

Infrastructure Provisioning, Computing Elements and Data Fusion

Documentation

Agrofelis

Document created on 2023-10-09 05:40:56

Abstract

The document presents the provisioning of the main computing elements for the Agrofelis robot and the initiation of its services. The peripheral elements established and attached to the local Agrofelis WiFi network and its USB hub are enumerated. In addition, the GPU component and the neural network accelerator tapping to the USBv3 interface of the Jetson Nano are presented. The Agrofelis Unificator Software, which connects all Agrofelis modules, is introduced along with its source code and features. The document also provides information on lightweight 3D printed structural elements that fit in the limited space available for the USB cables and secure the components and the USB hub in their compartment. The document concludes with a list of the components used and indicative suppliers, aiding the replication of the Agrofelis computing and networking systems.

Contents

Introduction	1
Jetson Nano Operating System Provisioning	4
Agrofelis Containerized Services	4
File System Provisioning	5
Docker Images Provisioning	6
Docker Images Instantiation	6
NginX	6
Unificator	6
Supportive	6
GPU and TPU Computing Modules	7
Agrofelis Unificator	7
Archived Data	10
Agrofelis Network	11
Agrofelis USB Hub	12
Structural Support for Computing Modules	12
Computing and Network Components	17
Summary	18

Introduction

The main computing elements of the Agrofelis robot are located in its back section on top of the steering plates. The following figure depicts their locations.

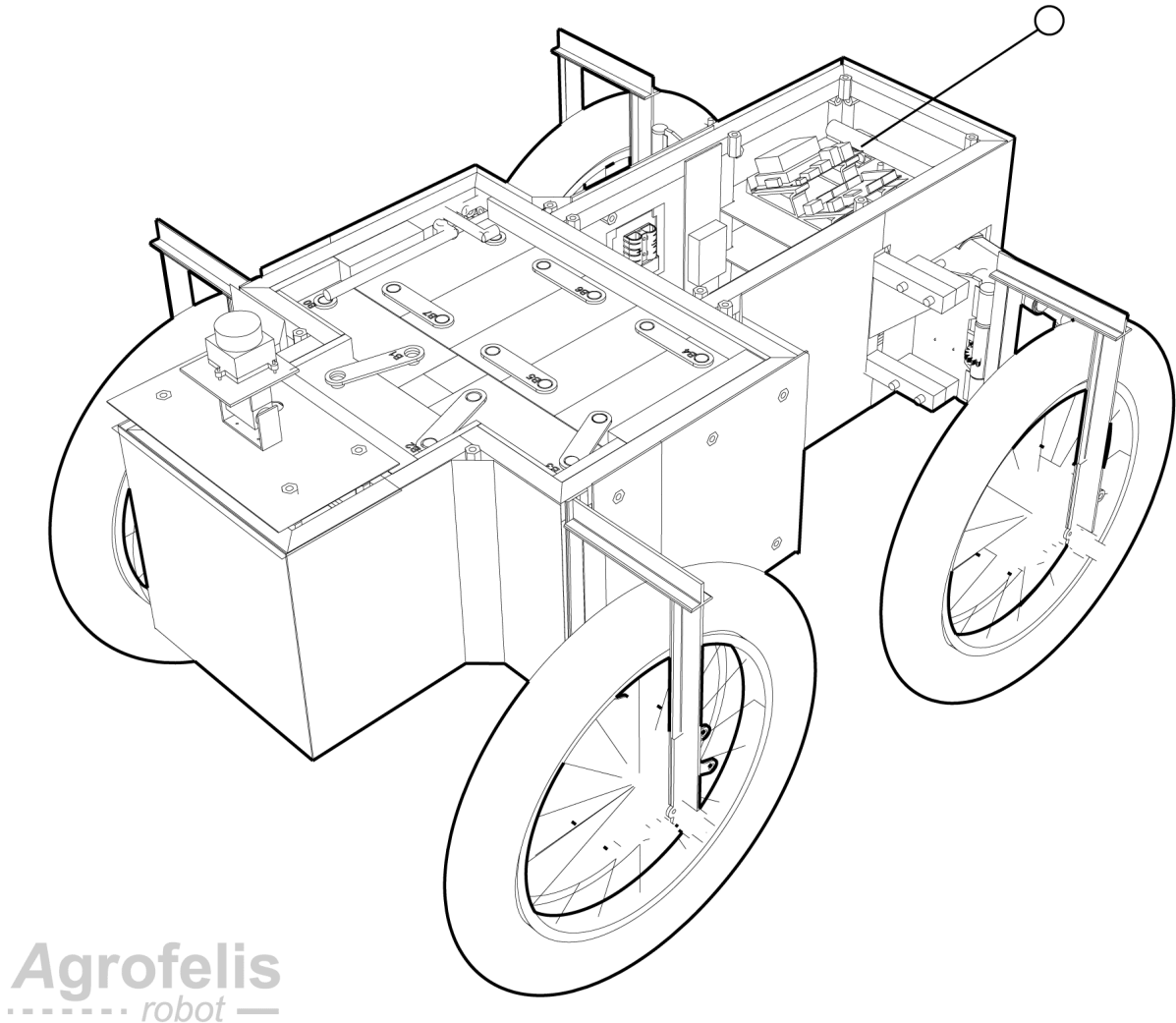


Figure 1: Overview

A relatively low-cost, fully-fledged GPU-capable computer, optimized for a small footprint and energy efficiency, aimed for AI applications, has been employed for the robot.



Figure 2: Jetson Nano

The computer is equipped with a high-speed 128 GB SD card with read and write speeds of 290 and 260 megabytes per second, addressing one of the key bottlenecks of the Jetson Nano computer.



Figure 3: high speed sd card

Furthermore, the Jetson Nano is equipped with an AI accelerator module connected to the USB3 port, expanding its capacity for parallel processing and neural network inference.



Figure 4: Coral accelerator

The Jetson Nano is machine vision ready, supporting high-speed, high-resolution camera inputs. The following photograph snapshots the camera of the Agrofelis robot, along with its 1-meter cable extension, connecting the computer to the front section of the vehicle.

