Ts, 2Ts, 3Ts,... (Nt-1)Ts]
Nt = length(hyd);           %NOTE: Could be obtained from another variable
t = (0:Nt-1)/Fs;
```

Now we have all the information needed to represent a waveform, the X axis time and Y axis datapoints. In the example above, we create a figure and use the "*plot*" function to show the waveform of accelerometer #49 (Figure 43).

```
% Create a simple waveform from a single channel
figure('Name', 'Acc#49 X axis Normalized Waveform')
plot(t, acc49_x)
grid on
axis tight
title('Acc#49 X')
xlabel('time(s)')
ylabel('normalized amplitude')
```

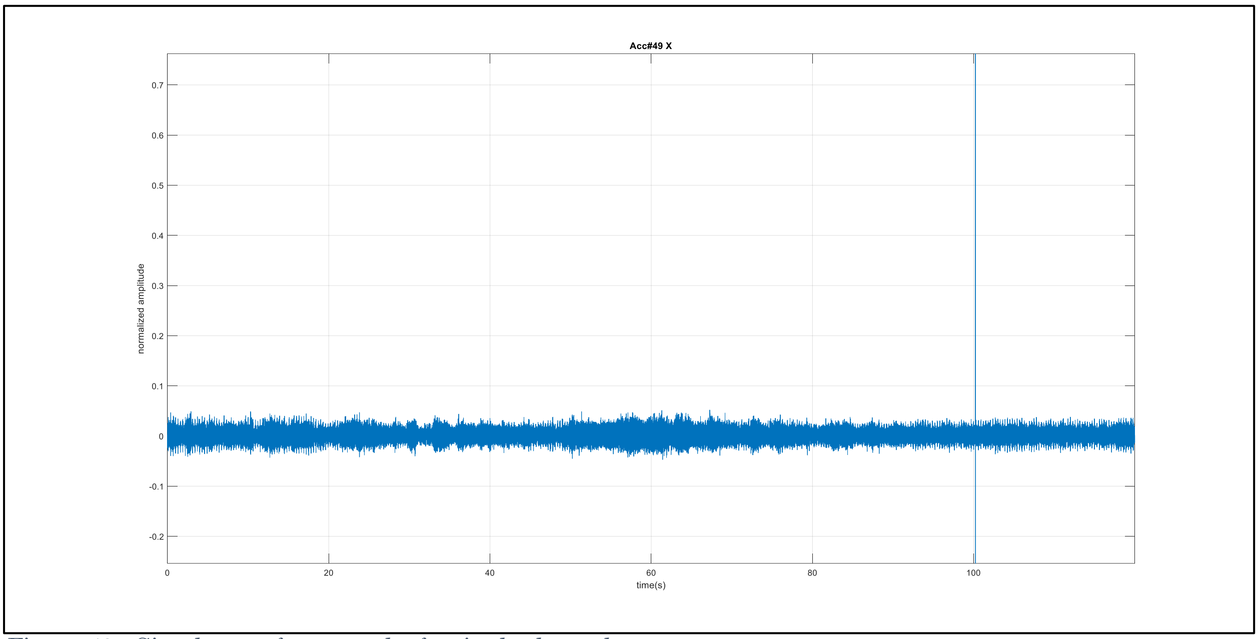


Figure 43 - Simple waveform graph of a single channel

It´s also possible to overlap waveforms in the same figure. The following code create a graph (Figure 44) with the data from accelerometer #49 overlapped.

```
% Create waveforms from accelerometer tri-channel data
figure('Name', 'Accelerometer #49 Normalized Waveforms')
plot(t, acc49_x, 'r', t, acc49_y, 'g--', t, acc49_z, 'b:' )
grid on
axis tight
legend
title('Acc#49 Waveforms')
xlabel('time(s)')
ylabel('normalized amplitude')
```

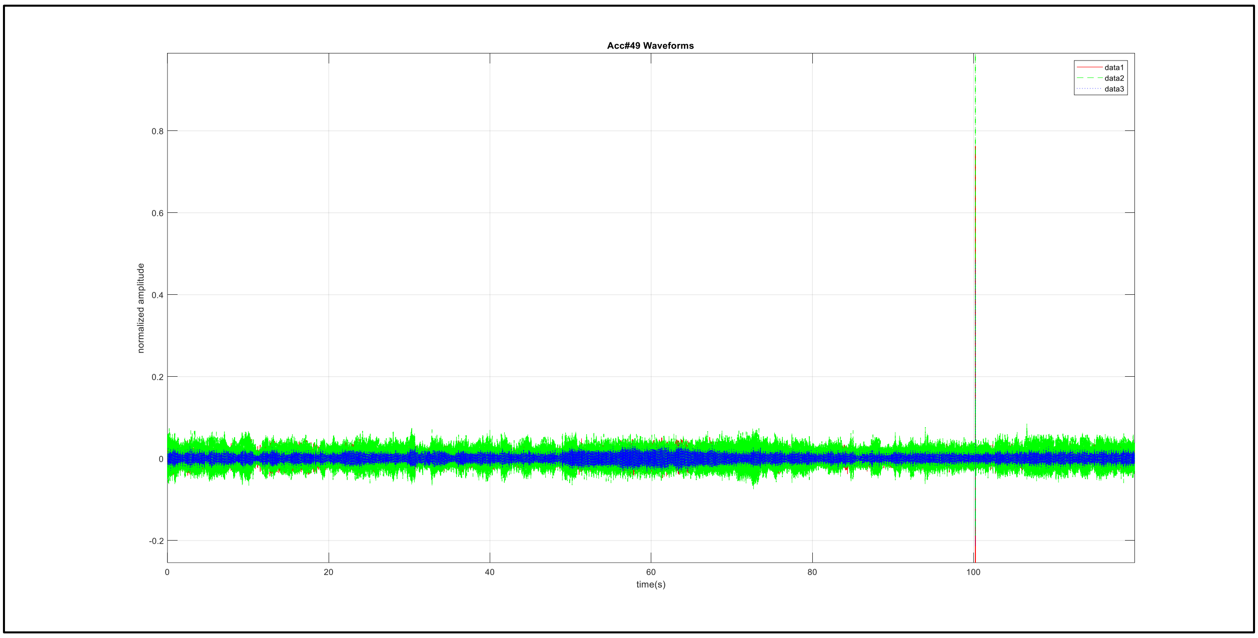


Figure 44 - Graph of the three #49 accelerometer axis data overlapped

Another visualization option is to generate individual graphs for each channel and show all in a single figure. Figure 45 shows all the accelerometer axis data in a single figure with 6 graphs. The complete source code is in the appendix A.3.

```
% Raw waveforms from accelerometer
figure('Name', 'Raw Accelerometer Measured Data')
subplot(2,3,1)
plot(t, acc49_x)
axis([0 ((Nt-1)/Fs) -1 1])
grid on
title('Acc#49 X axis')
xlabel('time(s)')
ylabel('normalized amplitude')
fprintf('.')
subplot(2,3,2)
plot(t, acc49_y)
axis([0 ((Nt-1)/Fs) -1 1])
grid on
title('Acc#49 Y axis')
xlabel('time(s)')
ylabel('normalized amplitude')
 …                              (check appendix for complete code)
```

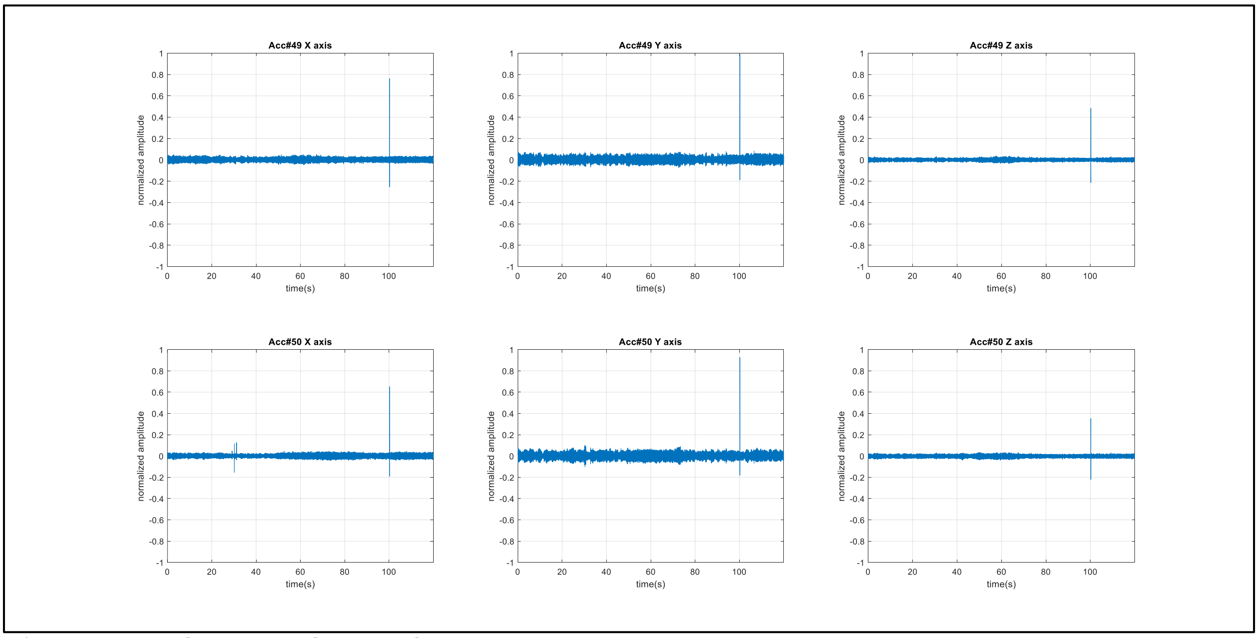


Figure 45 - Accelerometers data graph

To reproduce an audio file the "*audioplayer*" function can be used. Just define the beginning (***start_seconds***) and the end time (***stop_seconds***) where you want to listen the audio file. Then create an audio player object (***player***) with the source file (***acc49_x*** in our example) and sample rate of file (***Fs***). Use the "*play*" function to hear the ***player*** object content in the specified times and the "*stop*" function to stop playback. You can also listen to the complete file by running "*play(**player**)*".

```
% Audio play using audioplayer
start_seconds = 49;
stop_seconds = 63;
player = audioplayer(acc49_x, Fs);
% Play audio between 2 points>>to convert to seconds just multiply by Sample
Rate
```

```
play(player,  [player.SampleRate  *  start_seconds  ,  player.SampleRate  *
stop_seconds ])
% Hear the complete file
%play(player)
% Stop the audio reproduction
%stop(player)
```

4.2.2.2 Converting WAVE data into acceleration and sound pressure level

The WAVE values are normalized between ±1 and represent only a relative amplitude value. To convert these raw values into a physical quantity, we need to consider the modifications of the signal, when passing through the electronic components. In DAVS case, the accelerometer and hydrophone have a distinct path, so for each source we use a slightly different formula.

For the hydrophone, we can obtain the sound pressure $SP\ [\mu Pa]$ from the WAVE normalized data $Norm_{hyd}$, if we multiply the normalized data by the ADC sensitivity (fixed value 5) and divide it by the gains of preamplifier (39), PGA ($PGA_{hyd_{gain}}$, value selected by user) and the hydrophone sensitivity (fixed -195 dB, converted to linear unit):

$$SP\ [\mu Pa] = Norm_{hyd} * \frac{5}{10^{\frac{-195}{20}} * 39 * PGA_{hyd_{gain}}} \qquad (3)$$

Since sound pressure is commonly expressed in logarithmic units, relatives to a reference pressure ($1\ \mu Pa$), the so-called sound pressure level (SPL), the SP value can be converted into SPL using:

$$SPL\ [dB\ re\ \mu Pa] = 20 * \log\left(\frac{SP\ [\mu Pa]}{1\ \mu Pa}\right) \qquad (4)$$

For the accelerometer signal, we can obtain the acceleration $A\ [m/s^2]$ based on accelerometer sensitivity (fixed value of 51 mV/(m/s^2)), the selected PGA gain and the ADC sensitivity (value 5):

$$A\left[\frac{m}{s^2}\right] = Norm_{acc} * \frac{5}{51 * 10^{-3} * PGA_{acc_{gain}}} \qquad (5)$$

The WAVE files generated by DAVS have some information's on the header, namely the gains used for accelerometer and hydrophone. A simple parsing of this header allows us to obtain that information and automate the conversion process. With all the gains and settings as a variable, just apply the conversion formulas, obtaining the graphs from Figure 46. Please check the appendix A.3 for the complete code with the header parsing.

```
%Applying conversion formulas from chapter 4.2.2
acc49_x_a = acc49_x * (adc_sensitivity / (acc_sensitivity*10^-3 *
acc_pga_gain ) );
acc49_y_a = acc49_y * (adc_sensitivity / (acc_sensitivity*10^-3 *
acc_pga_gain ) );
acc49_z_a = acc49_z * (adc_sensitivity / (acc_sensitivity*10^-3 *
acc_pga_gain ) );
sp        = hyd * adc_sensitivity / [ 10^(hyd_sensitivity_db/20)*
hyd_preamp_gain * hyd_pga_gain ];
spl       = 20 * log10(sp) ;
acc50_x_a = acc50_x * (adc_sensitivity / (acc_sensitivity*10^-3 *
acc_pga_gain ) );
acc50_y_a = acc50_y * (adc_sensitivity / (acc_sensitivity*10^-3 *
acc_pga_gain ) );
acc50_z_a = acc50_z * (adc_sensitivity / (acc_sensitivity*10^-3 *
acc_pga_gain ) );
```

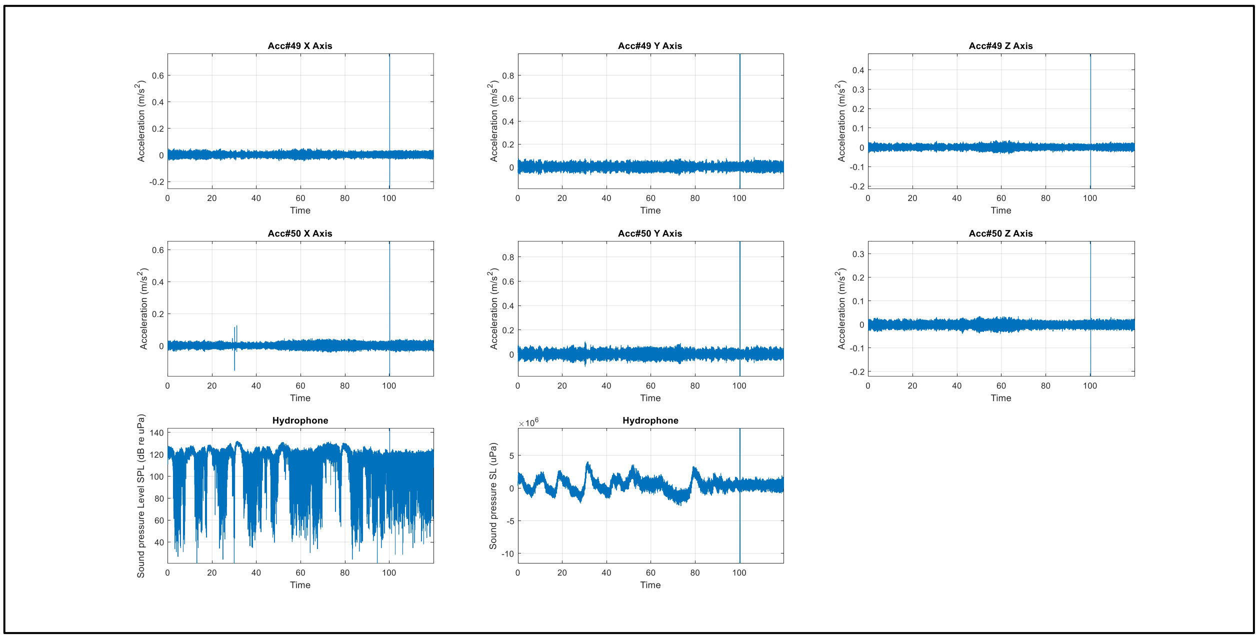


Figure 46 - Converted values. Acceleration for accelerometer channels, SP and SPL for hydrophone channel

4.2.2.3 Viewing the spectrogram of a channel

To generate a spectrogram in **Matlab** we can use the "*spectrogram*" function, from the Signal Processing Toolbox. In **Matlab**, we must specify some settings for the FFT algorithm to work. The function has 6 inputs:

```
spectrogram(x,window,noverlap,nfft,fs,f_axis)
```

which are:

- *x*, the original signal samples array;
- *window*, the type of window and number of divisions done in the signal. The default window type is Hanning (check appendix **Erro! A origem da referência não foi encontrada.** for a simple windows explanation);
- *noverlap*, the number of samples overlapped between adjacent windows processing;
- *nfft*, number of samples used by the FFT algorithm;
- *fs*, the signal sampling rate;
- *f_axis*, the chosen axis to represent frequency (x or y).

Figure 48 shows a figure of the six spectrogram accelerometer data, using a Hanning window with 1024 divisions, 1024 *nfft* and an overlap of 228 samples. A sample of the code is above.

```
% Raw spectrogram of recorded data
figure('Name', 'Raw Accelerometer Measured Data')
axis([0 ((Nt-1)/Fs) 0 Fs/2])
colormap(jet)
subplot(2,3,1)
spectrogram(acc49_x, Window_size, Overlap_size, Window_size, Fs, 'yaxis');
title('Acc#49 X axis Accelerometer')
fprintf('.')
subplot(2,3,2)
spectrogram(acc49_y, Window_size, Overlap_size, Window_size, Fs, 'yaxis');
title('Acc#49 Y axis Accelerometer')
fprintf('.')
…                              (check appendix for complete code)
```

Matlab has some other nice features, as the possibility to see a spectrogram in 3 dimensions. It´s also possible to manipulate (pan, zoom and rotate) the graph, using the graph tools or to annotate or get any point values (Figure 47). An example code is shown above, creating a spectrogram with a default 3D view (Figure 49).

