D4.7 User Manual
Version 2.1
November 23, 2023

Version History

Version	**Date**	**Description**	**Author**
0.1	September 10, 2020	First Draft	Alexandre Corizzi
0.2	September 16, 2020	Feedback from Clémence Goyens	Alexandre Corizzi
0.3	September 22, 2020	Notes from Clémence Goyens	Alexandre Corizzi
0.4	September 24, 2020	Comments from Javier Concha	Alexandre Corizzi
0.5	September 29, 2020	Draft Notebook Tests Section	Clémence Goyens
0.5b	October 1, 2020	Pan/Tilt RS485 Config + FTP Link Correction	Alexandre Corizzi
0.5c	October 1, 2020	Feedback from Joel Kuusk	Alexandre Corizzi
1.0a	March 29, 2021	Update with new software	Alexandre Corizzi
1.0b	May 27, 2021	Timezone and other minor changes	Alexandre Corizzi
1.0c	June 9, 2021	Add Yoctopuce annex	Alexandre Corizzi
1.1a	July 15, 2021	Switch to Beta Version & Yoctopuce Update	Alexandre Corizzi
1.1b	July 16, 2021	Comments by David Doxaran	Alexandre Corizzi
1.1c	July 28, 2021	Comments by Joel Kuusk	Alexandre Corizzi
2.0	February 21, 2023	Add Mounting Illustrations & Debian	Alexandre Corizzi
2.1	April 21, 2023	Update GUI screenshots, metadata configs	Alexandre Corizzi
2.2	July 17, 2023	Update Debian section	Alexandre Corizzi
2.2b	November 23, 2023	Disambiguation of DIO wiring	Ana I. Dogliotti

Contents

1 Executive summary **5**

2 Mounting Guide **6**
2.1 Material List . . . 6
2.2 Flip the metal piece on the top of the instrument . . . 9
2.3 The Inside of the Studiel Box . . . 11
2.3.1 Yocto-Pictor - Dissasembling & Light Sensor . . . 11
2.3.2 Yocto-Pictor - Reassembling in the Box . . . 12
2.3.3 Rugged PC and Serial Board Converter fixing . . . 13
2.4 Finalization . . . 15

3 Installing a Linux Distibution **17**
3.1 Debian Linux System . . . 17
3.1.1 Creating an Installation USB of Debian Linux . . . 17
3.1.2 Indicative settings for the Installation . . . 17

4 Hypernets Tools **19**
4.1 Presentation . . . 19
4.2 Installation . . . 19
4.3 Configuration Files . . . 20
4.4 Common Examples of Command . . . 20
4.4.1 Driving the Yocto-Pictor . . . 20
4.4.2 Moving the Pan-Tilt . . . 21
4.5 Protocol Files . . . 21
4.5.1 Geometry . . . 22
4.5.2 Measurements . . . 22
4.5.3 Examples . . . 22

5 First tests **23**
5.1 Prior Set-up of the System and First tests . . . 23
5.2 Validation Module Testing . . . 23

6 Autonomous System Mode **25**
6.1 Set Relay n°1 ON at Wake-up . . . 25
6.1.1 Enable SSH access . . . 25
6.2 Try a full sequence acquisition . . . 26
6.3 Setup Server Communication . . . 26

7 Annexes **27**
7.1 Configuration file: static_config.ini . . . 27
7.1.1 Section: yoctopuce . . . 27
7.1.2 Section: network . . . 27
7.2 Configuration file : dynamic_config.ini . . . 27

7.2.1 Section: general . . . 27
7.2.2 Section: GPS . . . 27
7.2.3 Section: pantilt . . . 28
7.2.4 Section: metadata . . . 28
7.2.5 Section: hypstar . . . 29
7.3 How To Access to the BIOS of the Cincoze . . . 30
7.3.1 Pan-Tilt: Serial Port Configuration . . . 30
7.4 Yoctopuce Tips . . . 31
7.4.1 Upgrade Yocto-Pictor firmwares . . . 31
7.4.2 How to catch debug informations . . . 31