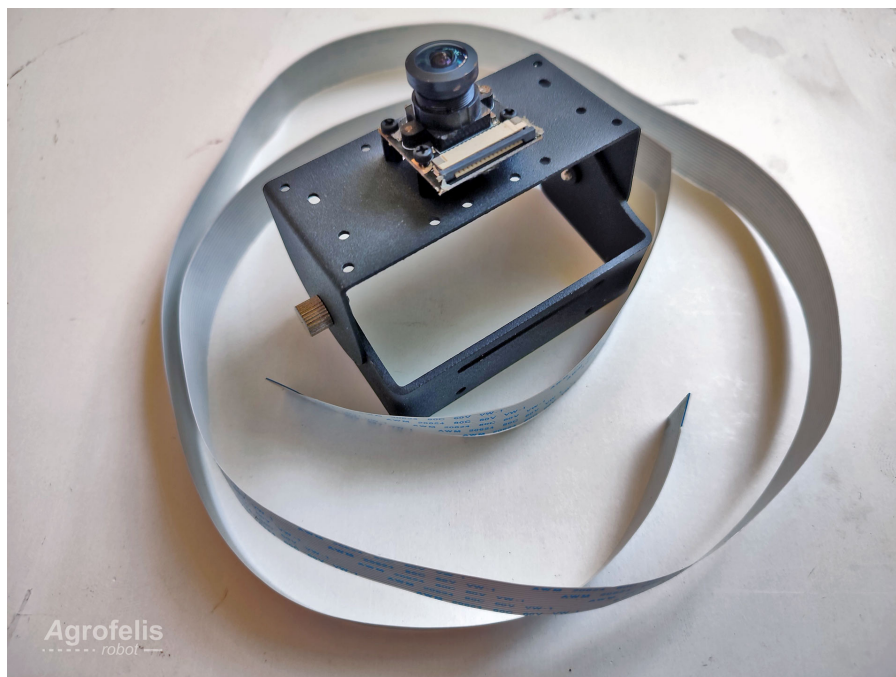


Figure 5: Camera and 1m cable

In the following sections, the details of provisioning the Jetson Nano computing module and its services, are documented.

Jetson Nano Operating System Provisioning

To initialize the computing module, one needs the same components as for a regular computer. In particular, there is a need for a keyboard, a mouse, a monitor, and a hard disk (SD card). An operating system can be written to the SD card using a turnkey Ubuntu-based image configured by NVIDIA, which has most of the needed software installed. The official [getting started document](#) can be followed to get acquainted with the embedded computer and bootstrap it.

Following NVIDIA's documentation, the 6.4 GB compressed operating system image can be downloaded from the following link.

- <https://developer.nvidia.com/jetson-nano-2gb-sd-card-image>

Due to the large file size, it is recommended to download it from a fast and stable internet connection. To write the image to the SD card, a Windows or a Linux computer can be utilised with the [balenaEtcher](#) as recommended by the NVIDIA documentation. Once the SD card is written it can be slotted in its placeholder and the computer can be connected with a 5V power supply.

Upon system initialization, the USB WiFi adapter or a network cable can be attached and used to connect to the internet in order to get the latest available updates, using the following commands.

```
apt-get update
apt-get get upgrade
apt-get autoremove
apt-get dist-upgrade
```

Two handy tools, one for monitoring the hardware resources of the computer and the other for multiplexing multiple command terminals, can be installed using the following command.

```
apt-get install htop screen
```

Once the Jetson Nano has been connected into the network, the monitor, keyboard and mouse can be removed and the computer can be operated using an SSH terminal. Although not essential, the computer can also be set up to stream its monitor using the open-source VNC software by issuing the following commands.

```
mkdir -p ~/.config/autostart
cp /usr/share/applications/vino-server.desktop ~/.config/autostart/.
gsettings set org.gnome.Vino prompt-enabled false
gsettings set org.gnome.Vino require-encryption false
gsettings set org.gnome.Vino authentication-methods "['vnc']"
gsettings set org.gnome.Vino vnc-password $(echo -n 'agrofelis'|base64)
```

The commands initialize a remote desktop server which can be handy when debugging for the first time the video camera, or alter the WiFi network settings or other settings provided by the graphical user interface of the operating system. The computer's monitor feed can be accessed using another computer via the freely available [VNC viewer](#) software.

Agrofelis Containerized Services

As observed, the operating system can be setup with just a handful commands, primarily because all common requirements have already been provisioned by NVIDIA in the aforementioned image. Another very useful software already pre-installed in the system is the NVIDIA Docker. Docker allows to containerize arbitrary computing environments, instantiate multiple instances of them and also revert, reuse or extend their definitions. Employing the Docker technology in the Agrofelis software stack means that its systems are easily reproducible, their definitions are transparent and their execution is somewhat isolated from the main operating system. Employing Docker, allows the operating system to remain clean of the application's dependencies and new technologies can be evaluated without the fear of "polluting" the system. The installed NVIDIA Docker variation can interface with the GPU hardware of the embedded computer which enables it to containerize AI applications.

In total, five Docker images have been developed, each layering functionality in a reusable manner, out of which two are mandatory for the robot's operation. The developed docker images are elaborated in further detail via the following table.

Image Name	Extends	Necessary	Description
agrofelis.unificator	node:16.17.0-bullseye-slim	Yes	The image extends the slim version of the main stream Nodejs image used to run the Agrofelis Unificator application. The particular image was found to have the smallest available footprint while still being able to run the intended application without further additions.
NGINX	Directly used	Yes	Although an image is not defined, a relevant folder is created next to the Agrofelis images definitions for persisting the NginX settings files that are mounted on the main stream container at the time of its instantiation. Nginx is the second most popular web server which at the same time is very lightweight and very fast because its implemented in C++. The Nginx docker instance is able to index and serve the www folder containing the web applications of the Unificator and of individual Agrofelis modules.
agrofelis.os	ubuntu:22.04	No	The docker image extends the main stream Ubuntu 22.04 operating system image, setups the local timezone related settings and installs some open source tools such as decompression, file synchronization, networking and file editing utilities. The image serves as a foundational layer used by consequent Agrofelis docker images as well as for evaluating new software on ephemeral instances.
agrofelis.os.arduino	agrofelis.os	No	The image extends the agrofelis.os foundational image and installs the arduino-cli command line tool, with it one can compile the Agrofelis steering module using an Arduino Mega and renew its firmware. The docker image definition encodes furthermore the exact 3rd party open source libraries required to compile the source code of the module. Although its not necessary to be able to compile the various micro-controllers firmware modules from the robot hardware, its convenient to share a transparent reproducible environment that anyone can effortlessly reuse to compile the software of the relevant modules. Having this docker image enables one to program the robot with just a regular computer with basic text processing capabilities. Moreover it could offer firmware updates by deploying a fraction of their binary size. The image source code can also be modified to compile the firmware of the ESP32 which is updated over the air.
agrofelis.os.R	agrofelis.os	No	The image extends the agrofelis.os foundational image and equips the environment with the R programming language focusing on statistics, machine learning and data manipulations and visualisation. The image is equipped with few powerful R packages the data.table a lightning fast data table manipulation engine and the FST package offering a very performant way of storing and compressing numerical information well balanced between the time needed to restore/write and the size needed to persist the information on disk. The combination of these packages and the power of the R language provide a very efficient framework to conduct data processing and machine learning procedures as well as rapidly prototype new applications.