```
% 3D Spectrogram view from hydrophone
figure('Name', 'Raw Hydrophone Measured Data')
axis([0 ((Nt-1)/Fs) 0 Fs/2])
spectrogram(hyd, Window_size, Overlap_size, Window_size, Fs, 'yaxis');
colormap(jet)
view(-45,70)
title('Hydrophone Data')
```

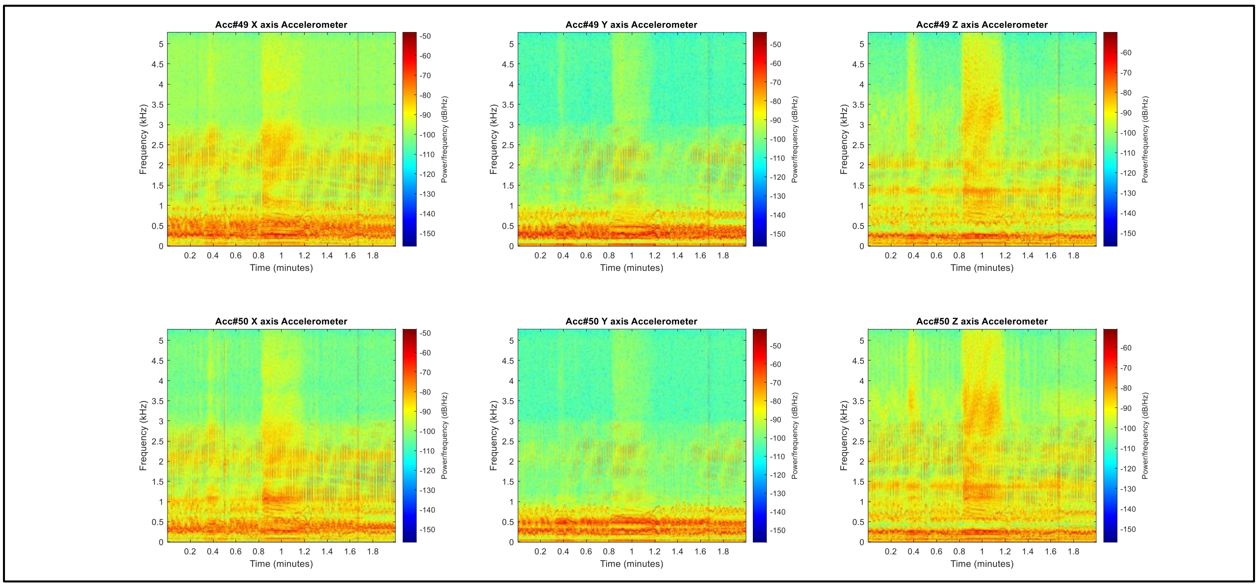


Figure 48 - Spectrogram figure for each of the 6 accelerometers channel

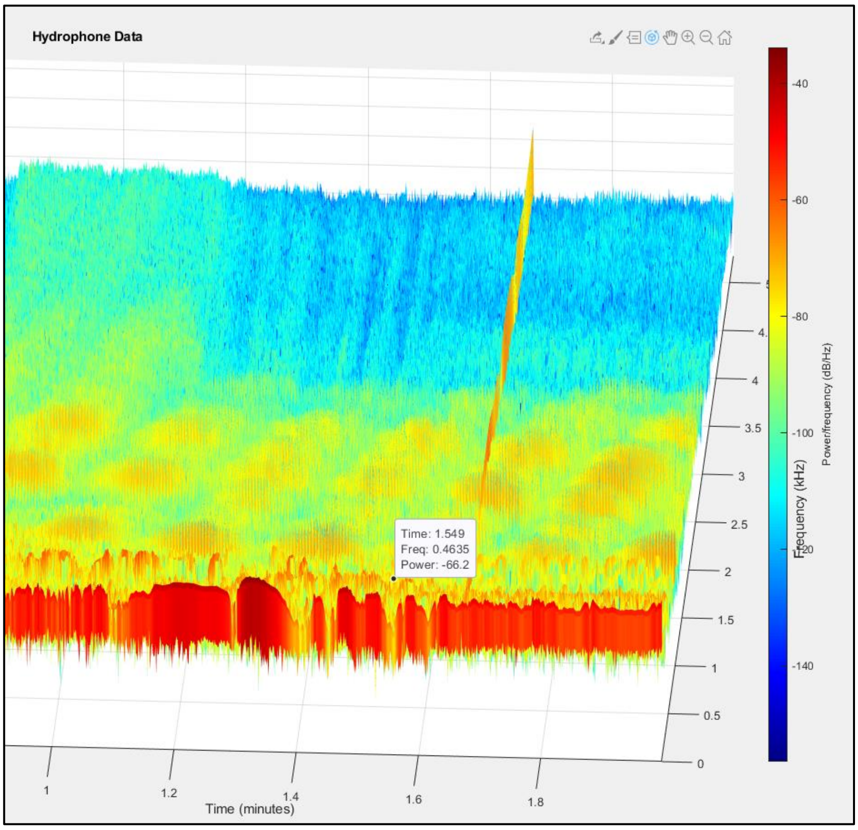


Figure 47 - Example of the spectrogram graph rotated and with a data point annotated

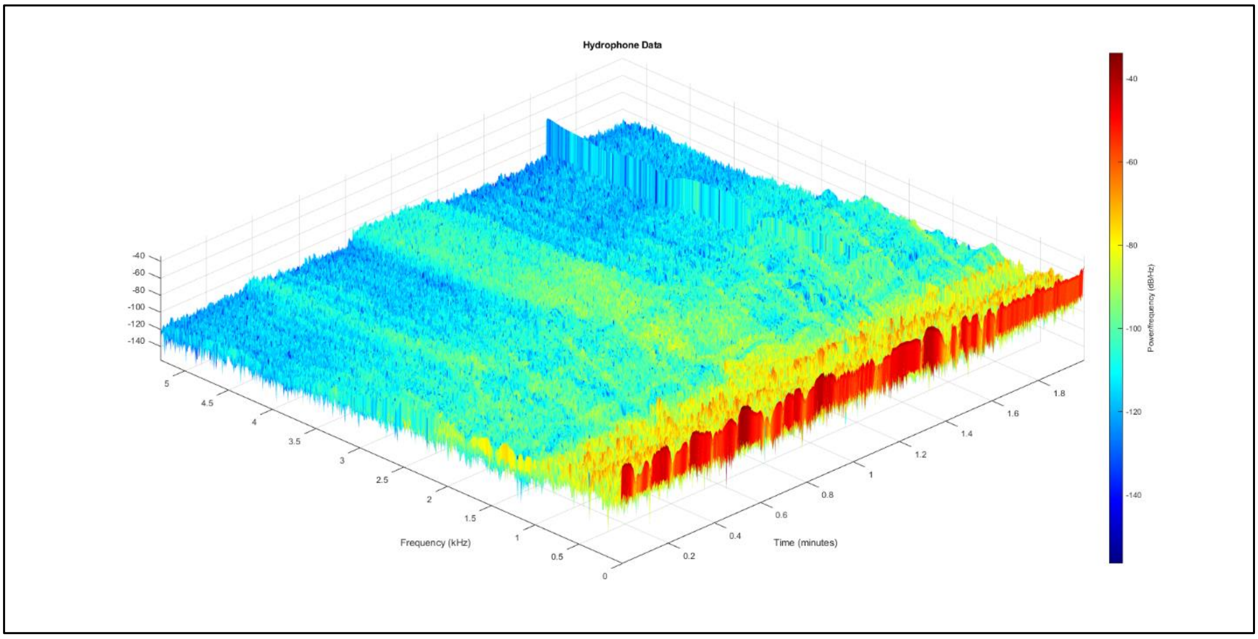


Figure 49 - Spectrogram in 3 dimensions, default view

4.2.2.4 Frequency analysis of a file

To analyse the frequencies from some part of the waveform, we need to use the Fast Fourier Transform again. However, in **Matlab** we need to create all the arrays and information's needed for the algorithm and visualization to work. We will use one simple example and without getting into much detail about how the algorithm works. We will analyse the same file and zone as in Audacity examples, being between 45.8 and 45.9 seconds of AX49 accelerometer data. First, we convert the start and finish time of our interest zone into sample indexes, creating then the corresponding waveform (***partial_processing***). We also create a time array for this waveform (***partial_processing_t***).

```
%Processing example, getting Acc49_x from 45.8 to 45.9 seconds and FFT it
start_f_analysis = 45.8;
end_f_analysis = 45.9;
start_samples = round( start_f_analysis / (1/Fs) );          %second to sample
end_samples = round( end_f_analysis / (1/Fs) );
fprintf('\nFrequency Analysis. Analysing file between %f seconds and %f
seconds.\nCalculating FFT ...', start_f_analysis, end_f_analysis)

%Analysis zone waveform
partial_processing = acc49_x(start_samples:end_samples);
partial_processing_t = (0:1/Fs:(length(partial_processing)-1)/Fs);
```

To get the FFT size, we need to find the next power of 2 value, from the waveform number of samples value. In our example, the interest zone has 1055 samples (***partial_processing_size***), being the next power of 2 the value 2048 (***partial_processing_n***). With this 2 values we can calculate the FFT, storing it in ***partial_processing_y***. This operation will generate 2048 points, however we only use the first half of them (the first 1024 points) which have the frequency information's we need (the second half is symmetric from the first half, so don't have new information).

```
%Calculating FFT
partial_processing_size = length(partial_processing);
partial_processing_n = pow2(nextpow2(partial_processing_size));%FF transform
length
partial_processing_y = fft(partial_processing, partial_processing_n);
```

To visualize the spectrum we need to create the frequency axis information (*partial_processing_f*) which will be shown on X axis of the graph. Each point created by FFT will correspond to a ***Fs/FFT_size*** frequency step. In our case the sampling frequency ***Fs*** is 10547 Hz and the FFT size is 2048, resulting in a frequency step of 5.149902344 Hz. So, the first FFT point is at 0 Hz (0*5.1499), the second FFT point will be at 5.1499 Hz (1*5.1499), the third at 10.2998 Hz (2*5.1499) and the last one at 10541.8453 Hz (2047*5.1499). To represent the Y axis FFT points it's needed to get the power values from FFT (***partial_processing_y***) divided by FFT size (***partial_processing_n***).

```
%Frequencies array
partial_processing_f = (0:partial_processing_n-1) * (Fs / partial_processing_n);
%Power spectrum
partial_processing_power= abs(partial_processing_y) .^2 / partial_processing_n );
```

Now we can create a graph, using ***partial_processing_f*** as the X axis and ***partial_processing_power*** as the Y axis. However, we will just use the first half of the data.

```
%Plot spectrum
plot(partial_processing_f(1:floor(partial_processing_n/2)) , partial_processing_power(1:floor(partial_processing_n/2)));
```

Figure 50 shows the selected area waveform from **acc49_x** accelerometer data (45.8 and 45.9 seconds) on the left side, the spectrogram from that area on the upper right side and the frequency analysis generated by FFT on bottom right side. It can be seen that the amplitude peaks of frequency analysis correspond to the yellow parts of spectrogram, representing frequencies with higher power values.

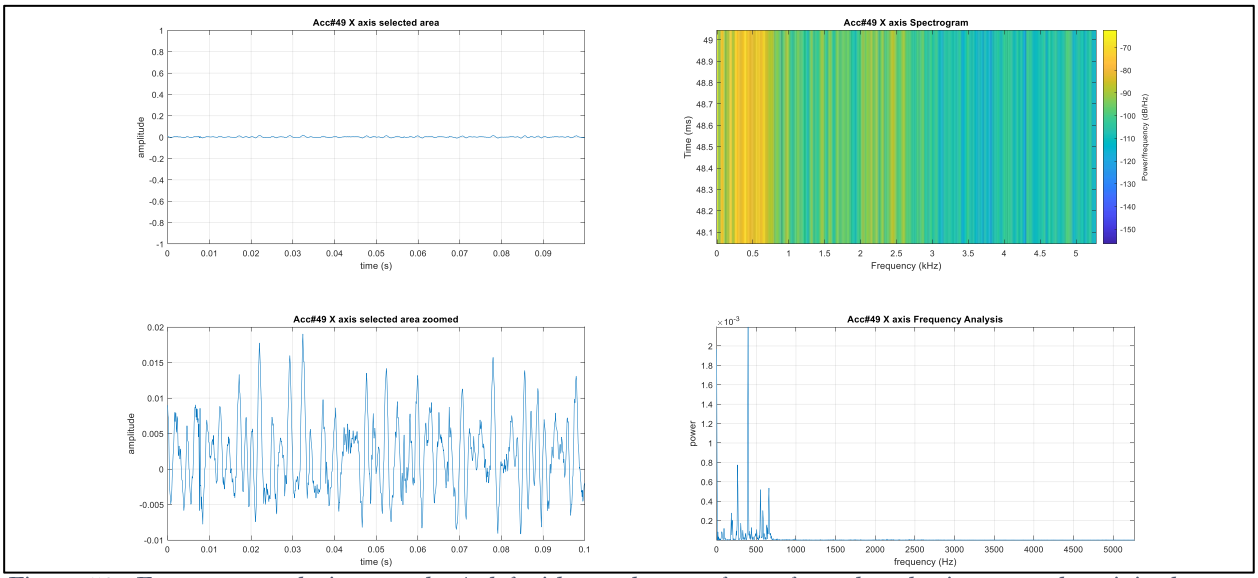


Figure 50 - Frequency analysis example. At left side are the waveforms from the selection zone, the original one on top and a zoomed one in bottom. On top right there's a spectrogram and on down side there's the frequency analysis plot generated

4.2.2.5 Applying a filter to the signal

It's also possible to filter data samples, using any of the **Matlab** available filter functions from Signal Processing Toolbox. As with the **Audacity** example, we will filter **acc49_x** data using a bandpass filter between 1 and 3 kHz. To do this, simply use the "*bandpass*" function specifying the cut off frequencies and storing that filtered data in a variable (***acc49_x_filtered***). In Figure 51 we can see the original waveform and spectrogram from **acc49_x** accelerometer (top graphs), as well as the filtered waveform and spectrogram (bottom graphs). The power reduction outside the band pass zone is clearly seen there.