8 Old Versions Documentation **33**
8.1 Manjaro Linux System . . . 33
8.1.1 Creating A Live USB Of Manjaro Linux . . . 33
8.1.2 Install and Update Manjaro Linux . . . 34
8.2 Wi-Fi Connection Between the Yocto-Pictor and the rugged PC . . . 35
8.2.1 Wi-Fi Hotspot Set-up . . . 35
8.2.2 Setting up the Yoctopuce VirtualHub . . . 36
8.2.3 Yocto-Pictor Wi-Fi Connection . . . 37
8.3 Linking Hypernets Tools to the Yocto-Pictor . . . 38
8.3.1 Radiometer Configuration and First tests with the GUI . . . 39

1 Executive summary

This document aims to guide step by step the user to perform the installation of a Hypernets System. Firstly, a basic illustrated guide to assemble the different mechanical and electronic parts of the system will take place. Secondly, a tutorial about the installation of the a Linux operating system will be described. This, including both Manjaro and Debian Linux on the embedded computer (rugged PC). In third place, the basic set-up and some essential tests to move the pan-tilt - motor part of the system - and making some Hypstar Radiometer measurements will be provided. Finally, a procedure to make automatic measurements at arbitrary times - the autonomous mode - will be described.

2 Mounting Guide

2.1 Material List

Provider	**Designation**	
Tartu	Pan-tilt, Instrument & Validation Module	Right part of the Figure 1
	RS485 Serial Board Converter	Green Card in the Figure 1
	Alignement Tool	Figure 19
	Anti-Dust Cap	Figure 20
	Mechanical Part	Metal parts in the Figure 1
Studiel	Main Box	Figure 3
	Ancillary Box	Figure 4
	Junction Box	Figure 5
	1 Green Samtec Cable (Main box – Junction box)	Figure 2
	1 Orange Samtec Cable (Main box – Ancillary box)	Figure 2
	2 Yellow Samtec Cables (Webcams)	
	1 Blue Samtec Cable (Power System – Main box)	Figure 2
	2 Ray tech Rapid Joint	
Yoctopuce	Yocto-Pictor	Figure 9
N/A	Rugged PC Cincoze	Left part of the Figure 1
	Power system	
	2 Webcams Wi-Fi TVIP ABUS 62561	

Figure 1: Yoctopuce, Rugged PC and Hypstar & Validation Module

Figure 2: Studiel Cables

Please note that the mechanical mount for pan-tilt and the powering system are not supported by this guide.

Figure 3: The Studiel main Box

Figure 4: The Studiel Ancillory Box

Figure 5: Studiel Junction Box

2.2 Flip the metal piece on the top of the instrument

Required time: 10 min

For shipping reasons – size of the package, avoiding shipping damages – the Hypstar Radiometer & the Validation Module are shipped with the top metal wing mounted in the wrong way. First, fix the pan-tilt on your mechanical mount. Unscrew the 4 screws (labelled in red on 6) and invert the direction of the wing like showed in the following pictures.

Material List:

- Allen key

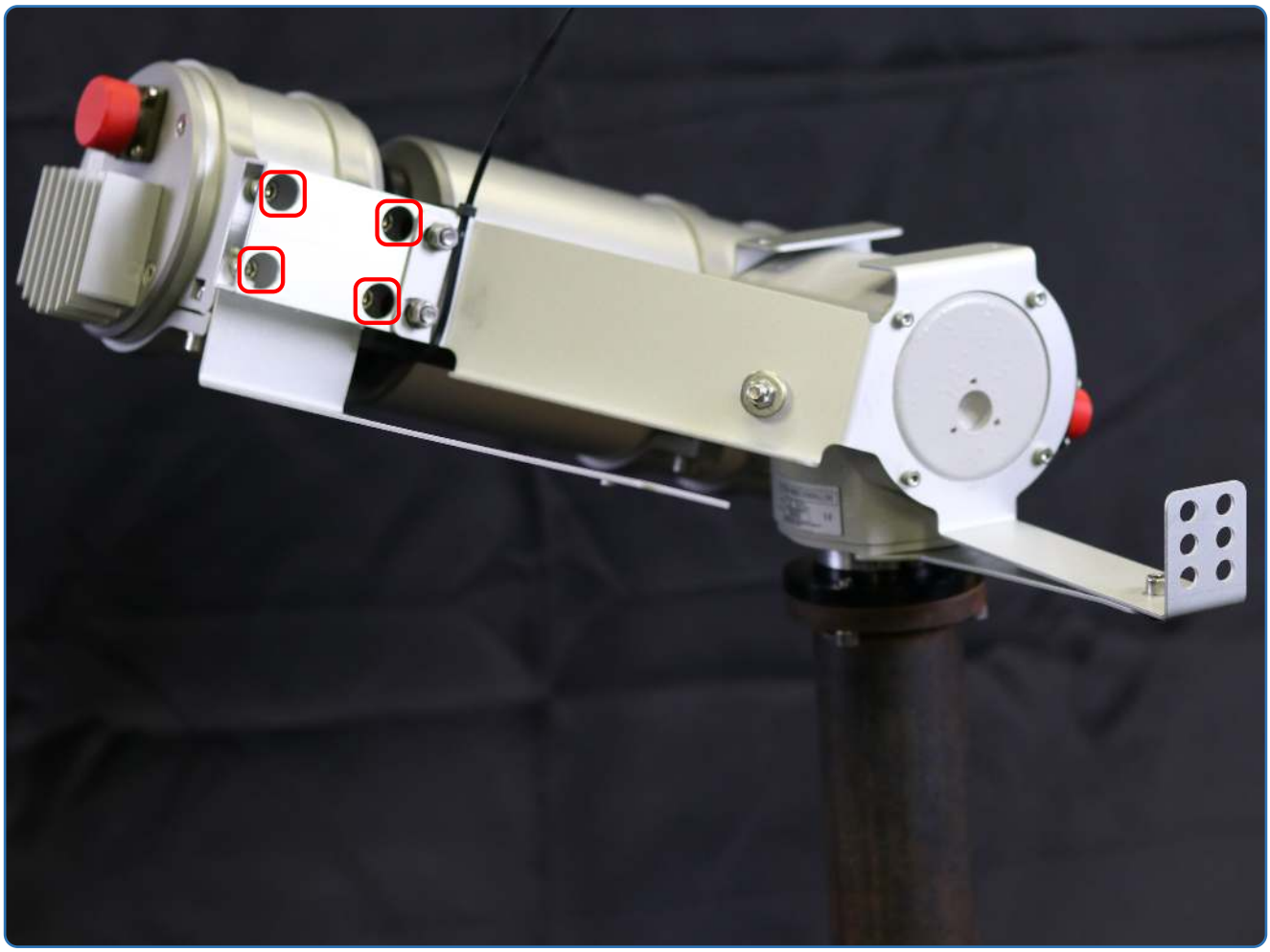

Figure 6: Flip metal piece 1/3

Figure 7: Flip metal piece 2/3

Figure 8: Flip metal piece 3/3

2.3 The Inside of the Studiel Box

Required time: 30 min

Material List:

- Allen key 2 and 5, Plate key 7
- Screw drivers (flat/slotted and cross slot)
- Cutting plier

2.3.1 Yocto-Pictor - Dissasembling & Light Sensor

Instructions:

1. Unscrew the four screws at the top of the Yocto-Pictor
2. Split the Yoctopuce in two pieces (see figure 9)
3. Cut the light sensor of the ground floor
4. Fix it and plug it in the Studiel ancillary box (see figure 10)

Warning: make sure to correctly cut the PCB track of the light sensor before pulling it; there is a risk of ripping it of.