File System Provisioning

The file system of the Agrofelis software is organized within the root path of the Jetson Nano operating system, in a folder named:

- /web-pub

The file system can be populated by exporting the [connectivity/](#) folder from the github repository and positioning it within the /web-pub folder of the Jetson nano. Following the filesystem installment and to have the web applications folder in a more prominent position, the **www** folder contained within the **connectivity** folder just established, is relocated within the **/web-pub/** folder. Lastly

for testing the steering source code compilation and firmware upload functionality, the [source code of the steering module](#) is exported and located at the `/web-pub/arduino/2023` folder.

Docker Images Provisioning

The Docker images definitions are contained within [images folder](#) established in the previous section. Respective folders named after the name of the aforementioned Agrofelis images contain a **Dockerfile** encoding the exact commands extending and adapting their related base image. The images folder provides one-line commands to easily provision the environments from scratch. The following example builds on the previously presented images.

```
cd /web-pub/connectivity/src/agrofelis-unificator/docker/images

./01.build.agrofelis.unificator.sh
./02.build.agrofelis.os.sh
./02.build.agrofelis.os.R.sh
./03.build.agrofelis.os.arduino.sh
```

Docker Images Instantiation

One folder above the images definitions path, named [docker](#) contains one-line commands instantiating the related containers based on their names as follows.

```
cd /web-pub/connectivity/src/agrofelis-unificator/docker
```

NginX

The following command instantiates the Nginx service and mounts the related setting files to the container:

```
./20.init.nginx.sh
```

Once the service is initiated, it will automatically start on systems boot or restart automatically if any issues arise.

Unificator

The following command will initiate the container running the Unificator Software. The instantiation command, apart from mounting the Agrofelis Unificator Node.js application source code, mounts privileged hardware resources namely the Arduino Mega steering system as well as that of the Front Sensors Lidar's USB connections. The command also exposes and links the port 8080 of the host machine with that of container's. Finally, it initiates the Unificator application using the Node.js technology and opens a [WebSocket](#) server on port 8080.

```
./01.init.agrofelis.unificator.sh
```

Supportive

To instantiate an ephemeral Ubuntu server as an evaluation environment, the following command can be issued.

```
./02.init.agrofelis.os.sh
```

Similarly, to instantiate an ephemeral Ubuntu server for conducting data manipulation experiments, the consequent command can be issued.

```
./04.init.agrofelis.os.R.sh
```

Finally, to compile and upload the firmware for the Arduino Mega steering system, the following command can be applied.

```
| ./03.agrofelis.arduino.compile.sh
```

Please note that when uploading firmware to the microcontroller, ensure that the USB connection is not already reserved by another application. To stop the Agrofelis Unificator Docker instance, the following command can be executed.

```
| docker rm -f agrofelis.unificator1
```

All Docker instances running on the host machine can be listed using the following command.

```
| docker ps
```

The consequent command can be used to monitor the hardware resources occupied by each container.

```
| docker stats
```

GPU and TPU Computing Modules

The Jetson Nano, apart from having a multicore processor, is also equipped with a 128-core Maxwell GPU module capable of implementing machine learning and machine vision tasks. This GPU allows for running multiple neural networks in parallel, making it suitable for applications such as image classification, object detection, segmentation, and speech processing ¹.

Another remedy equips the Agrofelis robot with additional machine learning capabilities, by employing the Coral Accelerator, which adds an Edge TPU coprocessor to the system, enabling high-speed machine learning inferencing ². The addition becomes highly efficient because Jetson Nano has one USB version 3 which can utilise the high speed throughput of the Coral Accelerator. The on-board Coral Edge TPU coprocessor is capable of performing 4 trillion operations (tera-operations) per second (TOPS), while using only 0.5 watts f per TOPS (2 TOPS per watt). For example, it can execute state-of-the-art mobile vision models such as MobileNet v2 at nearly 400 FPS, in a power efficient manner ³.

Being able to perform machine vision tasks at the edge, creates a robust system that operates independently of internet connectivity, enhancing its robustness and autonomy.

Agrofelis Unificator

The Agrofelis Unificator Software purpose, as its name suggests, serves the purpose of seamlessly integrating an arbitrary number of modules, regardless of their data streams, protocols, or bus types. The lightweight Agrofelis application was built on the NodeJS technology, due to its high performance and non-blocking I/O capabilities. The Unificator establishes connections with modules that utilize the USB interface, such as the Steering/Braking modules and the Front Sensors Lidar. Additionally, the application communicates with two WiFi power relay modules via the HTTP protocol. Lastly, the Unificator application establishes a connection with three modules utilizing the Websockets interface, namely the two MotorHubDrivers as well as the Remote Controller. The software moreover relays the information and routes commands to the appropriate module.

The Agrofelis Unificator server records unified data streams and rotates and archives them in compressed assets when they reach approximately 1 MB in their raw form. Notably, Lidar log data are stored in a separate file from the rest of the modules due to its significantly higher data rate compared to all the other modules combined. The application also tracks how far in time the archive folder has data for and erases older data batches to prevent occupying the entire disk space.

The application has been implemented within 200 lines of code, rendering the server easily maintainable. Using the overall recopy allows the Agrofelis modules to be operated as individual modules, but also as part of a collection. The technologies employed and the design principles, enable arbitrary hardware and software modules within the Agrofelis network to seamlessly integrate with the rest of the modules, thereby expanding the capabilities of the Agrofelis robot.

The Agrofelis Unificator Software is located in the following path.

¹ <https://www.raccoons.be/resources/insights/performance-comparison-:-coral-edge-tpu-vs-jetson-nano> "Performance comparison : Coral Edge TPU vs Jetson Nano"

² <https://coral.ai/products/accelerator> "Coral USB Accelerator"

³ <https://coral.ai/examples/> "Coral application examples"

- </web-pub/connectivity/src/agrofelis-unificator/server>

The software depends on few JavaScript libraries used for accessing the HTTP protocol, the Serial bus and the WebSockets interface, as well as for performing asynchronous web requests. These dependencies can be downloaded using the following commands.

```
apt-get update;  
apt-get install python3;  
npm install http serialport websocket axios
```

Multiple agents can connect to the Agrofelis Unificator server to observe, control, or supplement its operation. For instance, the Remote Controller can connect to the server simultaneously with another user observing data via the Unificator client-based web application. The following screenshot showcases the web application view in the aforementioned scenario.