```
% Define low and high cut frequencies
low_f = 1000;
high_f = 3000;
% filter example, using fixed stepness and attenuation
acc49_x_filtered = bandpass(acc49_x, [low_f high_f])
```

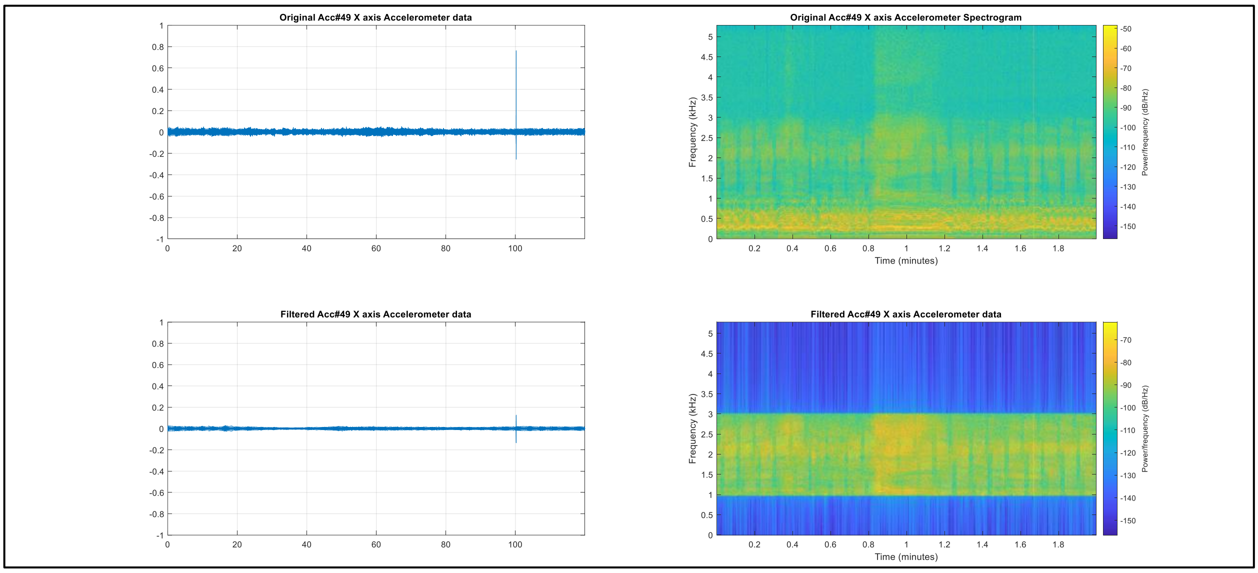


Figure 51 - Original (top) and filtered (bottom) AX49 waveforms and spectrograms

The "*bandpass*" function can also be used to generate a filter behaviour graph, with the original and filtered waveform and a frequency response graph, as shown in Figure 52. A frequency response graph shows what the filter does to the original spectrum of the signal.

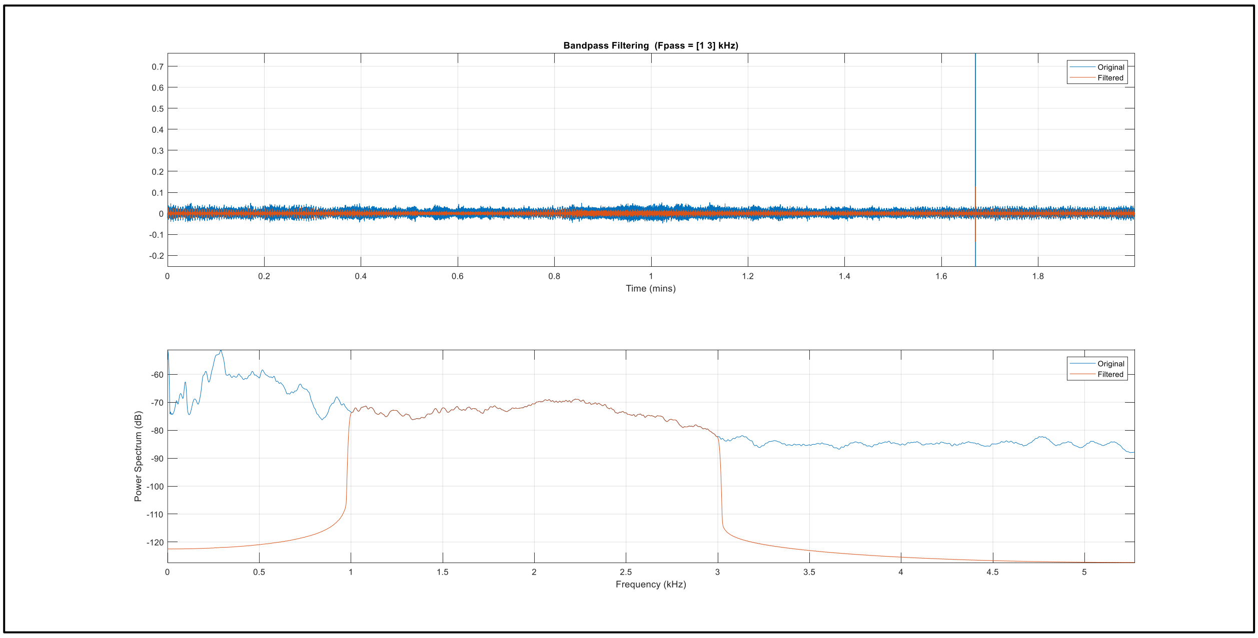


Figure 52 - Original and filtered data graphs, using bandpass function. Top graph shows the waveforms and bottom graph shows the frequency response of the filter

4.3 DAVS Maintenance

The DAVS is a low maintenance device. It should be cleaned with fresh water after deployment in salty water. However, before each deployment and when there is a need to open the DAVS, the sealing o-rings should be replaced and greased with o-ring silicon grease.

To reduce humidity inside the device it is also recommended to place silica desiccant packages inside the container and close it in a low humidity environment, for example in a lab with air conditioning [ ].

4.4 DAVS storage

For long term storage of DAVS it should be completely dry and can be left in the storage box with some silica gel bags.

Chapter 5 Bibliography

[1] A. MANTOUKA, P. FELISBERTO, P. SANTOS, F. ZABEL, M. SALEIRO, S.M. JESUS and L. SEBASTIÃO, "Development and testing of a Dual Accelerometer Vector Sensor for AUV acoustic surveys", Sensors, vol 17(6), pp.1328, June 2017 (https://doi.org/10.3390/s17061328).

[2] F. ZABEL, P. SANTOS, P. FELISBERTO and S.M. JESUS, "Dual Accelerometer Vector Sensor (DAVS): design, build and test", Rep. 03/18, SiPLAB Report, University of Algarve, April 2018.

[3] PCB 356A17 accelerometer datasheet, Available at: https://www.pcb.com/products?model=356A17 (accessed on 09/03/2020).

[4] TI ADS1278 Datasheet, Available at: http://www.ti.com/product/ADS1278/technicaldocuments (accessed on 09/03/2020).

[5] Microchip, K. Bekken, "Differential and single ended ADC" whitepaper. Available at: http://ww1.microchip.com/downloads/en/DeviceDoc/Differential-and-Single-Ended-ADC-WhitePaper-DS00003197A.pdf (accessed on 09/03/2020).

Appendix

A.1 DAVS detailed description

A.1.1 Overview

The DAVS is an acquisition system for underwater acoustics, composed of several components:

- The mechanical parts, like the casings, cables and connectors;
- The acoustic sensors (transducers), responsible for translating a physical quantity into an electric quantity;
- The electronics like the microcontroller or the signal conditioners;
- The firmware used to manage all the system.

The following chapters will present all of that components with some detail. Table 8 resume the characteristics of the DAVS system.

Characteristic	Description
Device type	Vector sensor with Autonomous acquisition and power system
Acquisition Modes	Autonomous (with batteries) or Cabled (with external power source)
Sensing Elements	2 x Accelerometer, 1x Hydrophone
Bandwidth	120 Hz – 4 kHz
Accelerometers	PCB Piezotronics 356A17 (2 x), IEPE type power source. Sensitivity 500 mV/g (50 mV/(m/s^2))
Hydrophone	Custom made, Cylindrical PZT element. Sensitivity -195 dB re uPa/V
Microcontroller	NXP LPC4337 ARM Cortex-M4
ADC Converter	Texas Instruments ADS1278, 7 simultaneous sampling channels, 24 bit Sigma Delta type (10547/52734 SPS)
Motion Sensors	ST INEMO-M1 9 axis IMU (based on LSM303DLHC accelerometer/ magnetometer and L3GD20 gyroscope). Accuracy: ±2deg. Roll and Pitch, ±1deg. Azimuth (with roll and pitch inside ±45 deg. range)
Timing Information	Internal Real Time Clock (accuracy 1s/month) or external host clock
Storage Capacity	2 micro SD slots (Max Capacity 128 GB per slot)
Device External Communication	Ethernet port
Power Sources	18650 Li-Ion battery pack (with 5 x 18650) or external 24 VDC cabled connection
Autonomy	20 hours with 5 x 18650 Li-Ion battery, 3100 mAh
Maximum Allowed Depth	100 meters
Device Size	Length: 525 mm, Diameter: 65 mm
Device Weight	1.4 kg (air)

Table 8 - DAVS characteristics

A.1.2 Mechanical Description

The DAVS system is composed of some mechanical parts, as can be seen in the exploded view of Figure 53. These different parts are:

- The DAVS nose which is the acoustic sensor, where the accelerometers and hydrophone are enclosed (yellow color in Figure 53);
- The casing tube, a white Delrin cylinder where the electronic components fit inside (white color);
- The end cap that close the device (red color);
- Several electronic boards (PCBs, dark green color);
- An optional battery pack (light green color).

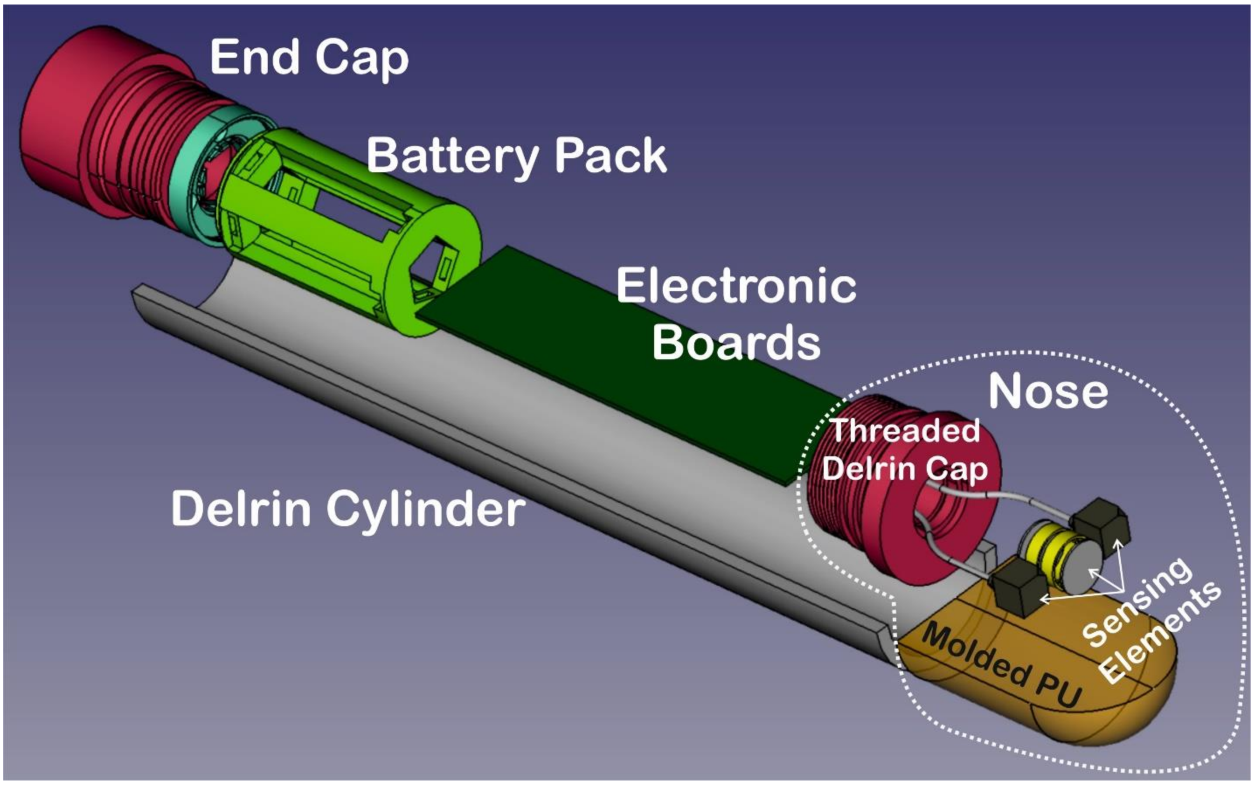


Figure 53 - Exploded view of DAVS system, with the components identified

The acoustic sensor is a black PolyUrethane (PU) solid block, with the sensing elements and cablings molded inside and a threaded white Delrin cap attached, which will connect to the casing tube. The drawing of Figure 54 shows the dimensions and arrangement of sensors inside the PU block, which are aligned and equally spaced. Note also the orientation of the sensing elements, where the Z axis cross the two accelerometers, the X axis is pointing to the front of the nose and the Y axis is perpendicular to ZX plane. This is the device coordinate system (Cartesian), centred in the middle of the hydrophone. The

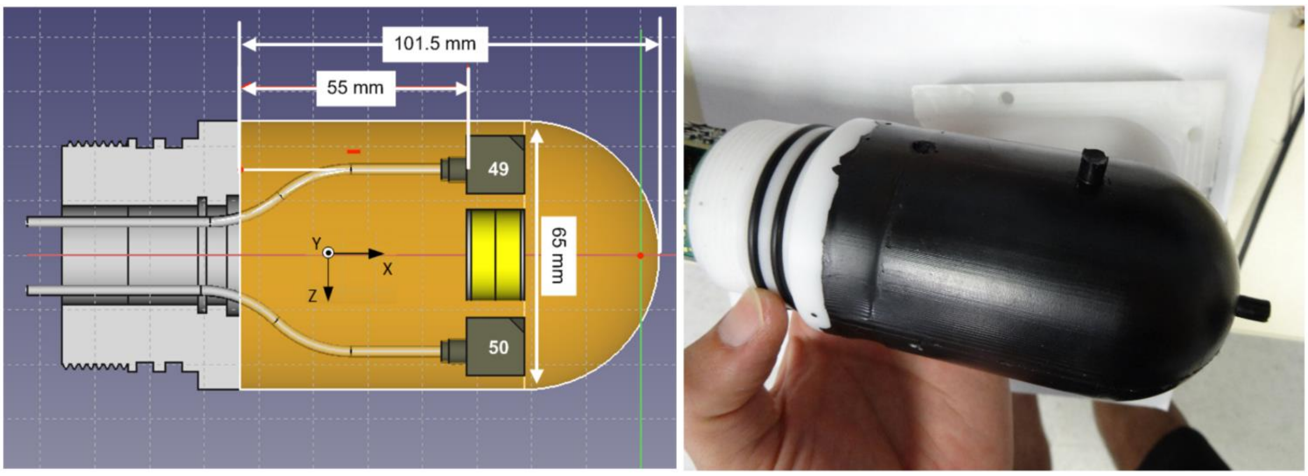


Figure 54 - Internal DAVS nose structure and position (left side) and DAVS nose after PU molding (right side)

accelerometers appear numbered with A49 and A50, which are just the last two number of their serial number. This is an easy way to relate the sensing elements position with the defined coordinate system.

The casing tube is a cylinder made of white Delrin, with an outside diameter of 65mm and a length of 423 mm. This container houses the electronic boards and the battery packs. The casing and the acoustic sensor (nose) measure a total of 525 mm.

To close this cylinder there's a white Delrin end cap (Figure 55). This end cap has an underwater connector, for cabled operation, which permit data transmission and external powering of the system. The connector has a dummy cap which protect it from the water and contacts corrosion. When not in use this dummy cap should be in place. The end cap will provide a waterproof sealing of the cylinder, using some rubber o-rings.

Figure 55 - DAVS end cap with external connector (right side), a connector dummy cap for protection (center) and a locking sleeve (left side)

Figure 56 shows the orientation axis of the DAVS. Note the relative positions of the accelerometers. When deploying at water, the mounted position should be well defined and considering the device axis. There are some marks in the outside of the cylinder, pointing to X (green dashed circle) and Y (yellow circle) directions. The XZ plane follows the mold lines that can easily be seen in the nose.

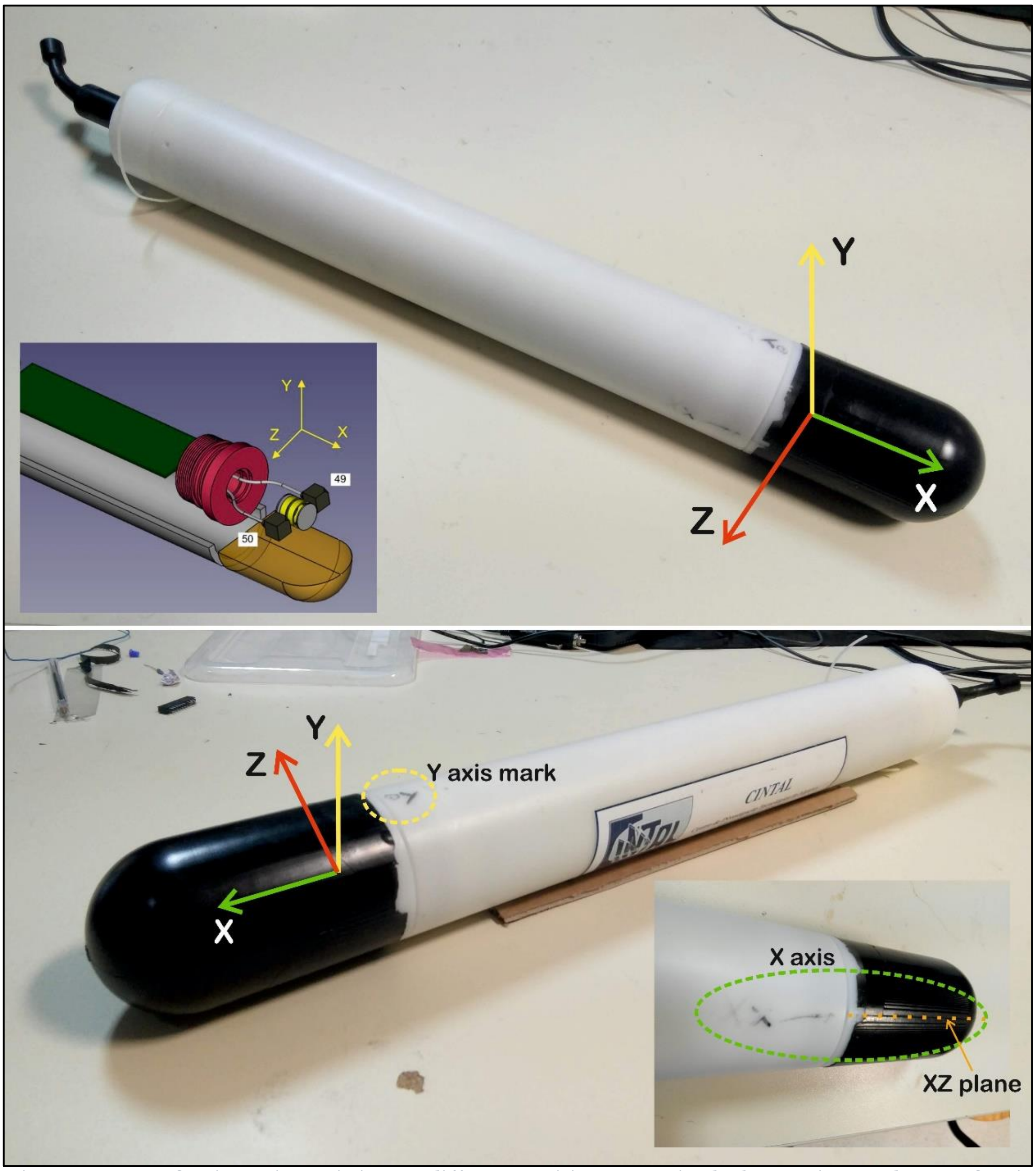


Figure 56 - DAVS orientation axis in two different positions. Note in the bottom image the DAVS axis marks, inside the dashed green circles and the PU mold lines which represent the XZ plane (yellow dashed line)

A.1.3 Electronic hardware description

A.1.3.1 General overview

The DAVS electronic system is an acquisition platform for acoustic sensors data as well as some inertial sensors data. The physical construction of the electronic boards (PCB) must fit inside the cylinder tube, so it´s a stack of several boards with distinct purposes. These different boards are:

- Main board, with the microcontroller, external connectors, clock and power management;
- Analog-Digital converter (ADC) board, where the analog sensor signals are converted to digital;
- Signal conditioning boards (analog frontend with pre-amplifiers and programmable gain amplifiers), to adapt and prepare the analog signal for digital conversion;
- Interconnection hubs, simple PCBs without any electronic, to stack the other boards together. The analog signals pass through them without any changes.

In Figure 57 can be seen the stacked construction, with the distinct PCBs identified.

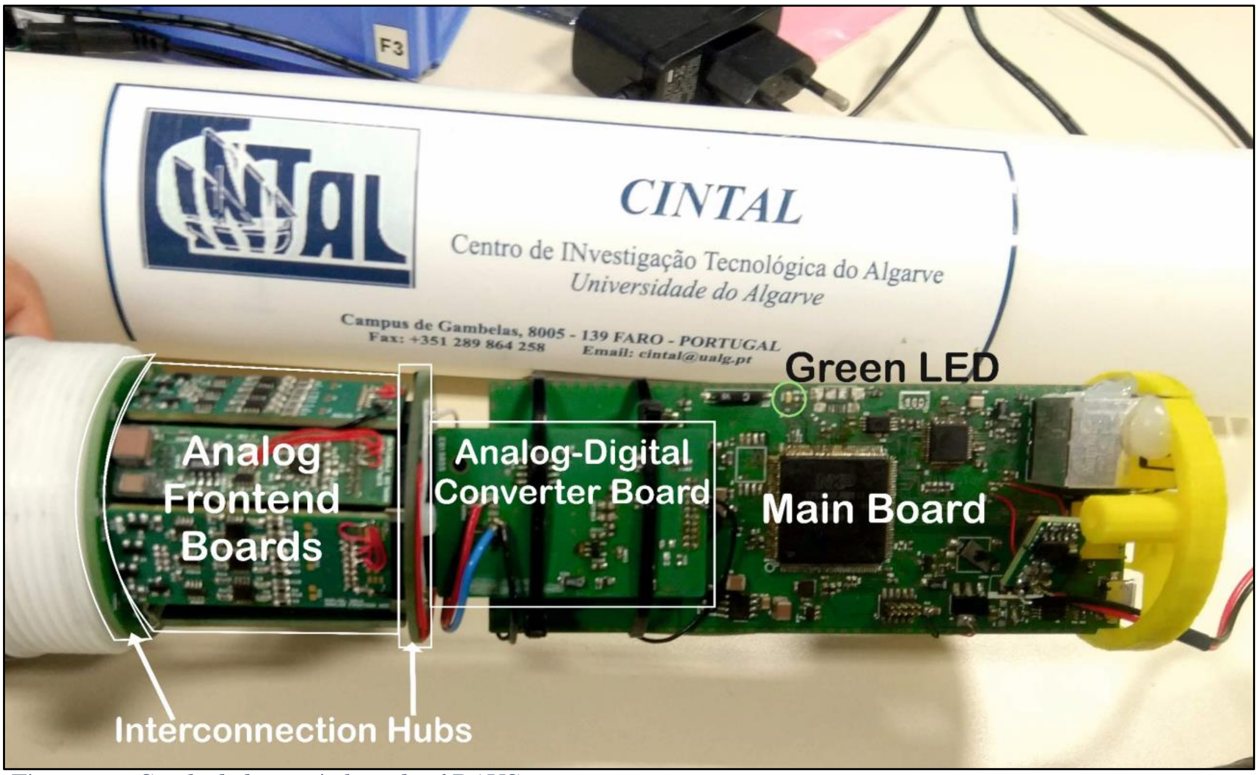


Figure 57 - Stacked electronic boards of DAVS system

The electronic system includes a microcontroller, a flash micro SD storage device, an inertial measurement unit, an analog multi-channel simultaneous acquisition device (the analog to digital converter), some analog frontends to prepare the analog signal to conversion, a real time clock, ethernet external communications and power management. A general overview of the component's interconnection can be seen in Figure 58. In following subchapters there is a brief description of each component.

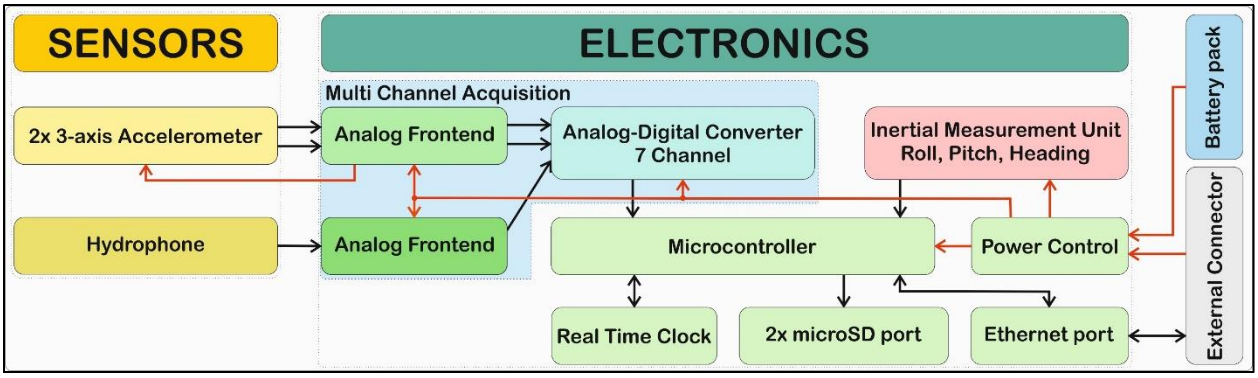


Figure 58 - Electronic components interconnection diagram

A.1.3.2 Acoustic Sensors

For sensing the underwater acoustics, DAVS uses two types of transducers: hydrophone and accelerometer. Each one will be described below.

A.1.3.2.1 Hydrophone

The used hydrophone was custom made by SipLab, using two ceramic rings of PZT-5 grade material (Steminc SMC22D20H6412) with 22 x 20 x 6 mm (Figure 59). This have a cylindrical end-cap design, with an estimated sensitivity of -195 dB re µPa/V, an air resonant frequency around 44 kHz (in water it should be around 35 kHz) and a maximum depth of operation near 400 meters [2].

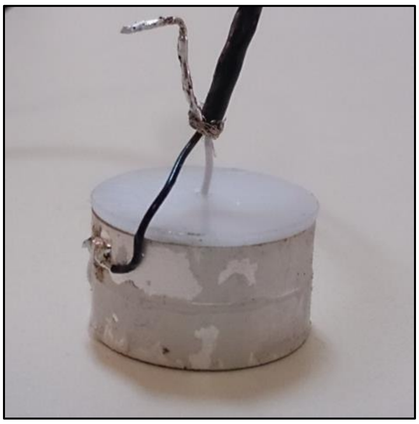


Figure 59 - DAVS custom made hydrophone

A.1.3.2.2 Accelerometer

The DAVS uses two ceramic triaxial accelerometers with shear design from PCB Piezotronics, model 356A17. This model has a frequency range of 0,5 to 4 kHz (at 10% sensitivity deviation), with 51 mV/(m/s^2) of sensitivity (500 mV/g) and a measurement range of ±98m/s^2 (± 10g) [3]. It can work between -54 and 80 ºC, however with some deviation in the sensitivity between extremes. The power supply needs to respect Integrated Electronics Piezo Electric (IEPE) standard, called IPC (Integrated Circuit Piezoelectric) in PCB Piezoelectric equipment. This IPC refers to some internal electronic circuits, whose function is to convert high impedance charge output of the piezo element into a low impedance voltage signal. This internal circuit requires a constant current power source (usually between 2 and 20 mA), supplied over a simple two-wire cable with the signal and power carried on a single wire, plus a ground wire. The ceramic sensing element is enclosed in a 14x20.3x14 mm aluminium housing, where a ¼-28 4 pin connector is available (Figure 60). Three of these four pins carry the constant current power source and output voltage, for each axis, and the fourth pin is a ground reference. This device can be seen on left side of Figure 60.

In DAVS system, these accelerometers use an IEPE type power source with constant current source of 3mA and 20 VDC. The input voltages for this IEPE source will range 18 to 30 V DC. The output signal will be sent through some signal conditioning circuits, before being sampled and converted to digital.