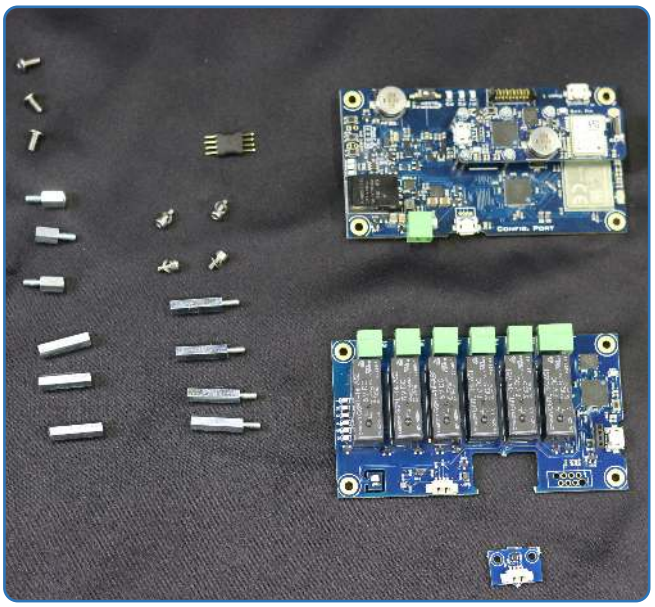

Figure 9: Dissasembling the Yocto-Pictor

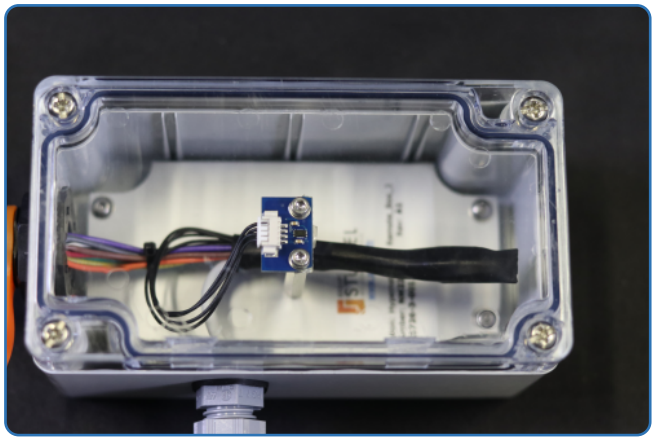

Figure 10: Placement of the Light Sensor

2.3.2 Yocto-Pictor - Reassembling in the Box

Instructions:

1. Place the Yocto-Pictor ground floor with relays on the top and the 4 screw tubes (figure 11)
2. Plug the bottom-top bridge and add carefully the top floor and fix it with screws (figure 12)
3. Fix and connect the GPS antenna; then follow the wiring Table 1

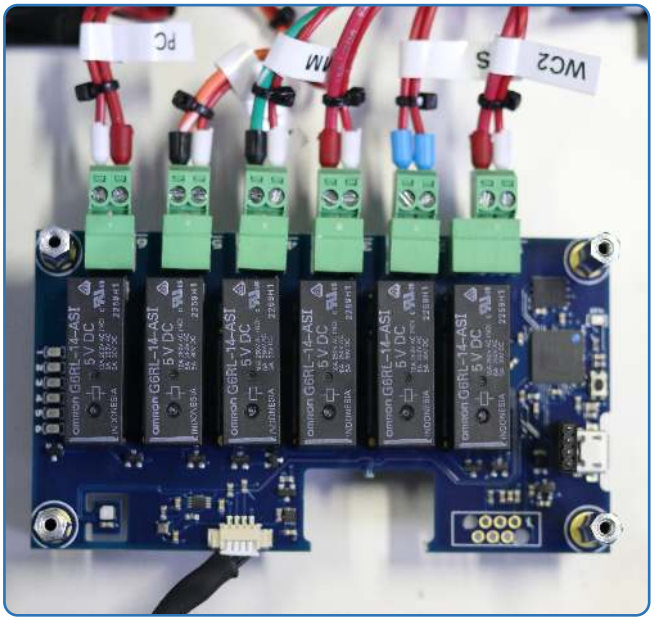

Figure 11: Fixing the bottom part

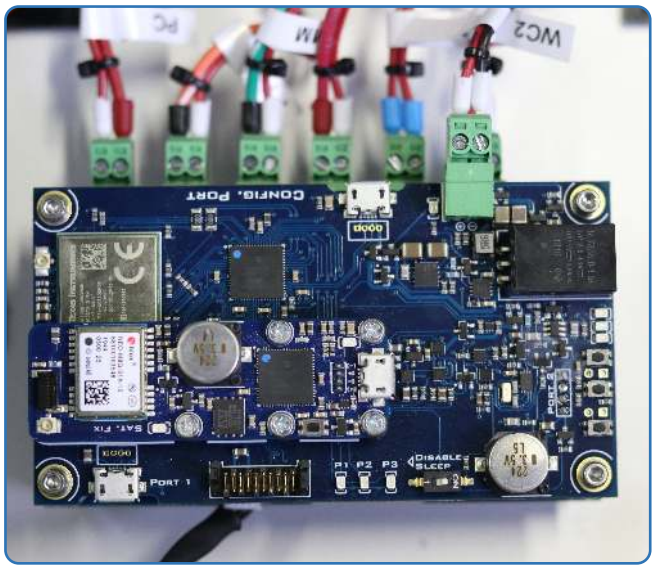

Figure 12: Fixing the top part

Yocto-Pictor Pinouts	**Studiel Box Connectors**
Relay 1	PC (Rugged PC power)
Relay 2	PT (Pan tilt)
Relay 3	MM (Measurement Module)
Relay 4	RS (Rain Sensor)
Relay 5	WC1 (Webcam 1)
Relay 6	WC2 (Webcam 2)
Main Power	YP (Studiel Box Connector)
Light Sensor Connector	Dedicated tablecloth USB
USB config Port	USB-B Micro cable