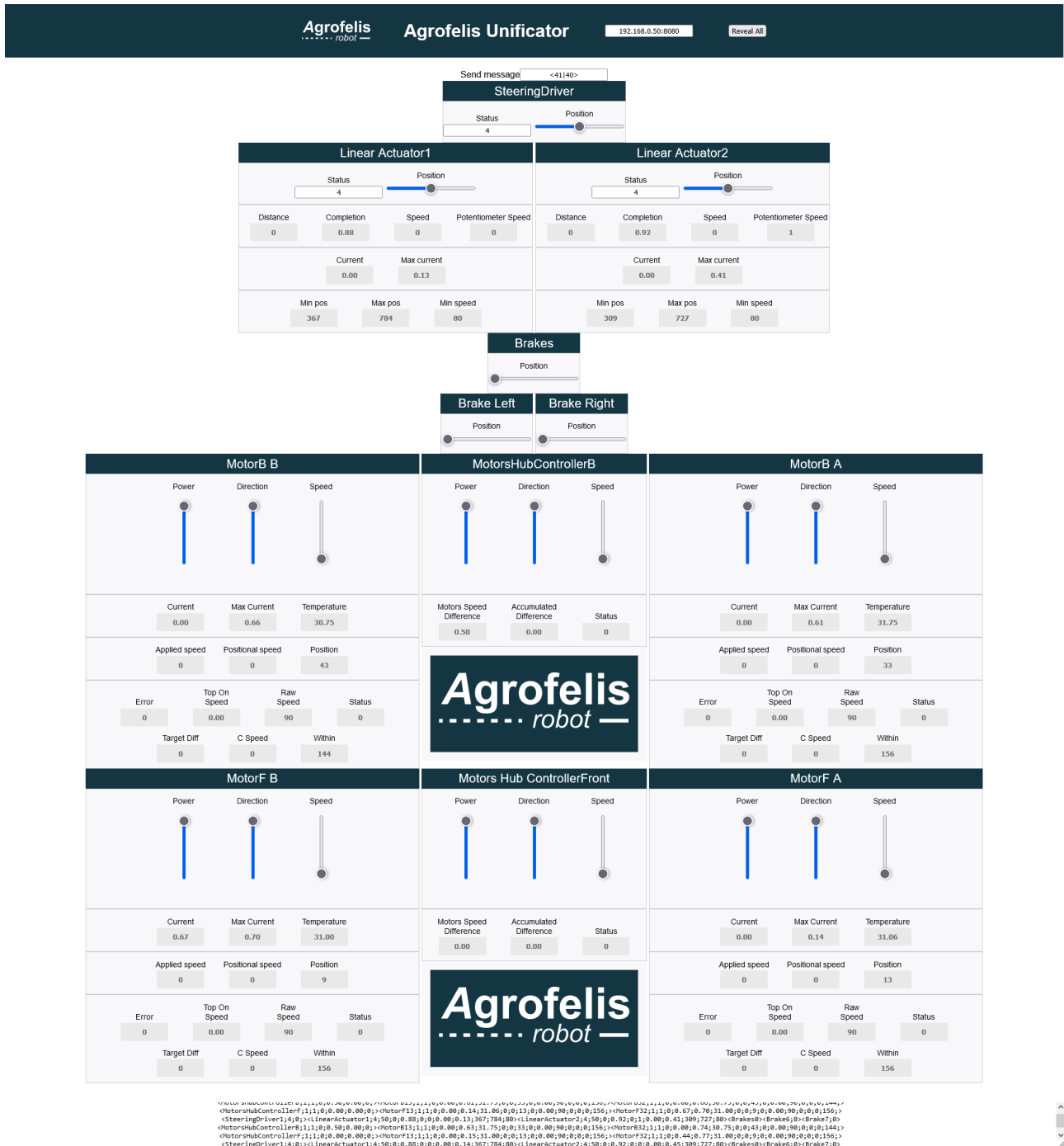


Figure 6: AgrofelisUnificator-snapshot

The Unificator web application, following the pattern established by the Motors Hub Controllers, enables the combination of multiple individual modules into one web application, providing a comprehensive overview of the entire system. The application source code can be accessed from the following path.

- [AgrofelisUnificator.html](#)

The HTML file also utilizes the following [assets](#):

File	Description
styles.css	Defines the CSS styles of the web application.
motorsHubController.js	Establishes a WebSocket connection with the related IP of the module. Parses the HTML to identify the related sensors and actuators. Listens for interface changes as well as WebSocket messages and accordingly reflects or submits the related information.
agrofelis_logo_white_web.svg	The scalable vector graphic logo of the project.
jquery.min.js	Minified JS library dependency jQuery .

Special care has been devoted to ensure that the setup and code are very lightweight, clean and straightforward in order to be easily modifiable, assisting the rapid prototyping.

Archived Data

The following screenshot shows the data log files archived by the Unificator server along with their size.