An important information for these types of accelerometers is it frequency response, which tells how the device sensitivity respond to a specific frequency. The typical curve is shown on the right side of Figure 60, where the sensitivity areas of 5 and 10% are marked. One important device characteristic is the resonant frequency (*fn* in Figure 60), which is stated to be more than 14 kHz for this model. During some test of the DAVS, it was found that when deployed near external sources operating in 18-34 kHz band, the

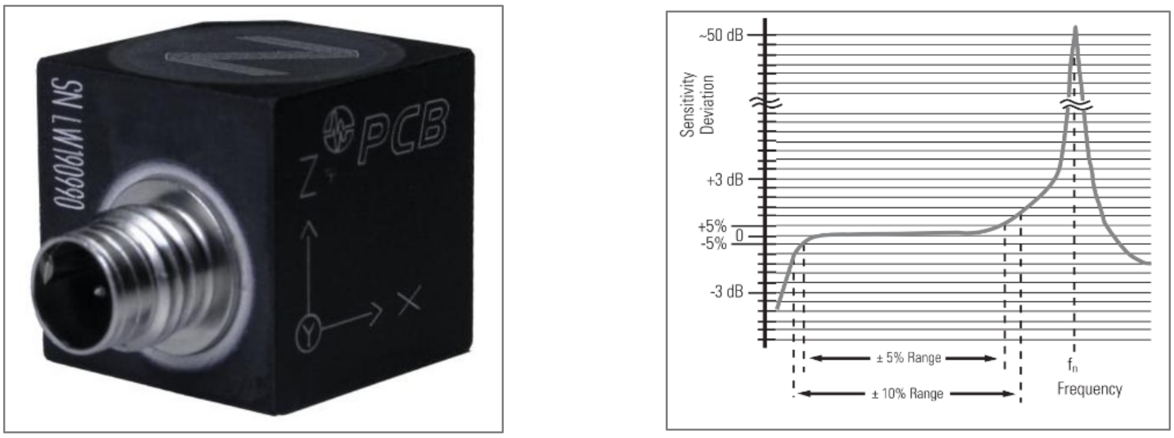


Figure 60 - DAVS PCB 356A17 accelerometer (left side) and typical frequency response of PCB accelerometers, noting the resonant frequency (fn).

signal will saturate on conditioning stages. This was related to the accelerometer resonant frequency behaviour, which increases the device sensitivity, amplifying the internal signal and causing the output saturation on subsequent signal conditioning stages. The output signal should be low pass filtered (LPF) in these cases, however this LPF isn't implemented in the hardware right now.

A.1.4 Electronics – Multi Channel Acquisition

The acquisition system of DAVS contains several parts, mainly the analog to digital converter (ADC) but also some auxiliary components, necessary to conditionate the raw signal from the sensors. Each analog signal from the sensors output will follow a distinctive path through different stages, depending on whether the source is the hydrophone or the accelerometers. Following is a brief description of the two sensors signal paths.

A.1.4.1 Hydrophone signal path

The hydrophone analog signal pass through several stages, as shown in Figure 61. The first stage is a pre-amplifier with a gain of 6 dB, designed to act as a high pass filter with cut off frequency at 120 Hz. This filtering will reduce the platform motion noise and low frequency noise from waves. It allows a maximum peak-to-peak input voltage of 10 V (±5 Volt), which for the sensitivity of -195 dB re µPa/V permit a maximum Sound Pressure Level (SPL) of 209 dB re µPa. The second stage of the hydrophone signal is a Programmable Gain Amplifier (PGA), with adjustable gains of 1x, 2x, 4x, 8x, 16x, 32x and 64x (0dB, 6dB, 12dB, 18dB, 24dB, 30dB and 36dB). This gain is controlled by the microcontroller, through an I2C interface. After leaving the second stage, the signal pass through a single ended to differential converter, to adapt the signal to the differential input of the Analog to Digital (ADC) converter. This stage converts the signal from ±5 Volt to ±2.5 Volt, a reduction of 0.5x (-3 dB). In the ADC, the signal is converted from an analog voltage signal to a 24 bit digital signal (Pulse Code Modulation in two's complement), which will then be send to microcontroller. The microcontroller store these samples to a SD card in WAVE format file or stream them through Ethernet.

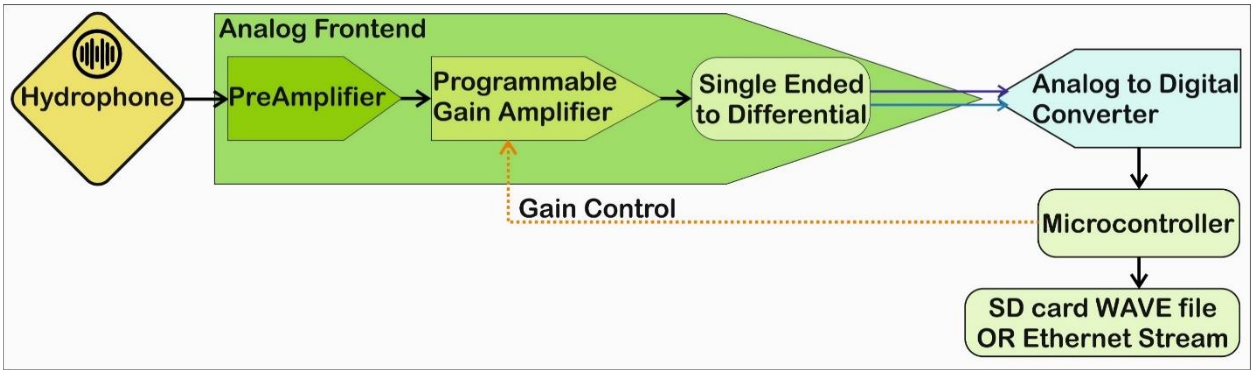


Figure 61 - Hydrophone signal path

A.1.4.2 Accelerometer signal path

The accelerometer requires a power source with constant current (IEPE source) to operate, and only use 1 wire for power supply and signal transmission (plus a common ground reference), for each axis. The output for each axis is connected to this IEPE power source, which bias the signal with a 11 VDC voltage. The raw accelerometer readings will be in the range ±5 Volt, plus that biasing voltage. IEPE source can work with input voltages between 18 and 30 Volt, being that in DAVS case this voltage will normally range between 20 and 24 Volt. This is the first stage of the analog signal, and at the IEPE output the biasing voltage has been removed, being just the ±5 V signal. The Figure 62 represents this stage and can be seen an example signal that pass through the IEPE source.

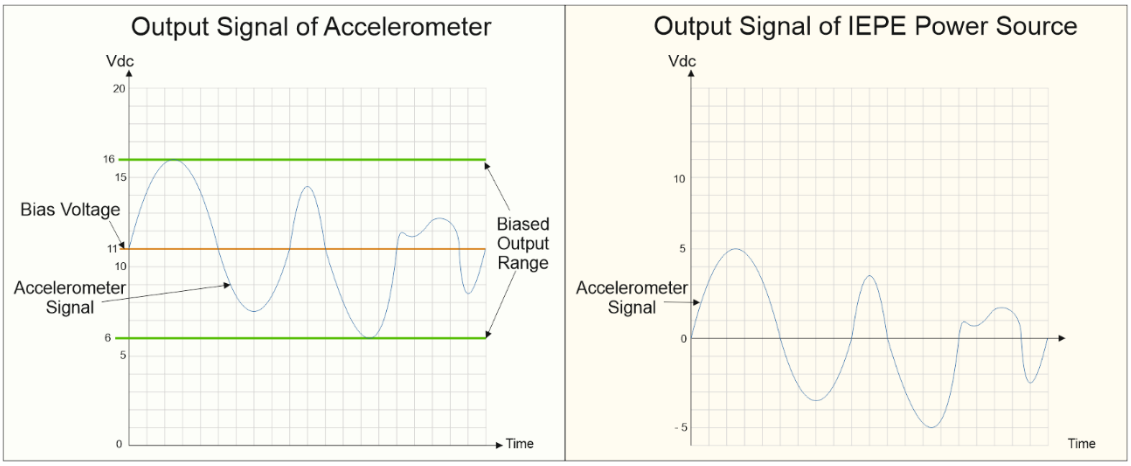


Figure 62 - Signal at the input and output of IEPE power source

The following stages are similar to the ones used for the hydrophone. The second stage is the PGA, with user selected gains (1x, 2x, 4x, 8x, 16x, 32x and 64x), the third stage is the single ended-differential converter and at last is the ADC. The converted signal data will then be stored or streamed by the microcontroller. Figure 63 represents all the stages of the accelerometer signal path.

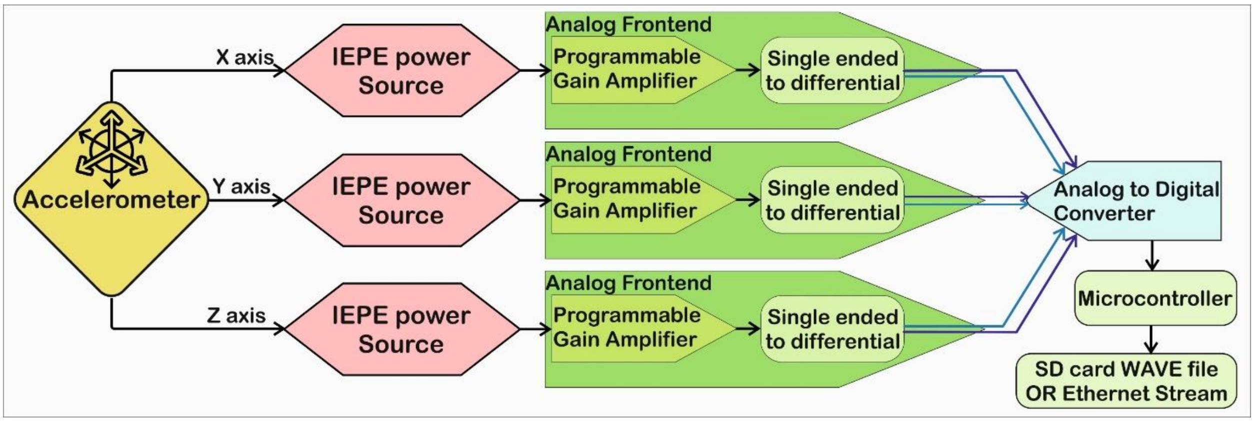


Figure 63 - Accelerometer signal path

A.1.4.3 Analog to Digital converter

The analog to digital converter will translate the analog continuous signal to a discrete digital signal. The ADC chip used by DAVS is a Texas Instruments ADS1278 24 bit sigma-delta type, with 8 simultaneous acquisition channels, working at 27 MHz [4]. In DAVS only 7 channels are used, one for each acoustic source, even if it record all the 8 channels. Each of these ADC channels have a differential input, being this the reason of a single to differential channel adaptation on signal paths. The use of differential inputs has benefits in small signals sampling and noise cancelling properties [5]. The input range of ADC is ± 2.5 Volt, so the input signals should also be in this range. The output is a 24 bit value in two's complement, representing the input voltage in the range ± 2.5 Volts.

This ADC have 4 different operation modes, allowing an optimization of resolution, power or speed:

- Low speed mode, has a sample rate of 10547 samples per second (SPS) and a typical Signal-to-Noise Ratio (SNR) is 107 dB;
- Low power mode, with a SPS of 52734 and SNR of 106 dB;
- High resolution mode, with a SPS of 52734 and SNR of 110 dB;
- High speed mode, with a SPS of 105469 and SNR of 106 dB;

Higher SPS values means more time resolution, which will allow a higher spectral range when performing frequency analysis. A SPS of 10547 will allow a frequency analysis up to 5273 Hz, while a

105469 SPS will allow 52734 Hz. Note that the acoustic sensors frequency ranges (4 KHz for accelerometer and 33 KHz for hydrophone) will limit this latter SPS, not being advantageous to use it. Higher SNR will get more amplitude resolution on sampling process, however it´s not adjustable but a chip characteristic.

This ADC chip is deployed on a specific PCB board with connectors to the other components, thus reducing some interferences from other high-speed interfaces on main microcontroller board. All operations and modes are controlled through pins, connected to the microcontroller via I2C. The sampled data output is transferred through a single pin to the microcontroller, using SPI protocol.

A.1.5 Electronics – Microcontroller

The microcontroller is the heart of the DAVS, being responsible for coordinate all the electronic elements. It´s based on the NXP LPC4337 ARM Cortex-M4 series (Figure 64), which have low power consumption but high performance, being a good choice for applications where data intensive and high speed connectivity are required. It´s main functions are:

- Retrieve the sampled signals from ADC;
- Retrieve the inertial sensor data;
- Store recorded data into the SD card flash storage;
- Setting the acquisition parameter of PGA and ADC;
- Communicate through Ethernet port, for sending real-time data or receive configurations parameters.

The firmware is stored on the internal flash memory of microcontroller and isn't used any operating system for managing the system. Instead, all functions are performed through the hardware pins (like I2C communications), interrupts (ADC informing that there's samples data available for read) and Direct Memory Access (DMA, for writing data to SD card).

Figure 64 - DAVS microcontroller

The sampled data transfer is done using SPI protocol, and occurs when the ADC activate an interrupt pin, telling the microcontroller that there's data available to read. The retrieved sampled data is buffered into the microcontroller RAM memory, in 10 blocks of 1536 bytes. When this buffer is filled, the data is transferred to the SD card, or streamed through Ethernet output, based on DAVS working mode (autonomous or cabled). The format used for this data storage is a regular WAV format, while the streaming data is the normalized samples from ADC. The Ethernet data is sent from the microcontroller to the physical Ethernet controller using the Reduced Media Independent Interface (RMII).

The configurations of ADC and PGAs are done through I2C interface, as well as reading the stored configs from EEPROM memory. User configurations can also be read from SD card flash storage or send to microcontroller through Ethernet. The inertial sensor sends his data to microcontroller through a serial

UART line and can be stored in the header of wave files or streamed. The RTC data is available to microcontroller and can be accessed and updated externally by the user, through Ethernet Interface.

A.1.6 Electronics – Non acoustic positional sensors

For the orientation of the DAVS is used an INEMO-M1 9 axis inertial measurement unit (IMU) from ST, composed of a 3D accelerometer (LSM303DLHC), 3D gyroscope (L3GD20), 3D magnetometer (LSM303DLHC) and a 32 bit STM32F103REY6 microcontroller (Figure 65). The combinations of these three inertial sensors gives us the roll, pitch and heading of the DAVS, and can be used to create a tilt compensated electronic compass. This IMU is a complete system, with microcontroller, multiple communication interfaces (UART, I2C, SPI, USB) and power regulator. It's capable of performing internally all the math and delivering only the required result (like the heading angle, for example). Sensor data is sent from INEMO UART to microcontroller once a second.

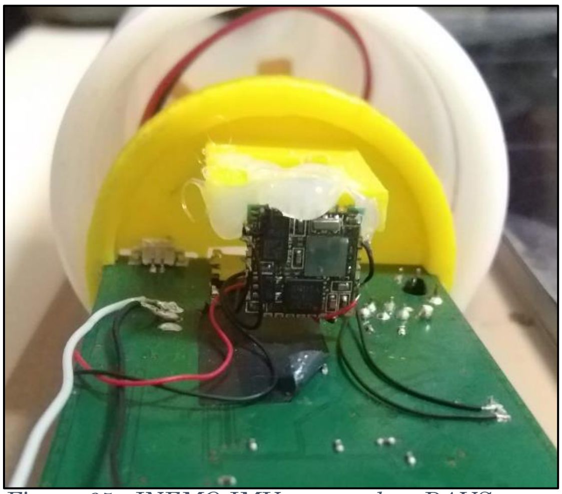
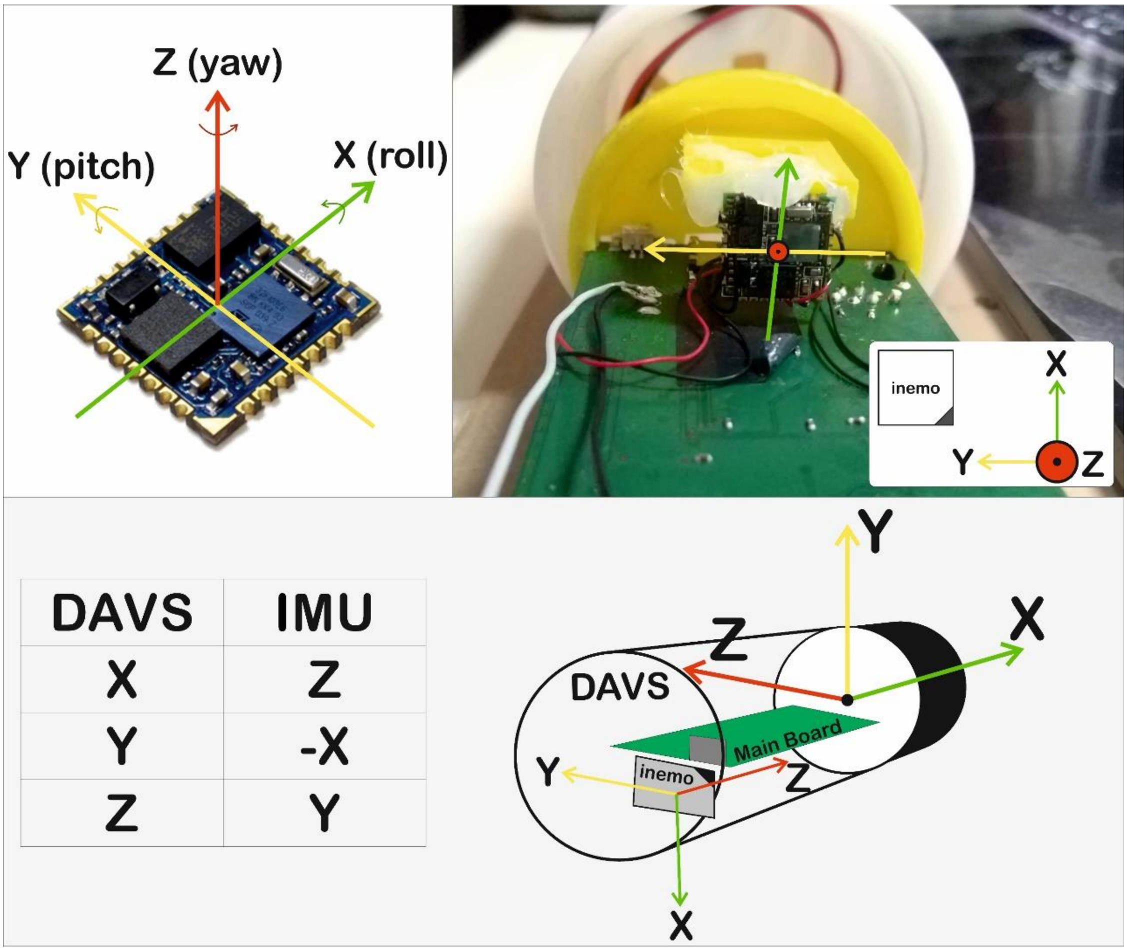


Figure 65 - INEMO IMU mounted on DAVS

In the DAVS system this IMU can used as an electronic compass, to estimate the DAVS position. To do this, it´s important to operate the IMU roll and pitch (X and Y axis) angle in +-45 degrees, for a good precision of headings calculation. Since the roll and pitch ranges +-90 degrees, and the DAVS can be used in a vertical or horizontal position, when headings need to be estimated it's best to mount the DAVS in a vertical position to maintain that ranges. It's possible to change IMU position in DAVS mainboard to use in horizontal position, however it's not advisable to make this change. Figure 66 shows the relative axis positions of DAVS and IMU.



DAVS	IMU
X	Z
Y	-X
Z	Y

Figure 66 - INEMO IMU device and axis (Top left side) and IMU orientation in DAVS system (Top right side). DAVS orientation and IMU correspondence (bottom)

A.1.7 Electronics – Communication ports

The DAVS contains three external communication ports (Figure 67). The most important is a 10/100 Mbps Ethernet port which can be used to configure the device or for data streaming. When streaming data the DAVS should be connected to a 100 Mbps device, due to high transmission data rates. There's an USB port but isn't used. For flashing the microcontroller there's a JTAG port, but the flash proceeding is out of the scope from this text.