Filename	Filesize	Filetype	Last modified	Permissions	Owner/Group
data.2023-08-18-16-46-21.log.tar.gz	257,305	WinRAR ar...	18-Aug-23 7:46:21 PM	-rw-r--r--	root root
data.2023-08-18-16-47-28.log.tar.gz	257,695	WinRAR ar...	18-Aug-23 7:47:29 PM	-rw-r--r--	root root
data.2023-08-18-16-48-35.log.tar.gz	257,964	WinRAR ar...	18-Aug-23 7:48:36 PM	-rw-r--r--	root root
data.2023-08-18-16-49-43.log.tar.gz	257,787	WinRAR ar...	18-Aug-23 7:49:43 PM	-rw-r--r--	root root
data.2023-08-18-16-50-50.log.tar.gz	258,544	WinRAR ar...	18-Aug-23 7:50:50 PM	-rw-r--r--	root root
data.2023-08-18-16-51-57.log.tar.gz	258,429	WinRAR ar...	18-Aug-23 7:51:57 PM	-rw-r--r--	root root
data.2023-08-18-16-53-05.log.tar.gz	263,686	WinRAR ar...	18-Aug-23 7:53:05 PM	-rw-r--r--	root root
data.2023-08-18-16-54-28.log.tar.gz	134,743	WinRAR ar...	18-Aug-23 7:54:28 PM	-rw-r--r--	root root
data.2023-08-18-16-55-29.log.tar.gz	241,050	WinRAR ar...	18-Aug-23 7:55:29 PM	-rw-r--r--	root root
data.2023-08-18-16-56-33.log.tar.gz	244,182	WinRAR ar...	18-Aug-23 7:56:33 PM	-rw-r--r--	root root
data.2023-08-18-16-57-38.log.tar.gz	251,041	WinRAR ar...	18-Aug-23 7:57:38 PM	-rw-r--r--	root root
data.2023-08-18-16-59-24.log.tar.gz	184,638	WinRAR ar...	18-Aug-23 7:59:24 PM	-rw-r--r--	root root
data.2023-08-18-17-00-36.log.tar.gz	19,781	WinRAR ar...	18-Aug-23 8:00:36 PM	-rw-r--r--	root root
data.2023-08-18-17-01-24.log.tar.gz	175,187	WinRAR ar...	18-Aug-23 8:01:24 PM	-rw-r--r--	root root
data.2023-08-18-17-02-30.log.tar.gz	260,226	WinRAR ar...	18-Aug-23 8:02:31 PM	-rw-r--r--	root root
data.2023-08-18-17-03-37.log.tar.gz	263,856	WinRAR ar...	18-Aug-23 8:03:38 PM	-rw-r--r--	root root
data.2023-08-18-17-04-45.log.tar.gz	261,735	WinRAR ar...	18-Aug-23 8:04:45 PM	-rw-r--r--	root root
data.2023-08-18-17-05-52.log.tar.gz	262,911	WinRAR ar...	18-Aug-23 8:05:52 PM	-rw-r--r--	root root
data.2023-08-18-17-06-59.log.tar.gz	263,079	WinRAR ar...	18-Aug-23 8:07:00 PM	-rw-r--r--	root root
data.2023-08-18-17-08-07.log.tar.gz	262,718	WinRAR ar...	18-Aug-23 8:08:07 PM	-rw-r--r--	root root
data.2023-08-18-17-09-14.log.tar.gz	262,367	WinRAR ar...	18-Aug-23 8:09:14 PM	-rw-r--r--	root root
data.2023-08-18-17-10-21.log.tar.gz	263,911	WinRAR ar...	18-Aug-23 8:10:21 PM	-rw-r--r--	root root
data.2023-08-18-17-11-29.log.tar.gz	263,042	WinRAR ar...	18-Aug-23 8:11:29 PM	-rw-r--r--	root root
data.2023-08-18-17-12-36.log.tar.gz	262,454	WinRAR ar...	18-Aug-23 8:12:36 PM	-rw-r--r--	root root
data.2023-08-18-17-13-43.log.tar.gz	260,219	WinRAR ar...	18-Aug-23 8:13:43 PM	-rw-r--r--	root root
data.2023-08-18-17-14-50.log.tar.gz	260,811	WinRAR ar...	18-Aug-23 8:14:51 PM	-rw-r--r--	root root

Figure 7: Archived Log DataImage

The subsequent screenshot showcases the content of the two files contained in each archive, one for the Lidar data and one for all other modules.

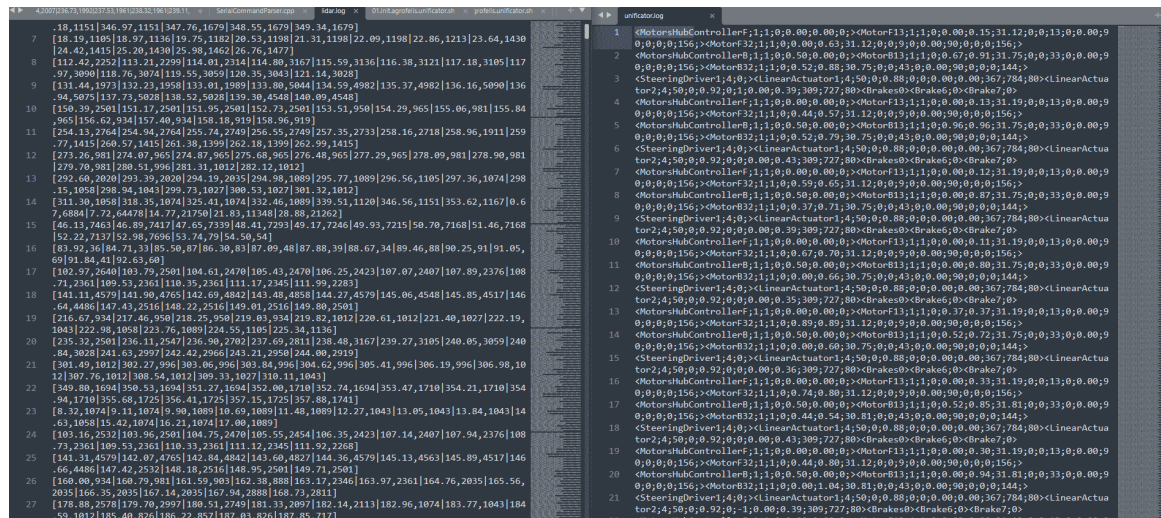


Figure 8: UnificatorAndLidarData

Agrofelis Network

In order to interconnect the various WiFi-capable modules of the robot, a compact access point module was employed. More specifically, the TP-LINK TL-WR802N v4 wireless router, supporting up to 300Mbps bandwidth, was chosen. This module was chosen due to its compact size, its capacity to operate in multiple modes as well as its 5V compatibility.



Figure 9: Wifi router

The wireless router is configured as a wireless router leading to a stable WiFi network and is used to associate the different modules with static IP addresses based on their unique MAC addresses. The WiFi network is secured with a password, which is also set for modules wishing to connect to the Agrofelis Network.

The default WiFi password followed by all WiFi modules of this repository is:

`felisagrofelistobor_`

The complete configuration of the WiFi router, as exported by its web application, has been persisted in the following path.

- [tplinkWR802N_conf_wirelss_Router.bin](#)

Additionally, the module was evaluated in “wisp” mode, allowing it to relay the WiFi of a mobile phone hot-spot to the Agrofelis Network. In this mode, the network’s stability depended on the mobile internet connection and was avoided during field missions. Instead, it was primarily used to provide internet access to the system during maintenance.

The complete configuration of the WiFi router in “wisp” mode, as exported by its web application, has been persisted in the following path:

- [tplinkWR802N_conf_wisp.bin](#)

The two modes can be swiftly interchanged by uploading the respective configuration file to the wireless router’s web interface.