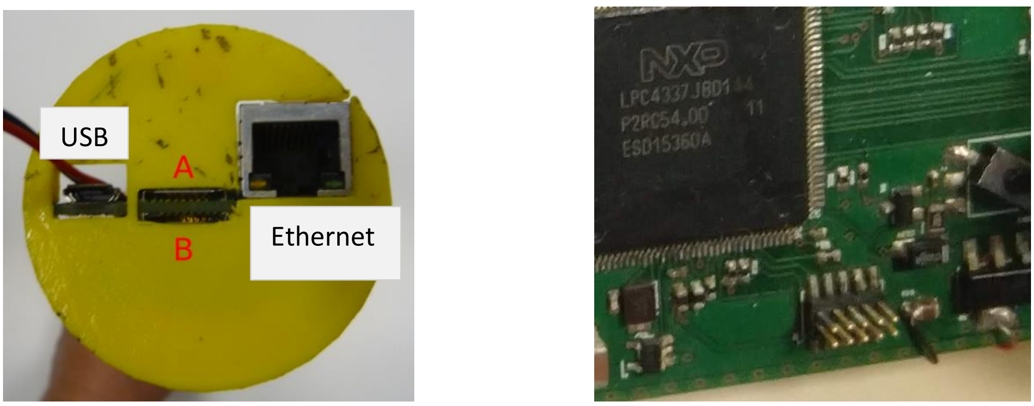


Figure 67 - DAVS communication ports (left side) and JTAG programming port in main board (right side)

A.1.8 Electronics – Real Time Clock

The main board includes a real time clock (RTC), to maintain time and date information, when main power is disconnected. This information is used for time-stamping the acquired data. This RTC uses a CR2032 coin battery (Figure 68) to maintain memory data and have a precision of approximately 1 second per month. The date and time can be adjusted over Ethernet interface.

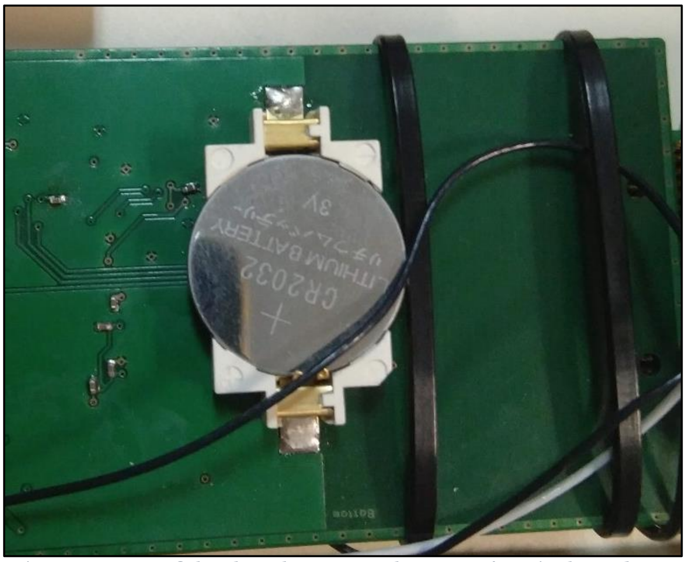


Figure 68 - RTC backup battery at bottom of main board

A.1.9 Electronics – Micro SD Storage

DAVS have 2 micro SD card slots, to store the obtained data (Figure 69). For autonomous operation, at least one SD card need to be used, connected to slot A. If a larger storage capacity is required, two micro SD cards can be used, connected to slots A and B. SD cards used should be from U3 class, allowing a high sampling rate to be stored successfully. Lower classes can fail to write at high SPS, corrupting the file system. The SD card need also be formatted in FAT filesystem.

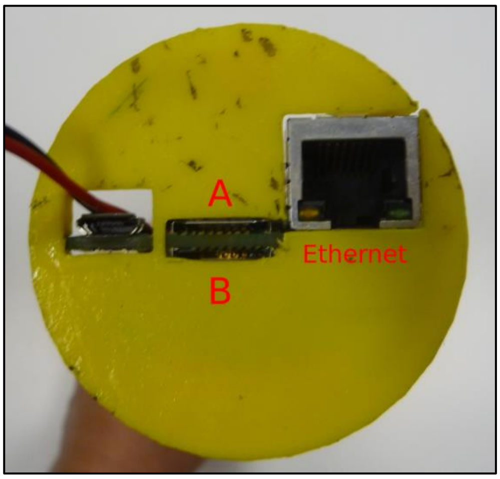


Figure 69 - micro SD card slots A and B

The obtained sensor data is stored in a regular WAVE file format. When starting DAVS in autonomous mode a new directory is created on SD card, following a naming scheme starting as "DATA0000" up to "DATA9999". Inside each folder, there will be several sensor data WAV files, starting with name DATA0000.WAV up to DATA9999.WAV. DAVS will never rewrite over a folder or file if that name already exists, preventing data loss. Instead it will create a new folder/file with the next name following the naming scheme.

For operation in autonomous mode the configuration of DAVS must be done in a text file, placed at the root of the SD card. When starting the DAVS, it will try to access SD card, read that configuration file and adjust the settings. If none SD card is detected, DAVS assume the cabled mode and will start streaming data through Ethernet.

A.1.10 Electronics – Power Control

For operation, DAVS system require 3 different voltage levels (Figure 70):

- 18-30 VDC, to power the accelerometer constant current IEPE circuits and all the DC-DC converters. This voltage is obtained from the power sources which for autonomous or cabled mode will range 20 to 24 Volt;
- ±5 VDC, for the amplifier stages and analog circuits. It's done through a DC-DC converter, which convert the 18-30 V into ±5 V;
- 3.3 VDC for the microcontroller and peripheral electronics. This requires an additional DC-DC converter.

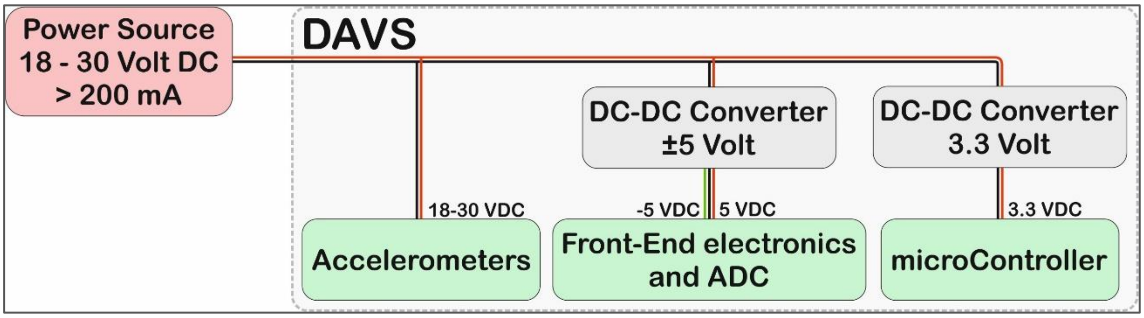


Figure 70 - Energy distribution diagram

The input voltage range will be limited by accelerometer IEPE circuits, being from 18 to 30 VDC. The DAVS power consumption range from 120 mA when storing data to SD card and 200 mA when streaming all the data through Ethernet cable (at 24 VDC).

A.1.11 Electronics – Power Sources

The DAVS can be powered by two options:

- External power source, through the Ethernet cable, for the cabled mode;
- Battery packs for autonomous mode.

The battery pack is a blue cylinder which use five rechargeable Li-Ion 18650 batteries, connected in series to get an approximate voltage of 20.5 V. An autonomy of 20 hours is expected when using five 18650 batteries with 3100 mAh of capacity. This pack connect to DAVS through a XT30 connector, as seen on Figure 71.

When connected through the cable, an external power source of 24 VDC, capable of sourcing more than 200 mA should be used. The input source should be noise free, to minimize the interference in the

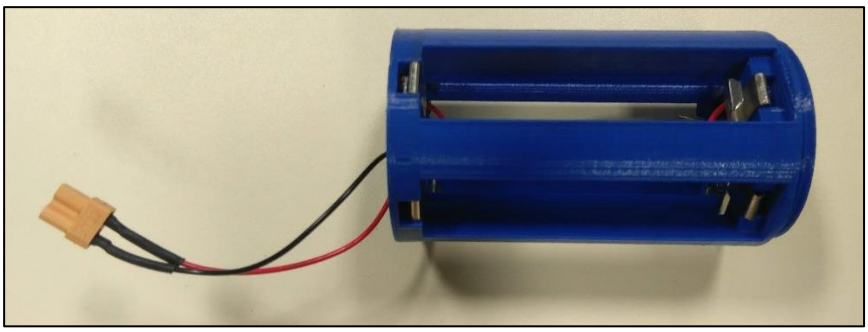


Figure 71 - DAVS 18650 battery pack with XT30 connector

acquisition system. It´s recommended to use two 12 V lead-acid batteries connected in series (24V) or a linear power supply instead of a switching one.

A.1.12 Connectors

DAVS main board have two types of connection:

- Power connector, a XT30 male plug type for battery packs and external sources;
- Ethernet cable for data streaming and configuration.

For the cabled mode, there´s a special MCBH-8-M connector in the external side of the cap, which permit communication and supply power to DAVS, from an external computer or telemetry system. In the inner side of the cap there are two pigtail cables, to connect to main board Ethernet socket and XT30 male connector (Figure 72).

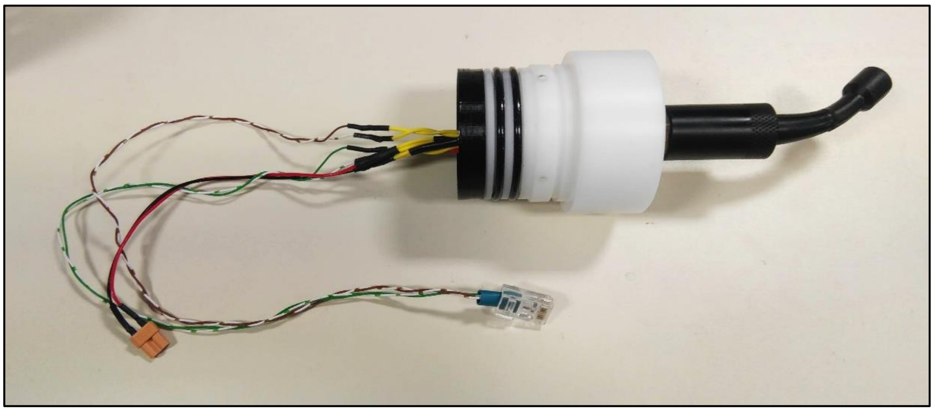


Figure 72 - End cap with the two pigtails: XT30 for power and Ethernet for network streaming

The pinout of the outer connector to the inner connectors is shown in Figure 73 and Table 9. The Ethernet pigtail cable is a simple Ethernet cable (CAT5e) with only two pairs used to connect to DAVS mainboard. Since the Ethernet interface at 100 Mbps only need 2 twisted pairs for operate, the 2 additional pair aren't used. The power pigtail is a simple pair of wires, connecting the XT30 connector.

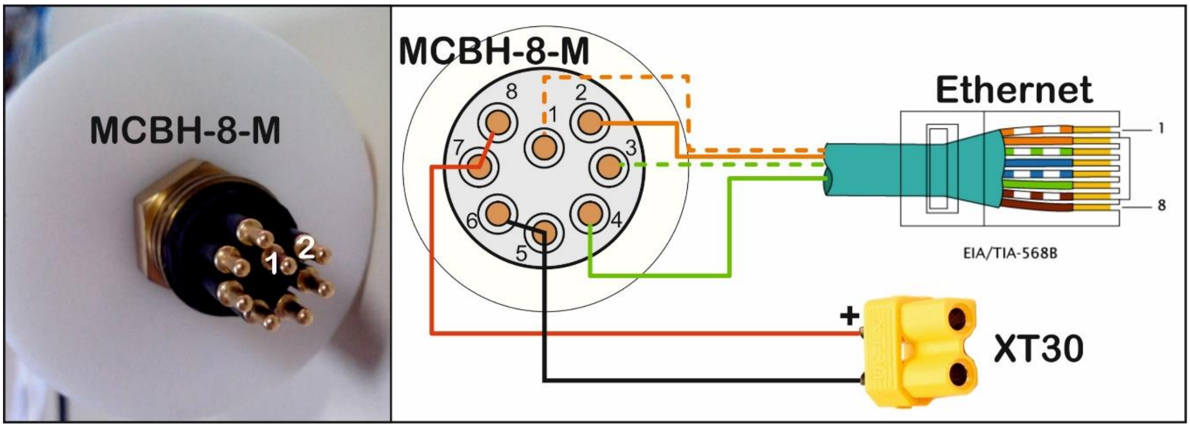


Figure 73 - DAVS External to internal pigtail cables pinout

Underwater Pin	Ethernet Pin	Xt30 Pin	Function
1	1		Ethernet 2+ (orange/white)
2	2		Ethernet 2+ (orange)
3	3		Ethernet 3+ (green/white)
4	6		Ethernet 3+ (green)
5		- (GND)	Power Ground (black)
6		- (GND)	Power Ground (black)
7		+ (VDC)	Power VDC (red)
8		+ (VDC)	Power VDC (red)

Table 9 - Pinout correspondence between connectors

A.2 DAVS BS detailed description

A.2.1 General Overview

DAVS Base Station is a telemetry and power system, companion of DAVS which allows an easy visualization of acquired signals and an interface for change some settings on the fly. This system was developed by Marsensing company. We will not detail how the system works, just explain the basic information to get it working and fix any simple problems that may occur.

All the system components are inside a Peli case (model 1200), protecting them from water and allowing for an easy sea deployment. It includes an underwater cable for direct connection to DAVS. Figure 74 shows the system and the opened case.

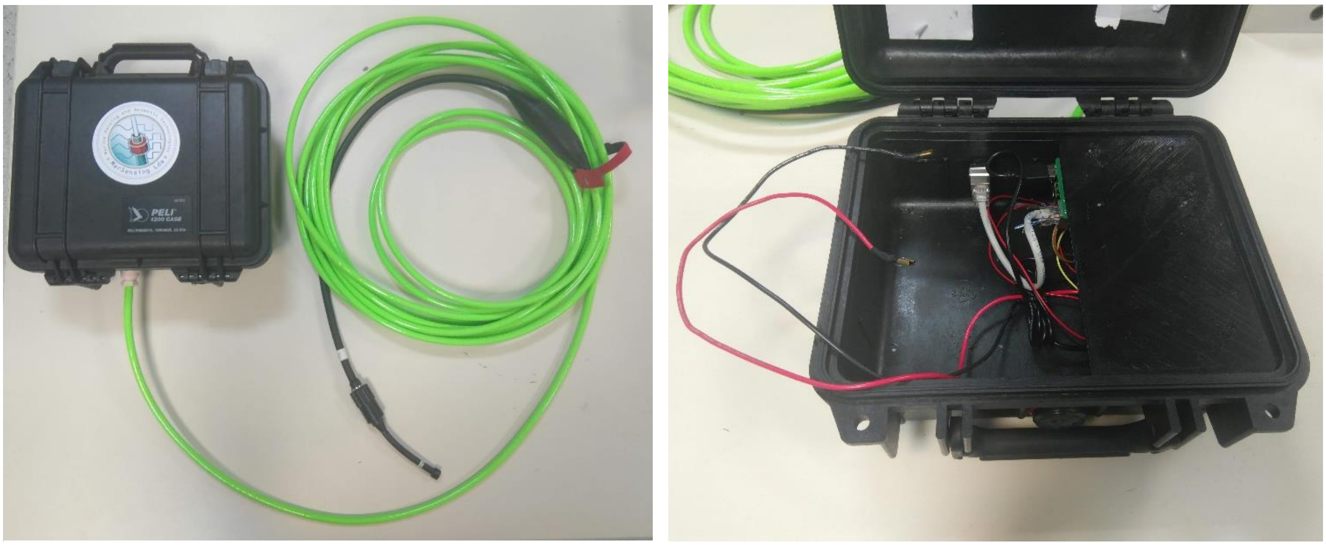


Figure 74 - DAVS Base Station with case and external Ethernet green cable (left image) and case opened without battery (right image)

DAVS Base Station will work as a Wi-Fi access point (hot spot) and web server, receiving the streaming data from DAVS and generating a spectrogram view of any channel on a web page. At the same time, this web page will allow the change of some DAVS parameters (Gains, sampling frequency) or access to recorded files. As a Wi-Fi hot spot, multiple host can connect to it, so be careful not to change any setting in different hosts.

A.2.2 Electronic Components

This system is composed of several distinct components, as shown in Figure 75, namely:

- 1 Raspberry Pi, which manage all the system components

- 4 voltage regulators, for converting the different voltage levels needed;
- 1 USB-to-Ethernet, allowing the usage of 2 Ethernet ports on Raspberry Pi;
- 1 Real Time Clock for storing date and time information;
- 1 External relay, for turning ON and OFF the power wires of external Ethernet cable. This will only turn DAVS ON when Raspberry Pi operating system define;
- 1 USB hub for extending USB ports. Not used in our case;
- 1 GSM module, for sending data through cellular network. Not used in our case.

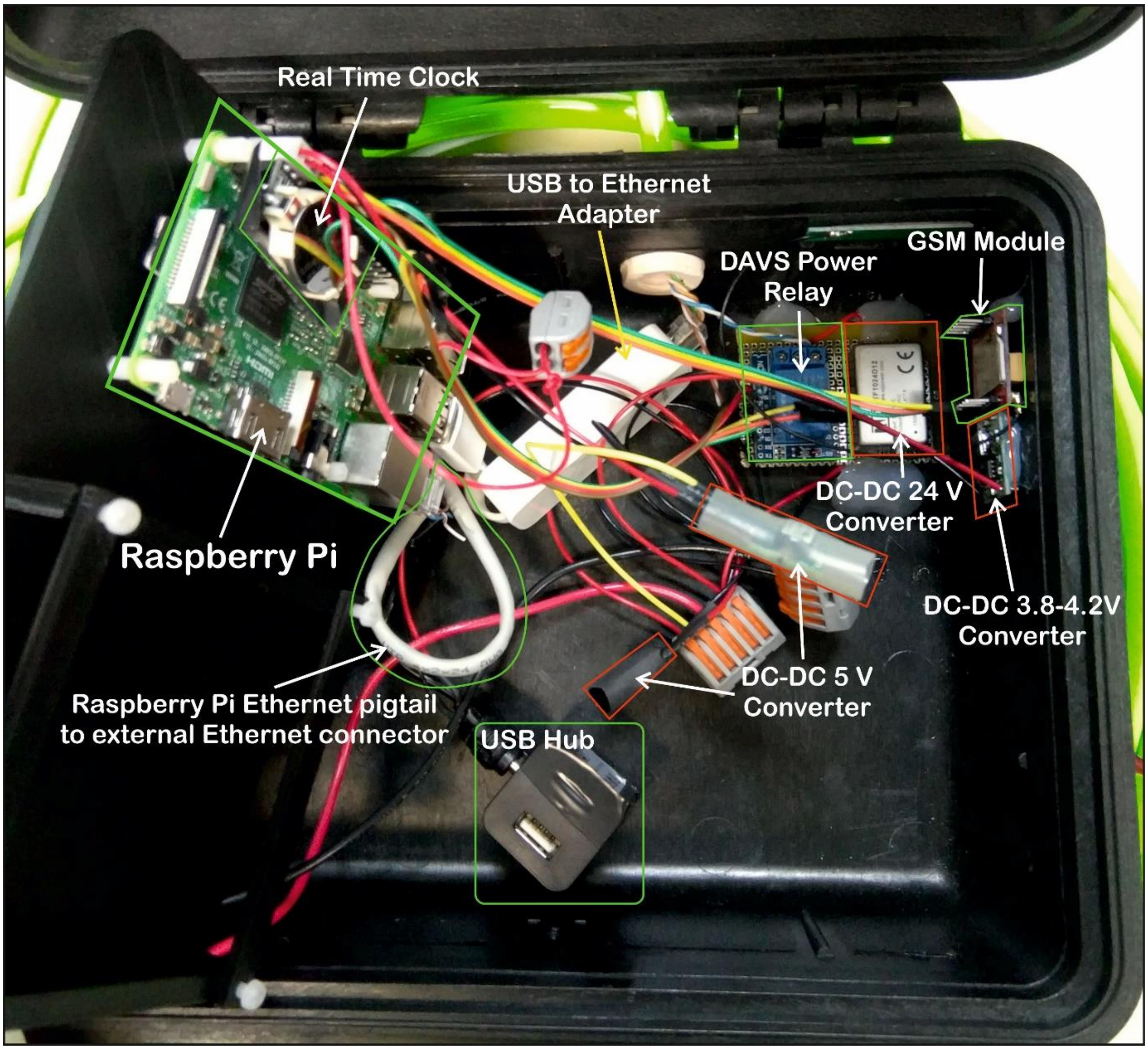


Figure 75 - DAVS Base Station components identified, inside the case

The interconnection between these components is shown in Figure 76, where the black arrows represents data communications and red arrows represents the energy flow. Note that no DC-DC regulators are represented.

A.2.3 Power Sources and Control

DAVS Base Station is powered by a 12 VDC VRLA type battery (model UL9-12), with 2 male spade type connectors and a capacity of 9 Ah. This battery distribute energy for 4 DC-DC voltage regulators, that adapt the voltages for the different components (Figure 77).

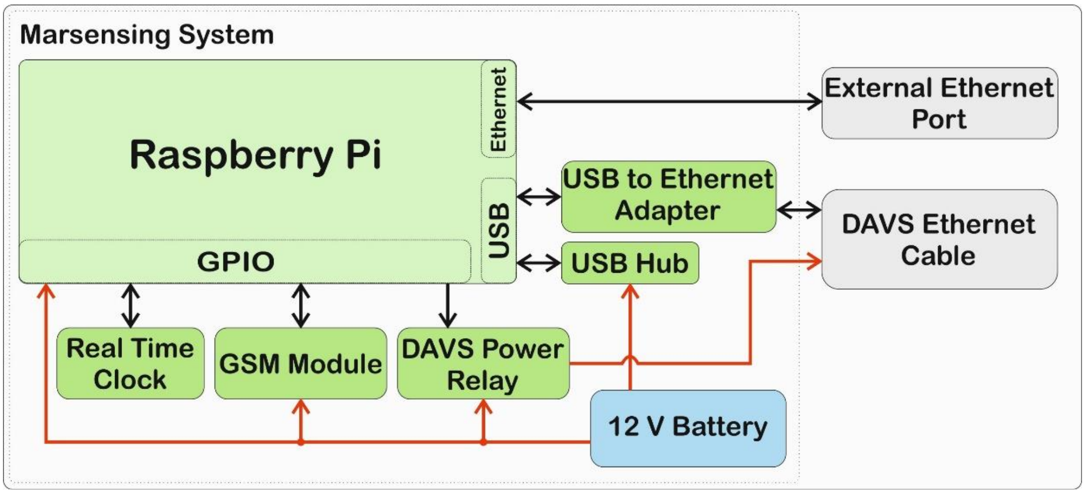


Figure 76 - DAVS Base Station components interconnection diagram

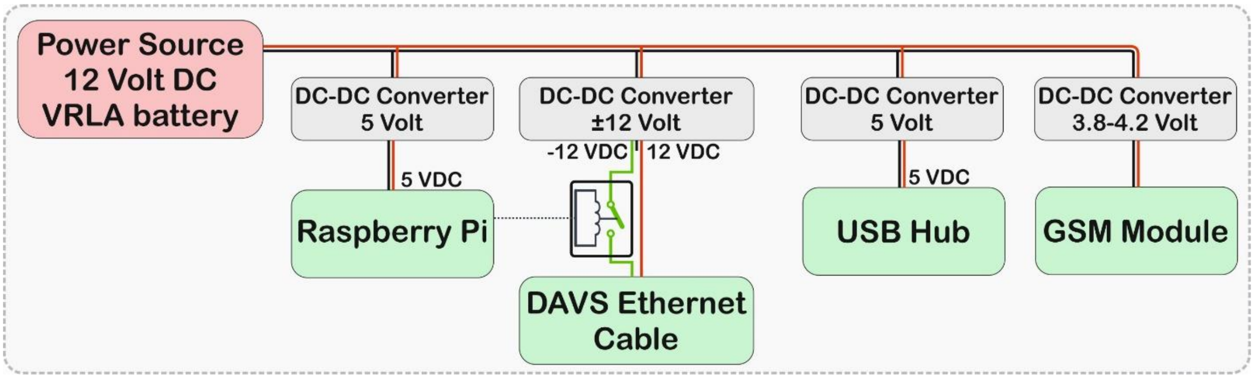


Figure 77 - Power distribution in DAVS Base Station

The used regulators are:

- 2 Murata OKI-78SR-5 which convert the 12 V from the battery to 5 V needed in the Raspberry Pi and USB Hub.
- 1 XP Power JTF1024D12, which have a dual symmetrical output of ±12 V, allowing to be used as a 24 V source, when using -12 V output as a ground. This output will be connected to a relay and to Ethernet cable which provide energy to DAVS. This relay is controlled by Raspberry Pi, only powering DAVS when the system is ready to receive data;
- 1 DC-DC regulator for GSM module which have a restricted input of 3,8 to 4,2 Volts.

A.2.4 External Connections

The system has two external connectors, which are part of the box and can't be removed from it (check Figure 78):

- An external Ethernet port, allowing the connection to any other host computer;
- An underwater Ethernet Cat 5e cable (Falmat Xtreme Cat series), terminated with a MCBH-8-F connector for DAVS connection.

The Ethernet cable includes a dummy cap for protection of the terminals and the locking sleeve which will tighten to DAVS connector. The dummy cap should be connected to MCBH connector for protection, when not using the cable.

In the inside of the case, there are 2 separate connections for Ethernet ports. One is a small pigtail to connect the external connector to Raspberry Pi and the other one is the Ethernet connector for green external cable. Last one should be pluged to USB-to-Ethernet adapter as in Figure 79. The USB side of the adapter must be connected to any USB port of Raspberry Pi.

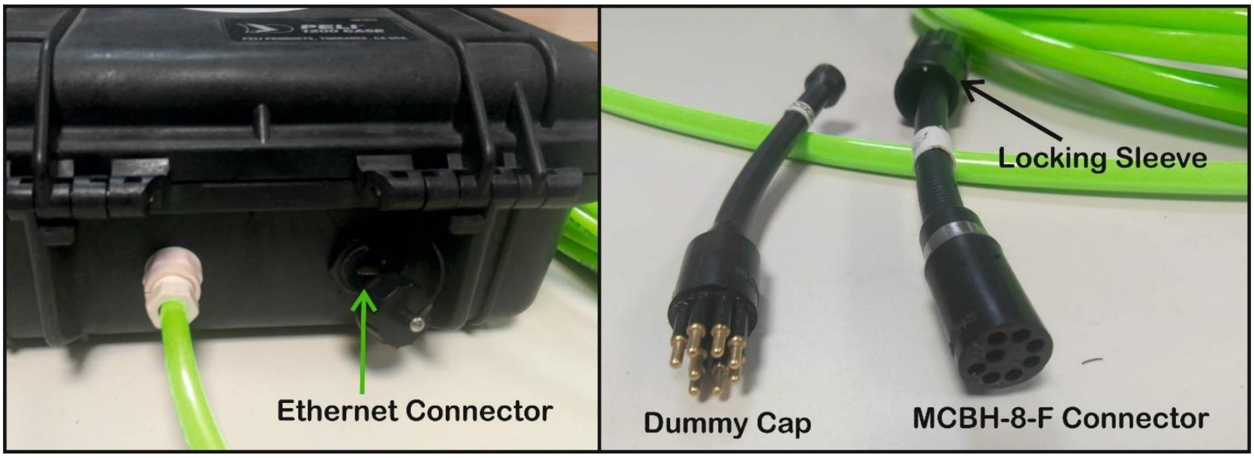


Figure 78 - External connectors. Ethernet external connector (left side) and MCBH-8-F connector in underwater green cable, with dummy cap and locking sleeve (right side)

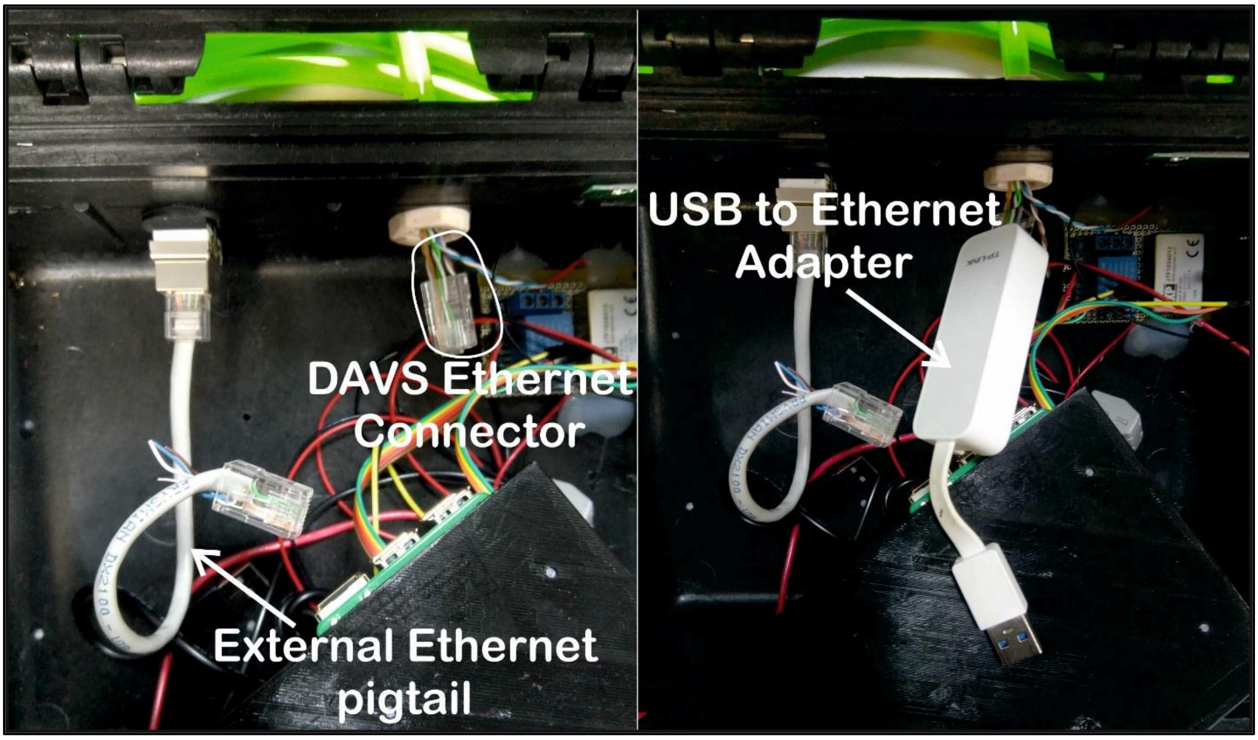


Figure 79 - Internal pigtail cables (left side) and USB to Ethernet adapter (right side)

A.3 Matlab example code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% nfpinto@ualg.pt       V02     09/03/2020
% DAVS data visualization and simples data processing examples
% #Visualization:
%       -Open WAVE files and plot waveforms
%       -Convert Data
%       -Plot spectrograms
% #Processing:
%       -Calculate FFT and show frequency analysis
%       -Bandpass filter signal and plot responses
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear; clc;

% Wave File location
path_file = 'C:\path_to_file\DATA0024.WAV';
% Getting Sampling Frequency Fs and Y datapoints from WAVE file
[y, Fs] = audioread(path_file);
fprintf('\nWAVe file loaded\nWAVe File sample rate: %d Hz ; Time Period: %f s', Fs, 1/Fs);      %NOTE: Just to
show in console whats beeing done

% Separate each channel from *.wav to a variable with source name
acc49_x = y(:,1);
acc49_y = y(:,2);
acc49_z = y(:,3);
hyd     = y(:,4);
acc50_x = y(:,5);
acc50_y = y(:,6);
acc50_z = y(:,7);

% Create a time array [0, Ts, 2Ts, 3Ts,... (Nt-1)Ts]
Nt = length(hyd);          %NOTE: Could be obtained from another variable
t = (0:Nt-1)/Fs;


%%
%   Print normalized waveform examples

fprintf('\nCreating normalized waveforms figures examples...')
% Create a simple waveform from a single channel
figure('Name', 'Acc#49 X axis Normalized Waveform')
plot(t, acc49_x)
grid on
axis tight
title('Acc#49 X')
xlabel('time(s)')
ylabel('normalized amplitude')
fprintf('.')

% Create waveforms from accelerometer tri-channel data into the same figure
figure('Name', 'Accelerometer #49 Normalized Waveforms')
plot(t, acc49_x, 'r', t, acc49_y, 'g--', t, acc49_z, 'b:' )
grid on
axis tight
legend
title('Acc#49 Waveforms')
xlabel('time(s)')
ylabel('normalized amplitude')
fprintf('.')

% Raw waveforms from accelerometer
figure('Name', 'Raw Accelerometer Measured Data')
subplot(2,3,1)
plot(t, acc49_x)
axis([0  ((Nt-1)/Fs)  -1  1])
grid on
title('Acc#49 X axis')
xlabel('time(s)')
ylabel('normalized amplitude')
fprintf('.')
subplot(2,3,2)
plot(t, acc49_y)
axis([0  ((Nt-1)/Fs)  -1  1])
grid on
title('Acc#49 Y axis')
xlabel('time(s)')
ylabel('normalized amplitude')
fprintf('.')
subplot(2,3,3)
plot(t, acc49_z)
axis([0  ((Nt-1)/Fs)  -1  1])
grid on
title('Acc#49 Z axis')
xlabel('time(s)')
```

```
ylabel('normalized amplitude')
fprintf('.')
subplot(2,3,4)
plot(t, acc50_x)
axis([0 ((Nt-1)/Fs) -1 1])
grid on
title('Acc#50 X axis')
xlabel('time(s)')
ylabel('normalized amplitude')
fprintf('.')
subplot(2,3,5)
plot(t, acc50_y)
axis([0 ((Nt-1)/Fs) -1 1])
grid on
title('Acc#50 Y axis')
xlabel('time(s)')
ylabel('normalized amplitude')
fprintf('.')
subplot(2,3,6)
plot(t, acc50_z)
axis([0 ((Nt-1)/Fs) -1 1])
grid on
title('Acc#50 Z axis')
xlabel('time(s)')
ylabel('normalized amplitude')
fprintf('.')

fprintf(' DONE - Raw data waveforms created!\n')


%%
%   Constant list for data conversion

%PGA gains list (for header parsing)
list_pga_gain = [1, 2, 4, 8, 16, 32, 64];

%HYD Sensitivity and preamp gain
hyd_sensitivity_db  = -195;                          % -195 dB re V/uPa
hyd_preamp_gain     = 39;

%ADC The normalized WAVE in range +-1 correspond to a +-5 V at the input of SIngleEnded to DIFF converter
adc_sensitivity = 5;

%ACC Manufacturer sensitivity, an acceleration of 1 (m/s^2) produce 51mV at acc output
acc_sensitivity = 51; %51 mV/(m/s^2)


%%
%   WAVE header simple parser

%Get PGA gains from WAVE header
fr = matlab.io.datastore.DsFileReader(path_file );
seek(fr, 132);  %jump to header byte position 132, where Hydrophone PGA gain info is stored
index = read(fr,1,'OutputType','uint8');
if (index == 0) %Its possible to have 0 value in gain, which is == 1
    index = 1;
end
hyd_pga_gain    = list_pga_gain(index);
seek(fr, 37); %jump to header position 170 (from 133)
index = read(fr,1,'OutputType','uint8');
if (index == 0)
    index = 1;
end
acc_pga_gain    = list_pga_gain(index);
clear fr index


%%
%   Convert RAW values into Acceleration and SP/SPL

%Applying conversion formulas from chapter 4.2.2
acc49_x_a = acc49_x * (adc_sensitivity / (acc_sensitivity*10^-3 * acc_pga_gain ) );
acc49_y_a = acc49_y * (adc_sensitivity / (acc_sensitivity*10^-3 * acc_pga_gain ) );
acc49_z_a = acc49_z * (adc_sensitivity / (acc_sensitivity*10^-3 * acc_pga_gain ) );
sp        = hyd * adc_sensitivity / [ 10^(hyd_sensitivity_db/20)* hyd_preamp_gain * hyd_pga_gain ];
spl       = 20 * log10(sp) ;
acc50_x_a = acc50_x * (adc_sensitivity / (acc_sensitivity*10^-3 * acc_pga_gain ) );
acc50_y_a = acc50_y * (adc_sensitivity / (acc_sensitivity*10^-3 * acc_pga_gain ) );
acc50_z_a = acc50_z * (adc_sensitivity / (acc_sensitivity*10^-3 * acc_pga_gain ) );

%Acceleration and SPL plot
fprintf('Creating converted values figures examples...')
figure('Name','Acceleration and Pressure converted Data');
subplot(3,3,1)
plot(t, acc49_x);
axis tight; grid on;
title('Acc#49 X Axis'); xlabel('Time'); ylabel('Acceleration (m/s^2)'); fprintf('.')
subplot(3,3,2)
plot(t, acc49_y);
```

```
axis tight; grid on;
title('Acc#49 Y Axis'); xlabel('Time'); ylabel('Acceleration (m/s^2)'); fprintf('.')
subplot(3,3,3)
plot(t, acc49_z);
axis tight; grid on;
title('Acc#49 Z Axis'); xlabel('Time'); ylabel('Acceleration (m/s^2)'); fprintf('.')
subplot(3,3,4)
plot(t, acc50_x);
axis tight; grid on;
title('Acc#50 X Axis'); xlabel('Time'); ylabel('Acceleration (m/s^2)'); fprintf('.')
subplot(3,3,5)
plot(t, acc50_y);
axis tight; grid on;
title('Acc#50 Y Axis'); xlabel('Time'); ylabel('Acceleration (m/s^2)'); fprintf('.')
subplot(3,3,6)
plot(t, acc50_z);
axis tight; grid on;
title('Acc#50 Z Axis'); xlabel('Time'); ylabel('Acceleration (m/s^2)'); fprintf('.')
subplot(3,3,7)
plot(t, abs(spl) );
axis tight; grid on;
title('Hydrophone'); xlabel('Time'); ylabel('Sound pressure Level SPL (dB re uPa)'); fprintf('.')
subplot(3,3,8)
plot(t, sp );
axis tight; grid on;
title('Hydrophone'); xlabel('Time'); ylabel('Sound pressure SL (uPa)'); fprintf('.')
fprintf(' DONE - Converted data figures created!\n')


%%
%   Audio play using audioplayer

start_seconds = 49;
stop_seconds = 63;
player = audioplayer(acc49_x, Fs);
% Play audio between 2 points>>to convert to seconds just multiply by Sample Rate
play(player, [player.SampleRate * start_seconds , player.SampleRate * stop_seconds ])
% Hear the complete file
%play(player)
% Stop the audio reproduction
%stop(player)


%%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %   Spectrogram examples
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fprintf('\nCreating spectrograms...')
% Spectrogram settings, using Matlab default window algorithm (Hanning window)
% Function Input: spectrogram(x,window,noverlap,nfft,fs,f_axis)
% Windows size >> window and nfft (Lower NFFT more time detail, less freq detail)
Window_size = 1024;
% Number of overlap points
Overlap_size = round( Window_size/4.5);

% 3D Spectrogram view from hydrophone
figure('Name', 'Raw Hydrophone Measured Data')
axis([0 ((Nt-1)/Fs) 0 Fs/2])
spectrogram(hyd, Window_size, Overlap_size, Window_size, Fs, 'yaxis');
colormap(jet)
view(-45,70)
title('Hydrophone Data')

% Raw spectrogram of recorded data
figure('Name', 'Raw Accelerometer Measured Data')
axis([0 ((Nt-1)/Fs) 0 Fs/2])
colormap(jet)
subplot(2,3,1)
spectrogram(acc49_x, Window_size, Overlap_size, Window_size, Fs, 'yaxis');
title('Acc#49 X axis Accelerometer')
fprintf('.')
subplot(2,3,2)
spectrogram(acc49_y, Window_size, Overlap_size, Window_size, Fs, 'yaxis');
title('Acc#49 Y axis Accelerometer')
fprintf('.')
subplot(2,3,3)
spectrogram(acc49_z, Window_size, Overlap_size, Window_size, Fs, 'yaxis');
title('Acc#49 Z axis Accelerometer')
fprintf('.')
subplot(2,3,4)
spectrogram(acc50_x, Window_size, Overlap_size, Window_size, Fs, 'yaxis');
title('Acc#50 X axis Accelerometer')
fprintf('.')
subplot(2,3,5)
spectrogram(acc50_y, Window_size, Overlap_size, Window_size, Fs, 'yaxis');
title('Acc#50 Y axis Accelerometer')
fprintf('.')
subplot(2,3,6)
```

```
spectrogram(acc50_z, Window_size, Overlap_size, Window_size, Fs, 'yaxis');
title('Acc#50 Z axis Accelerometer')
fprintf('.')

fprintf(' DONE - Raw data spectrograms created!\n')


%%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %   Processing examples - frequency analysis
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Processing example, getting Acc49_x from 45.8 to 45.9 seconds and FFT it
start_f_analysis = 45.8;
end_f_analysis = 45.9;
start_samples = round( start_f_analysis / (1/Fs) );         %second to sample
end_samples = round( end_f_analysis / (1/Fs) );
fprintf('\nFrequency Analysis. Analysing file between %f seconds and %f seconds.\nCalculating FFT ...',
start_f_analysis, end_f_analysis)

%Analysis zone waveform
partial_processing = acc49_x(start_samples:end_samples);
partial_processing_t = (0:1/Fs:(length(partial_processing)-1)/Fs);
fprintf('.')

%Calc FFT
partial_processing_size = length(partial_processing);
partial_processing_n = pow2(nextpow2(partial_processing_size));  % transform length
partial_processing_y = fft(partial_processing, partial_processing_n);
fprintf('.')

%Frequencies array
partial_processing_f = (0:partial_processing_n-1) * (Fs / partial_processing_n);
%Power spectrum
partial_processing_power = abs(partial_processing_y).^2/partial_processing_n;
%partial_processing_power = abs(partial_processing_y/partial_processing_n);
fprintf('.')

%Spectrogram configurations
partial_processing_ws = 1024;
partial_processing_os = round( partial_processing_ws/4.5);
partial_processing_nfft = 1024;

%Waveform, Spectrogram and Frequency analysis image
figure('Name', 'Frequency Processing Data')
subplot(2,2,1)
plot(partial_processing_t,partial_processing)
axis([0  ((partial_processing_size-1)/Fs)  -1  1])
grid on
title('Acc#49 X axis selected area')
xlabel('time (s)')
ylabel('amplitude')
fprintf('.')
subplot(2,2,2)
spectrogram(partial_processing, partial_processing_ws, partial_processing_os, partial_processing_nfft, Fs,
'xaxis');
title('Acc#49 X axis Spectrogram')
fprintf('.')
subplot(2,2,3)
plot(partial_processing_t,partial_processing)
grid on
title('Acc#49 X axis selected area zoomed')
xlabel('time (s)')
ylabel('amplitude')
fprintf('.')
subplot(2,2,4)
plot(partial_processing_f(1:floor(partial_processing_n/2)) ,
partial_processing_power(1:floor(partial_processing_n/2)));
grid on
axis tight
title('Acc#49 X axis Frequency Analysis')
xlabel('frequency (Hz)')
ylabel('power')

fprintf('. DONE - FFT calculated!\n')


%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Processing examples - Bandpass filtering
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Processing example, bandpass filetring example for AX49 data
% Define low and high cut frequencies
low_f = 1000;
high_f = 3000;
fprintf('\nFilter data from  ACCEL #49, using a bandpass filter from %d to %d Hz.\nFiltering...', low_f ,high_f
)
% filter example, using fixed stepness and attenuation
```

```
acc49_x_filtered = bandpass(acc49_x, [low_f high_f], Fs, 'Steepness', 0.99, 'StopbandAttenuation' , 30);

% Original and filtered data image
figure('Name', 'Acc#49 X axis Accelerometer Data and bandpass filtering')
subplot(2,2,1)
plot(t, acc49_x)
axis([0 ((Nt-1)/Fs) -1 1])
grid on
title('Original Acc#49 X axis Accelerometer data')
fprintf('.')
subplot(2,2,2)
spectrogram(acc49_x, Window_size, Overlap_size, Window_size, Fs, 'yaxis');
title('Original Acc#49 X axis Accelerometer Spectrogram')
fprintf('.')
subplot(2,2,3)
plot(t, acc49_x_filtered)
axis([0 ((Nt-1)/Fs) -1 1])
grid on
title('Filtered Acc#49 X axis Accelerometer data')
fprintf('.')
subplot(2,2,4)
spectrogram(acc49_x_filtered, Window_size, Overlap_size, Window_size, Fs, 'yaxis');
title('Filtered Acc#49 X axis Accelerometer data')
fprintf('.')

% Waveform and frequency response graphs
figure('Name', 'Acc#49 X axis Accelerometer waveform and frequency response using bandpass function')
bandpass(acc49_x, [low_f high_f], Fs, 'Steepness', 0.99, 'StopbandAttenuation' , 30);
fprintf('.')

fprintf(' DONE - Filtered data created!\n')
```