The attached configuration files also encode the IP associations of each WiFi module, according to its MAC address which is unique in the world. Therefore, these mappings should be changed accordingly after observing the MAC address of the employed WiFi-capable modules. Below, the association map defined for our devices, is provided as an example.

MAC Address	IP Address	Name
48:B0:2D:2F:25:FA	192.168.0.50	Jetson Nano
A0:20:A6:2E:C8:8C	192.168.0.51	Power module A
60:01:94:7D:1C:BF	192.168.0.52	Power module B
A0:B7:65:61:53:84	192.168.0.55	Motors hub back
A0:B7:65:61:7D:7C	192.168.0.56	Motors hub front
24:0A:C4:1F:CF:50	192.168.0.60	Remote controller

Agrofelis USB Hub

A compact 5V USB hub is employed to enable multiple devices to be connected to the Jetson Nano. Specifically, the following modules are connected to the hub, which is subsequently connected to the USBv2 port of the computer.

- WiFi USB
- BMS USB sensor cable
- WiFi router
- Agrofelis steering module
- Agrofelis front Lidar sensors

Structural Support for Computing Modules

One often-overlooked aspect when arranging components is that their connection cables can occupy up to 50% more of their size. Although various connection cables can be made compact, this is not an option for USB cables, which could otherwise obstruct neighboring components. The solution designed to mitigate the identified problem, was four 3D-printed compact tilted mounts, allowing to pass the cables above the neighboring elements and to pass the cables through the space created underneath.

The following schematic illustrates the location of the computing modules, their extensions, the USB hub and the 3D-printed tilted mounts positioned above the steering plates in the back section of the robot.

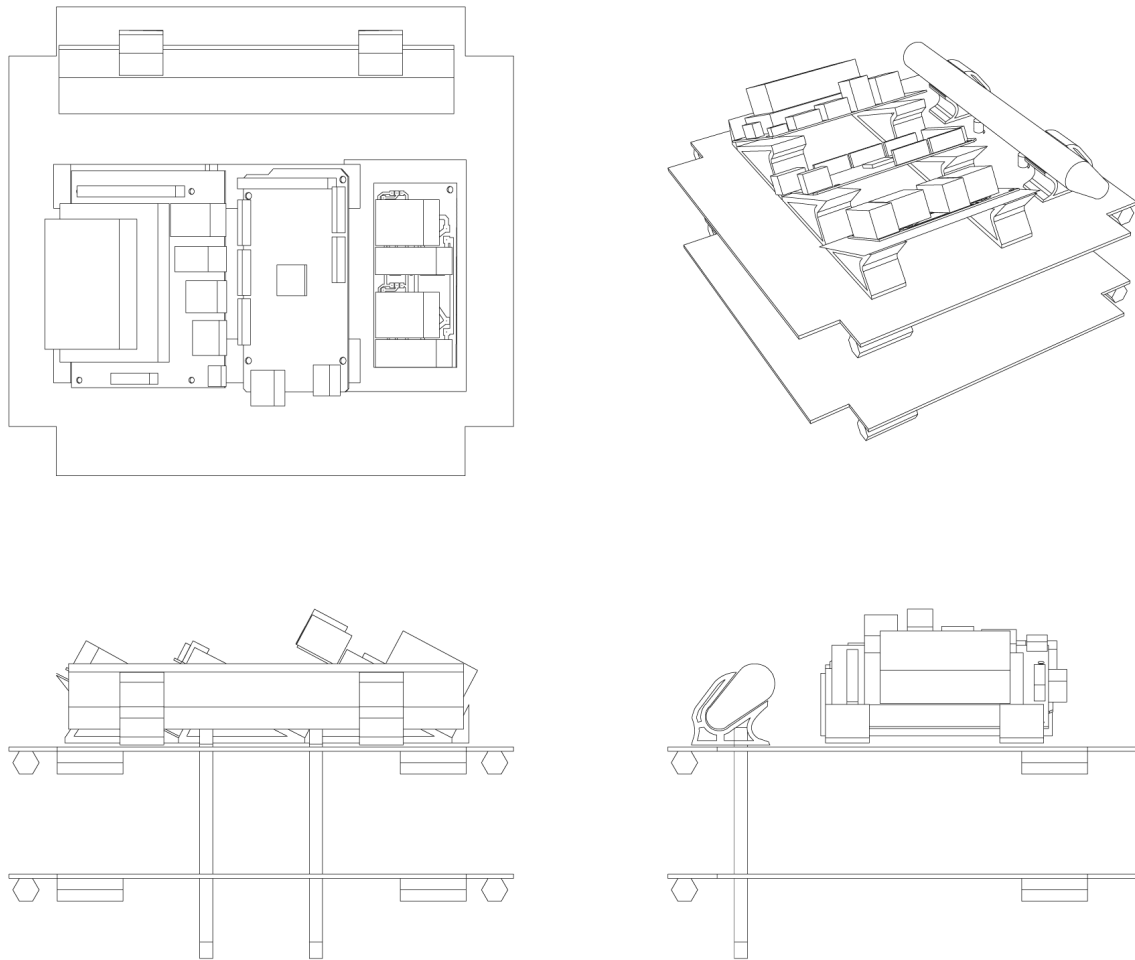


Figure 10: computing elements schematic

Two types of structures were designed and 3D-printed twice. The brackets and their arrangement on the steering plate can be seen in the following diagram.