A.4 Fast Fourier Transform

In a very simplified way, FFT algorithm works by dividing the original signal samples into small groups (the FFT size), applying then some math to that group of samples and obtaining the spectral components. However, these groups of samples in time domain are usually multiplied by a window function that change his shape before doing the FFT. Figure 80 demonstrates an example of how the windowing of a signal is done. The original signal is a non-periodic signal with 1024 samples, which start and finish with different amplitudes. This difference of amplitudes cause problems when converting to frequency domain (spectral leakage). To avoid it, it's common to multiply the original signal by a windowing function, turning the first and last samples to be the same value (zero in this case).

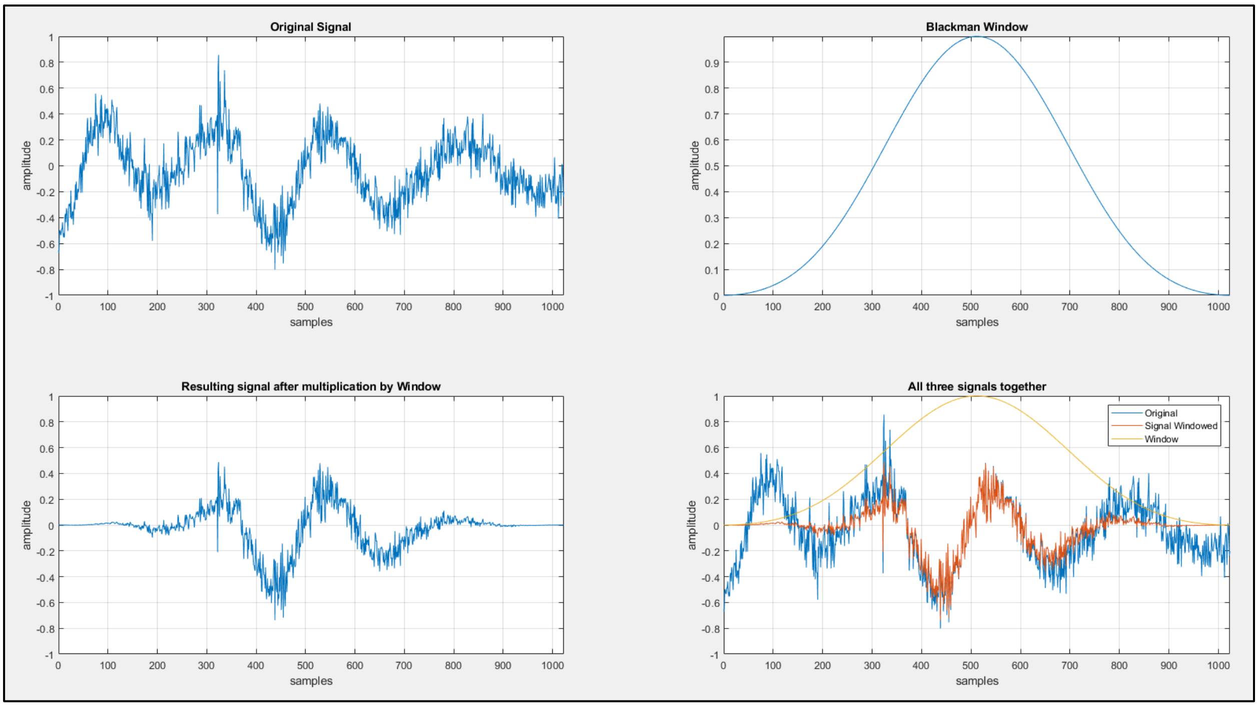


Figure 80 - Windowing a signal using a Blackman window. Top left graph is the original 1024 samples of the signal and top right is the Blackman window function. Bottom left graph is the samples after the multiplication by the window function. Bottom right graph shows the three signals for comparison

There are several types of window functions, each one with different shapes. In our case, the DAVS Base Station web interface have Rectangular, Hamming, Hanning, Blackman, Nuttal and Blackman-Nuttal window types available. These windows shapes are shown in Figure 81. In DAVS Base Station web interface it's possible to adjust the FFT window type and FFT samples size.