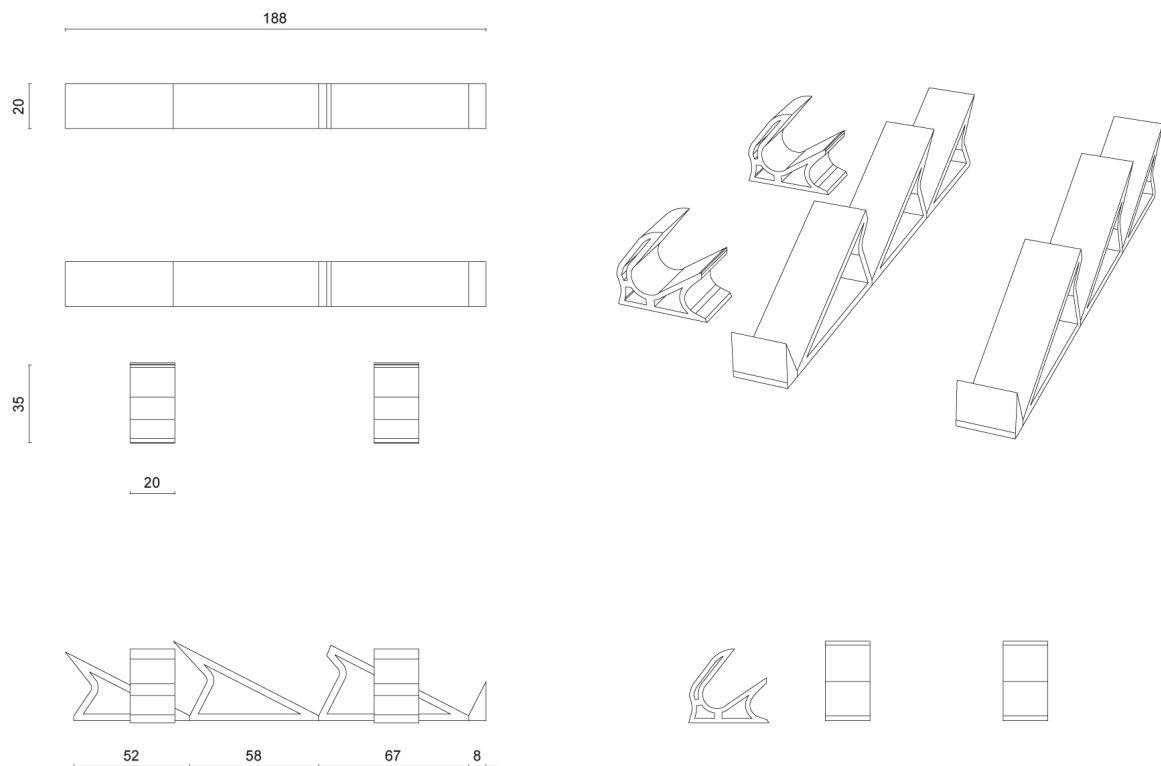


Figure 11: 3d printed brackets

The 3D printing source code files for manufacturing the related brackets, are listed by the following table.

Name	Standard Triangle Language file	Geometric Code
Computing modules tilted brackets	STL	Gcode
USB hub titled brackets	STL	Gcode

The following image shows the computing modules supportive structure and how the Jetson Nano is attached to it.



Figure 12: 3d print brackets-a

The use of magnets located on their bottom side, enables the structure to snap into place when in contact with the iron steering plate. The use of magnets also provides the ability to slightly relocate the modules if necessary. Despite the magnets not being strong enough to completely secure all components in place, the additional forces of multiple cables interconnecting the modules with the frame help keep the modules in place.

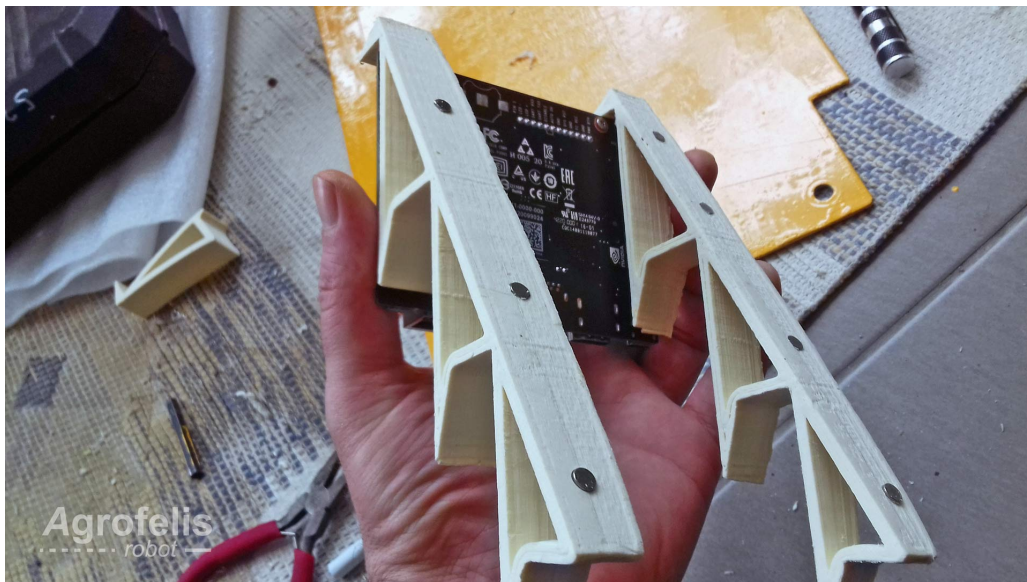


Figure 13: 3d print brackets-b

The following image showcases the Jetson Nano and the Arduino Mega attached to the related 3D-printed brackets.

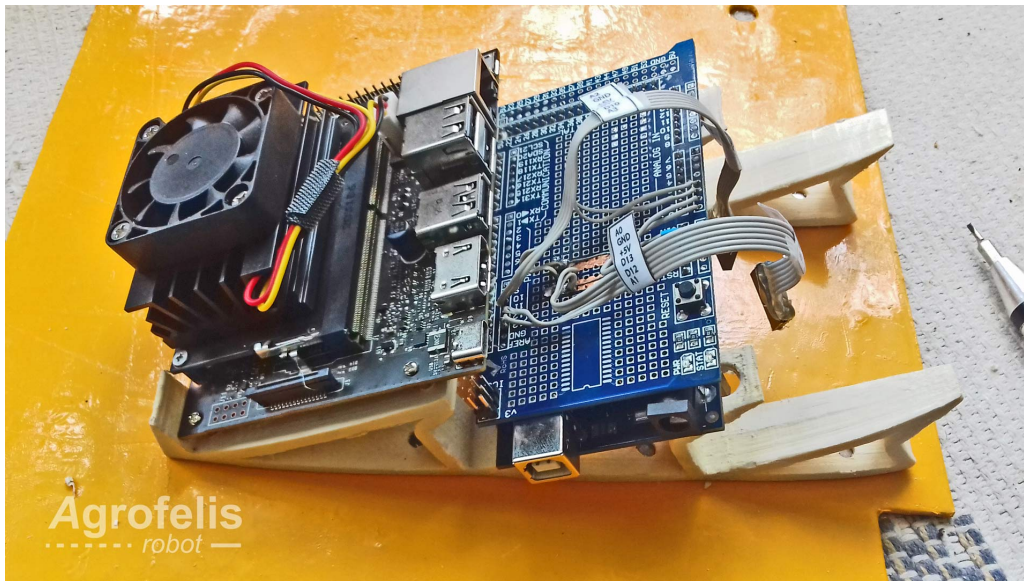


Figure 14: 3d printbrackets-c

A side view of the compilation demonstrates the degree of tilt, how a USB cable to be easily plugged into the Jetson Nano without being obstructed by the presence of the Arduino Mega. Also, the figure illustrates the space created to pass cables under the components towards the USB hub, and the minimum amount of material used to fabricate the brackets.



Figure 15: 3d print brackets-d

The following photos snapshot the computing modules' structure installed in the relevant compartment of the Agrofelis robot.

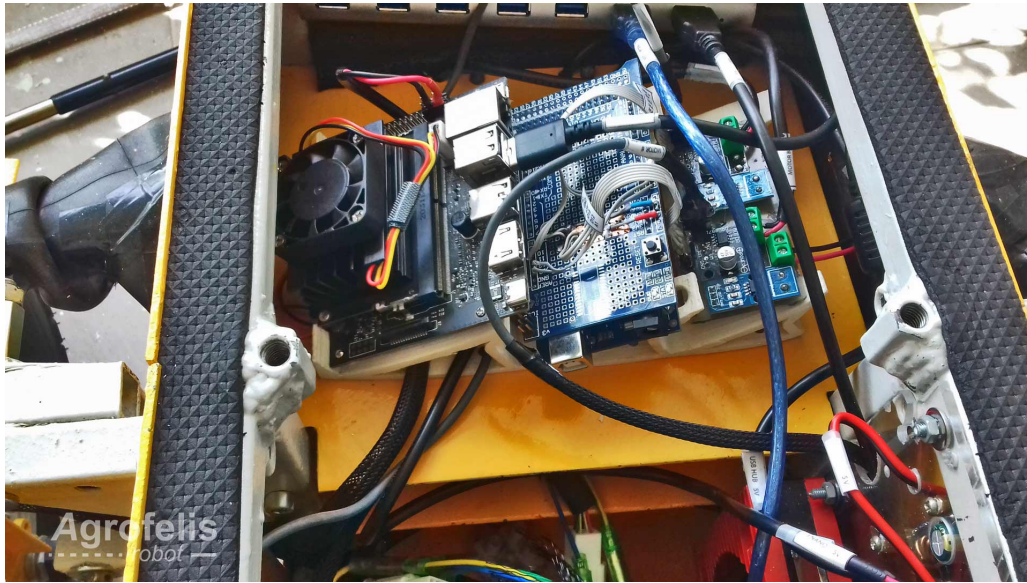


Figure 16: elements view-a

A front view photo of the compartment with its elements is provided below.

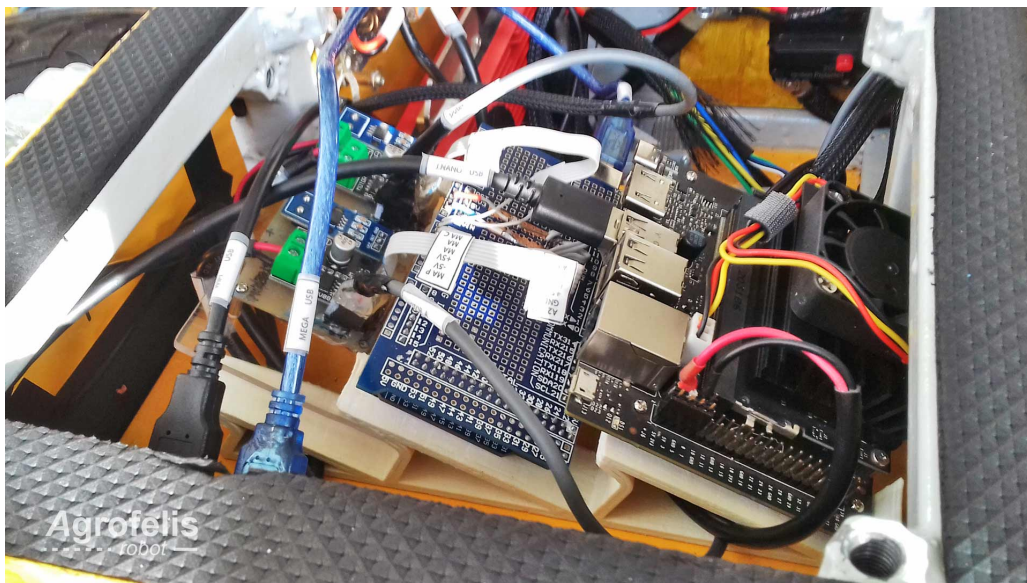


Figure 17: elements view-b

Computing and Network Components

The following table lists the individual components employed for manufacturing the Agrofelis computing modules and networking system. The index table includes moreover the product URLs, the indicative suppliers, as well as unit prices and sum totals.

No.	Product	Product URL	Supplier	Used Quantity	VAT Price (€)	Subtotal (€)	Note
#1	NVIDIA Jetson Nano Developer Kit	Jetson Nano	Hellas Digital	1	199.00	199.00	-

No.	Product	Product URL	Supplier	Used Quantity	VAT Price (€)	Subtotal (€)	Note
#2	High speed ADATA premier one 128gb	Hard Disk	Skrouzt	1	81.09	81.09	-
#3	Jetson nano fan	CPU Fan	Hellas Digital	1	8.00	8.00	-
#4	Jetson nano case	Jetson Nano Case	GRobotronics	1	15.80	15.80	Keeping latching buttons and camera mount
#5	Coral USB Accelerator	Coral	Nettop	1	87.90	87.90	-
#6	8 MP night vision camera	Camera	Hellas Digital	1	31.99	31.99	-
#7	TP-LINK TL-WR802N v4 Wireless Router	Wireless Router	Skrouzt	1	24.30	24.30	-
#8	15 Pin Ribbon Flex CSI Cable 100cm	Flex CSI Cable	Aliexpress	1	0.67	0.67	-
#9	7 port USB v3.1 HUB	USB Hub	Skrouzt	1	36.59	36.59	-
Total						485.34	

The total cost to manufacture the Agrofelis computing modules and network, exclusive of shipping and labor costs, amounts to approximately **485** euros.

Summary

The document presented an overview of the computing elements utilized in the Agrofelis robot. The developed services containerized via Docker were described and indexed. Additionally, the Agrofelis Unificator Software, which connects all Agrofelis modules, was introduced and its source code and features were presented. The lightweight 3D-printed structural elements designed for organizing the components and their cables source code files, were also provided. The document concluded with a list of the components used and indicative suppliers, aiding the replication of the Agrofelis computing and networking systems.

End of document