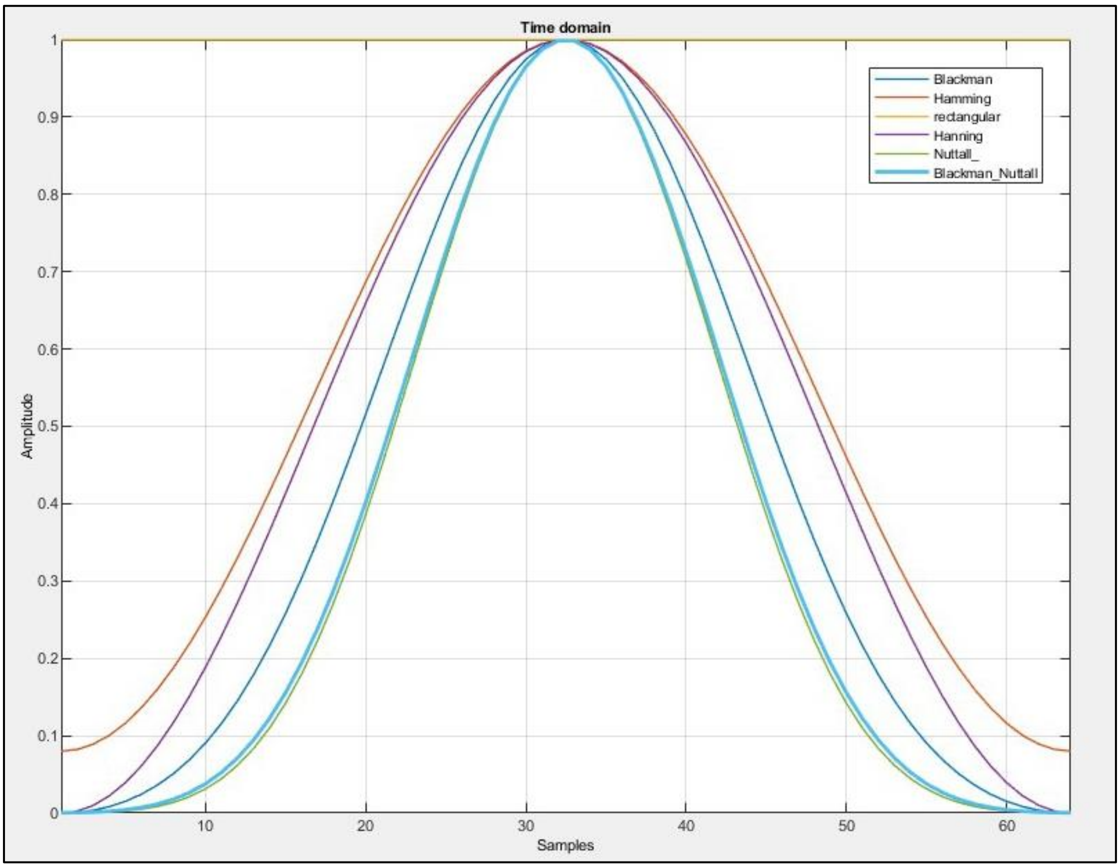


Figure 81 - DAVS Base Station web interface FFT Window functions (64 samples size)

A.5 DeviceDetect CSV File structure

The “*DeviceDetect0_XXXhhmmss.csv*” files stores DAVS information captured in cabled mode and during the operative time of the device. It´s a comma separated values file, containing 20 data fields with DAVS information, settings and inertial sensor data. Each line is a second of operating time or a change in some setting. This file doesn’t have any header line, so the 20 columns are:

1) **Unix timestamp** of stored data, the DAVS Base Station internal time in Posix format;
2) **DAVS ID**, the DAVS identifier (0215);
3) The **device description**, in text format;
4) The **number of channels available**;
5) **DAVS unix timestamp**, the internal DAVS posix time in microseconds. Based on internal RTC time. However, this time only have seconds resolution, so it should be divided by 1000000 to remove the unused zeros;
6) **DAVS IP address**;
7) **Number of Sample Rate settings available**;
8) List of available **Sample Rate settings**;
9) **Number of gain settings available**;
10) List of **gains settings available**;
11) **Chosen Sample Rate setting list index**, being a number between 0 and 3 corresponding to sample rate list index of chosen sample rate through web interface;
12) **Chosen Sample Rate** setting value;
13) **ADC bits** used (always 24 bits);
14) **Chosen hydrophone gain**;
15) **Chosen hydrophone gain list index** (between 0 and 7);
16) **Chosen accelerometer gain**;
17) **Chosen accelerometer gain list index** (between 0 and 7);
18) **Roll angle** (degrees);
19) **Pitch angle** (degrees);
20) **Heading angle** (degrees).

The last 3 columns are the inertial information, useful for know the DAVS position. In Figure 82 there’s an excerpt of a file opened in Microsoft Excel and with the column names on first line. Please note that the original file doesn’t have this header line. For more information’s about these parameters, please check appendix A.1.

<table>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th><th>G</th><th>H</th><th>I</th><th>J</th><th>K</th><th>L</th><th>M</th><th>N</th><th>O</th><th>P</th><th>Q</th><th>R</th><th>S</th><th>T</th></tr>
<tr><td>1</td><td>Unix Timestamp</td><td>DAVS ID</td><td>Device Description</td><td>Channels</td><td>DAVS Unix timestamp</td><td>DAVS IP</td><td>SPS N</td><td>Available SPS</td><td>Gains N</td><td>Available Gains</td><td>SPS Idx</td><td>SPS</td><td>Bits</td><td>HYD Gain</td><td>HYD Gain Idx</td><td>Accel. Gain</td><td>Accel. Gain Idx</td><td>Roll</td><td>Pitch</td><td>Heading</td></tr>
<tr><td>2</td><td>1575454263</td><td>215</td><td>digitalHyd TP-1 7Ch DAVSv1</td><td>8</td><td>1575455086000000</td><td>192.168.99.121</td><td>4</td><td>10547|52734|52734|105469|</td><td>8</td><td>0|1|2|4|8|16|32|64|</td><td>0</td><td>0</td><td>24</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0.00</td><td>0.00</td><td>360.00</td></tr>
<tr><td>3</td><td>1575454264</td><td>215</td><td>digitalHyd TP-1 7Ch DAVSv1</td><td>8</td><td>1575455086000000</td><td>192.168.99.121</td><td>4</td><td>10547|52734|52734|105469|</td><td>8</td><td>0|1|2|4|8|16|32|64|</td><td>0</td><td>0</td><td>24</td><td>1</td><td>1</td><td>1</td><td>1</td><td>-84.57</td><td>-8.31</td><td>183.53</td></tr>
<tr><td>4</td><td>1575454265</td><td>215</td><td>digitalHyd TP-1 7Ch DAVSv1</td><td>8</td><td>1575455087000000</td><td>192.168.99.121</td><td>4</td><td>10547|52734|52734|105469|</td><td>8</td><td>0|1|2|4|8|16|32|64|</td><td>0</td><td>0</td><td>24</td><td>1</td><td>1</td><td>1</td><td>1</td><td>-84.57</td><td>-8.31</td><td>183.53</td></tr>
<tr><td>5</td><td>1575454266</td><td>215</td><td>digitalHyd TP-1 7Ch DAVSv1</td><td>8</td><td>1575455088000000</td><td>192.168.99.121</td><td>4</td><td>10547|52734|52734|105469|</td><td>8</td><td>0|1|2|4|8|16|32|64|</td><td>0</td><td>0</td><td>24</td><td>1</td><td>1</td><td>1</td><td>1</td><td>-84.57</td><td>-8.31</td><td>183.53</td></tr>
<tr><td>6</td><td>1575454267</td><td>215</td><td>digitalHyd TP-1 7Ch DAVSv1</td><td>8</td><td>1575455089000000</td><td>192.168.99.121</td><td>4</td><td>10547|52734|52734|105469|</td><td>8</td><td>0|1|2|4|8|16|32|64|</td><td>0</td><td>0</td><td>24</td><td>1</td><td>1</td><td>1</td><td>1</td><td>-84.57</td><td>-8.31</td><td>183.53</td></tr>
<tr><td>7</td><td>1575454268</td><td>215</td><td>digitalHyd TP-1 7Ch DAVSv1</td><td>8</td><td>1575455090000000</td><td>192.168.99.121</td><td>4</td><td>10547|52734|52734|105469|</td><td>8</td><td>0|1|2|4|8|16|32|64|</td><td>0</td><td>0</td><td>24</td><td>1</td><td>1</td><td>1</td><td>1</td><td>-84.57</td><td>-8.31</td><td>183.53</td></tr>
<tr><td>8</td><td>1575454269</td><td>215</td><td>digitalHyd TP-1 7Ch DAVSv1</td><td>8</td><td>1575455091000000</td><td>192.168.99.121</td><td>4</td><td>10547|52734|52734|105469|</td><td>8</td><td>0|1|2|4|8|16|32|64|</td><td>0</td><td>0</td><td>24</td><td>1</td><td>1</td><td>1</td><td>1</td><td>-84.57</td><td>-8.31</td><td>183.53</td></tr>
<tr><td>9</td><td>1575454270</td><td>215</td><td>digitalHyd TP-1 7Ch DAVSv1</td><td>8</td><td>1575455092000000</td><td>192.168.99.121</td><td>4</td><td>10547|52734|52734|105469|</td><td>8</td><td>0|1|2|4|8|16|32|64|</td><td>0</td><td>0</td><td>24</td><td>1</td><td>1</td><td>1</td><td>1</td><td>-84.57</td><td>-8.31</td><td>183.53</td></tr>
<tr><td>10</td><td>1575454271</td><td>215</td><td>digitalHyd TP-1 7Ch DAVSv1</td><td>8</td><td>1575455093000000</td><td>192.168.99.121</td><td>4</td><td>10547|52734|52734|105469|</td><td>8</td><td>0|1|2|4|8|16|32|64|</td><td>0</td><td>10547</td><td>24</td><td>1</td><td>1</td><td>1</td><td>1</td><td>-84.57</td><td>-8.31</td><td>183.53</td></tr>
<tr><td>11</td><td>1575454272</td><td>215</td><td>digitalHyd TP-1 7Ch DAVSv1</td><td>8</td><td>1575455094000000</td><td>192.168.99.121</td><td>4</td><td>10547|52734|52734|105469|</td><td>8</td><td>0|1|2|4|8|16|32|64|</td><td>0</td><td>10547</td><td>24</td><td>1</td><td>1</td><td>1</td><td>1</td><td>-84.57</td><td>-8.31</td><td>183.53</td></tr>
<tr><td>12</td><td>1575454272</td><td>215</td><td>digitalHyd TP-1 7Ch DAVSv1</td><td>8</td><td>1575455095000000</td><td>192.168.99.121</td><td>4</td><td>10547|52734|52734|105469|</td><td>8</td><td>0|1|2|4|8|16|32|64|</td><td>0</td><td>10547</td><td>24</td><td>32</td><td>6</td><td>1</td><td>1</td><td>-84.57</td><td>-8.31</td><td>183.53</td></tr>
<tr><td>13</td><td>1575454272</td><td>215</td><td>digitalHyd TP-1 7Ch DAVSv1</td><td>8</td><td>1575455095000000</td><td>192.168.99.121</td><td>4</td><td>10547|52734|52734|105469|</td><td>8</td><td>0|1|2|4|8|16|32|64|</td><td>0</td><td>10547</td><td>24</td><td>32</td><td>6</td><td>32</td><td>6</td><td>-84.57</td><td>-8.31</td><td>183.53</td></tr>
<tr><td>14</td><td>1575454272</td><td>215</td><td>digitalHyd TP-1 7Ch DAVSv1</td><td>8</td><td>1575455095000000</td><td>192.168.99.121</td><td>4</td><td>10547|52734|52734|105469|</td><td>8</td><td>0|1|2|4|8|16|32|64|</td><td>2</td><td>52734</td><td>24</td><td>32</td><td>6</td><td>32</td><td>6</td><td>-84.57</td><td>-8.31</td><td>183.53</td></tr>
<tr><td>15</td><td>1575454273</td><td>215</td><td>digitalHyd TP-1 7Ch DAVSv1</td><td>8</td><td>1575455095000000</td><td>192.168.99.121</td><td>4</td><td>10547|52734|52734|105469|</td><td>8</td><td>0|1|2|4|8|16|32|64|</td><td>2</td><td>52734</td><td>24</td><td>32</td><td>6</td><td>32</td><td>6</td><td>-84.57</td><td>-8.31</td><td>183.53</td></tr>
<tr><td>16</td><td>1575454274</td><td>215</td><td>digitalHyd TP-1 7Ch DAVSv1</td><td>8</td><td>1575455096000000</td><td>192.168.99.121</td><td>4</td><td>10547|52734|52734|105469|</td><td>8</td><td>0|1|2|4|8|16|32|64|</td><td>2</td><td>52734</td><td>24</td><td>32</td><td>6</td><td>32</td><td>6</td><td>-84.57</td><td>-8.31</td><td>183.53</td></tr>
<tr><td>17</td><td>1575454275</td><td>215</td><td>digitalHyd TP-1 7Ch DAVSv1</td><td>8</td><td>1575455097000000</td><td>192.168.99.121</td><td>4</td><td>10547|52734|52734|105469|</td><td>8</td><td>0|1|2|4|8|16|32|64|</td><td>2</td><td>52734</td><td>24</td><td>32</td><td>6</td><td>32</td><td>6</td><td>-84.57</td><td>-8.31</td><td>183.53</td></tr>
<tr><td>18</td><td>1575454276</td><td>215</td><td>digitalHyd TP-1 7Ch DAVSv1</td><td>8</td><td>1575455098000000</td><td>192.168.99.121</td><td>4</td><td>10547|52734|52734|105469|</td><td>8</td><td>0|1|2|4|8|16|32|64|</td><td>2</td><td>52734</td><td>24</td><td>32</td><td>6</td><td>32</td><td>6</td><td>-84.57</td><td>-8.31</td><td>183.53</td></tr>
<tr><td>19</td><td>1575454277</td><td>215</td><td>digitalHyd TP-1 7Ch DAVSv1</td><td>8</td><td>1575455099000000</td><td>192.168.99.121</td><td>4</td><td>10547|52734|52734|105469|</td><td>8</td><td>0|1|2|4|8|16|32|64|</td><td>2</td><td>52734</td><td>24</td><td>32</td><td>6</td><td>32</td><td>6</td><td>-84.57</td><td>-8.31</td><td>183.53</td></tr>
</table>

Figure 82 - A DeviceDetect0 csv file excerpt, showing all the columns and with column name header on first line ((Microsoft Excel on Windows 10 OS)

A.6 WAVE file header structure

The standard WAVE file is a RIFF container file, where some multimedia data can be placed. Additionally, can contain some information related to multimedia data properties. A RIFF file is divided into some parts (called chunks) with different utilities based on the content of file. In our case, it's divided into 4 parts: the standard header (the first 36 bytes), a custom header with DAVS information (following 456 bytes), a standard "fact" chunk (12 bytes) and the "data" chunk (the rest of the file).

In our case, the WAVE files will contain the direct output of the 8 24 bits ADC channels, at some sample rate in Pulse Code Modulation (PCM) format, represented in two's complement. This information has fixed positions into header, so can easily be read by any software. The number of used channels (bytes position 22), sample rate (24) and bits per sample (34) are part of the standard header information. The custom header space store hydrophone (132) and accelerometers selected gain (170), the acquisition date and time (from internal RTC, at positions 150-156) and 52 compass values (Roll, Pitch and Heading, from position 170 to 483). This last two information's only exists when in autonomous mode. The samples are stored into "data" chunk, starting into position 512. The three bytes in positions 512-514 contain the first sample of ADC channel 1, the 515-517 the first sample of second channel, and so on till the first samples of eight channel at position 533-535. Then there are the second groups of samples, following the same channel order. This structure repeats till the end of the file.

In Table 10 there's a resume of the DAVS WAVE header structure. The offset refers to the byte position in the header, while the field size is the space occupied by that information (in bytes). The field name has the standard name or function of that field, as described in field description. The endianness of each filed is described into Endian field, since the endianness change through the fields of a WAVE file. The field format indicates the data type and the field content indicate the default value from that field, if any. Some fields have constant values (like the number of channels) however some fields can change based on user settings (like sample rate or selected gains).

DAVS WAVE	Offset (dec) bytes	Field Name	Field size (bytes)	Field Description	Endian	Field Format	Field Content ,default or possible values (Hex and Decimal)
Standart	0	ChunkID	4	"RIFF" letters in ASCII. Indicates RIFF file format.	BIG	ASCII	0x52494646
	4	ChunkSize	4	File size - 8 (bytes)	Little	INT32	
	8	Format	4	"WAVE" letters in ASCII. Indicates a WAVE audio file format.	BIG	ASCII	0x57415645
	12	fmt subChunk	4	"fmt " letters in ASCII. Start of fmt subChunk	Big	ASCII	0x666D7420
	16	fmt subChunk size	4	tThe fmt subchunk size in bytes	Little	Int32	0xD8010000 (472d)
	20	Audio Format	2	The WAVE audio format, 01 for PCM (in two's complement)	Little	int16	0x0100 (01d)
	22	NumChannels	2	Number of audio channels, 8 for DAVS	Little	int16	0x0800 (08d)
	24	SampleRate	4	Sample rate used (Hz)	Little	int32	0x33290000 (10547d) ; 0xFD9B0100 (105469d)
	28	ByteRate	4	SampleRate*NumChannels*bitspersample/8	Little	int32	
	32	Block Align	2	NumChannels*BitsPerSample/8	Little	int16	0x1800 (24d)
	34	BitsPerSample	2	Bits per sample	Little	int16	0x1800 (24d)
Custom Header Info	36		2	Custom Header size	Little	int16	0xC601 (454d)
	38		2	Bits per sample	Little	int16	0x1800 (24d)
	44	??	4				
	50	??	16				
	132	Hyd Gain	1	Hydrophone Gain	Little	int8 (ou int16)	(1 - 16d)
	138	Bits per Sample	2	Bits	Little	int32	0x1800 (24d)
	140	SampleRate	4	Sample rate used (Hz)	Little	int32	0x33290000 (10547d) ;

	150	Year	2	Year (Only appears in Autonomous mode, 0x00 in Cabled Mode)	Little	int16	
	152	Month	1	Month	Little	int8	
	153	Day	1	Day	Little	int8	
	154	Hour	1	Hour	Little	int8	
	155	Minute	1	Minute	Little	int8	
	156	Second	1	Second	Little	int8	
	160	???	7	Appears on cabled mode, 0x00 in Autonomous			
	170	Acc Gain	1	Accelerometer Gain	Little	int8	
	172	Roll sample 1	2	Roll[1]	Little	int16	
	174	Pitch sample 1	2	pitch[1]	Little	int16	
	176	Heading sample 1	2	Heading[1]	Little	int16	
	178	Roll sample 2	2	Roll[2]	Little	int16	
	..	..	2	…			
	482	Heading sample 52	2	Heading[52]	Little	int16	
	484		2	End of info	Little	int16	0xFF
FMT sub chunk	492	"fact" subChunk	4	"fact" letters in ASCII	Big	ASCII	0x66616374
	496	"fact" subChunk size	4	"fact" subChunk size, 4 bytes	Little	int32	0x04000000 (04d)
	500	sample length per chn.	4	Number of Samples per channel	Little	int32	
Data sub chunk	504	"data" subChunk	4	"data" letters in ASCII. Where WAVE samples are	Big	ASCII	0x64617461
	508	"data" subChunk size	4	"data" subChunk size, depending on sample rate, and duration	Little	int32	
	512	1st channel PCM sample 1	3	First Sample from ADC channel1 in TWO COMPLEMENTS format	Little	2'sComplement	
	515	2nd channel	3	First Sample from ADC channel2 in TWO COMPLEMENTS format	Little	2'sComplement	
	518	3rd channel	3	…	Little	2'sComplement	
	521	4th channel	3		Little	2'sComplement	
	524	5th channel	3		Little	2'sComplement	
	527	6th channel	3		Little	2'sComplement	
	530	7th channel	3	…	Little	2'sComplement	
	533	8th channel	3	First Sample from ADC channel8 in TWO COMPLEMENTS format	Little	2'sComplement	
	536	1st channel PCM sample 2	3	Second Sample from ADC channel1 in TWO COMPLEMENTS format	Little	2'sComplement	
		…. Repeat till the end			Little	2'sComplement	

Table 10 - WAVE file header structure