Table 1: Wiring table for the Yocto-Pictor

2.3.3 Rugged PC and Serial Board Converter fixing

Instructions:

1. Attach the fixation support provided with the rugged PC, the grids upwards (see the Figure 13)
2. Unscrew the four srews (shown on the Figure 14)
3. Fix the rugged PC ; the side with DIO and the four serial ports must be toward the top of the Studiel Box
4. Attach the serial board Converter and plug the brown cable in the slot A, the red cable in the slot B (Figure 16)

Figure 13: Rugged PC - Fixation System

Figure 14: Area for the Rugged PC in the Box

Now, you should have connectors pending next to their rugged PC pinouts that you can plug:

- Connect the two DIO terminals as shown on the Figure 18
- Connect the RS485 wire (the one of the top) to the COM1
- Connect the power supply and the Ethernet connector
- Connect the USB cable for RS485 converter (HYPSTAR – Rugged PC connection)
- Connect the USB cable for Yocto-Pictor

Unscrew the top of the Junction Box and fix it on the top of the pan-tilt using the provided srews (see Figure 17).

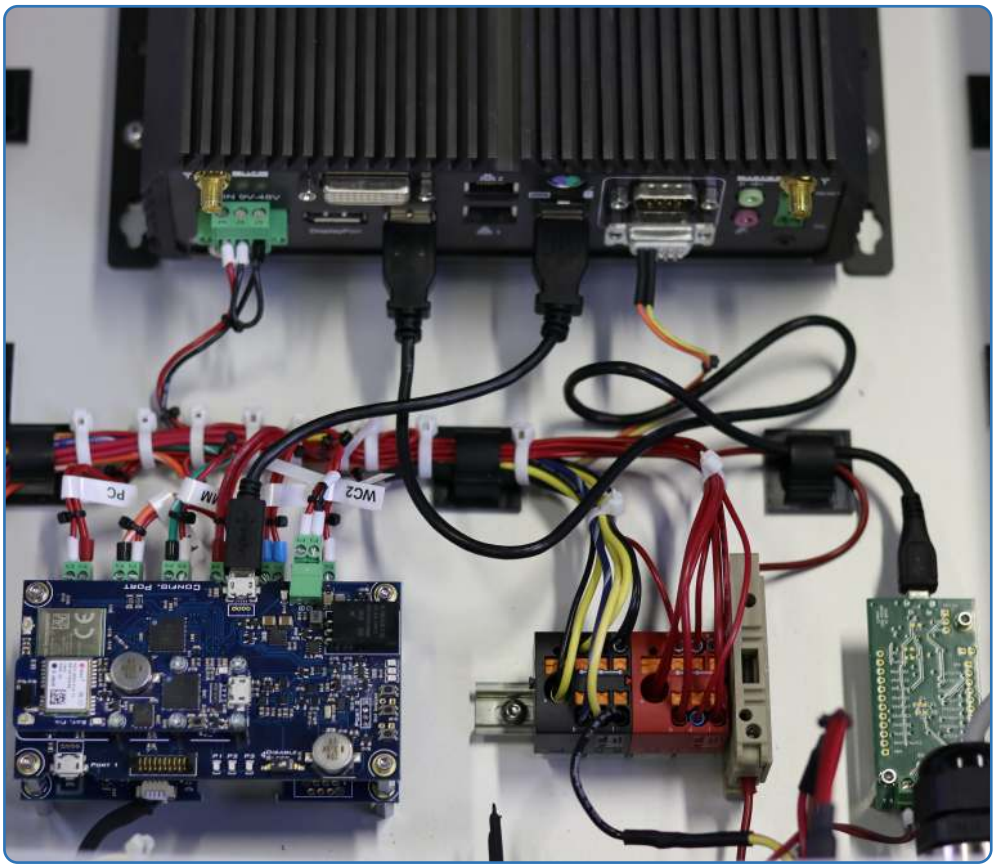

Figure 15: Global view of the inside

Figure 16: Serial Board Converter

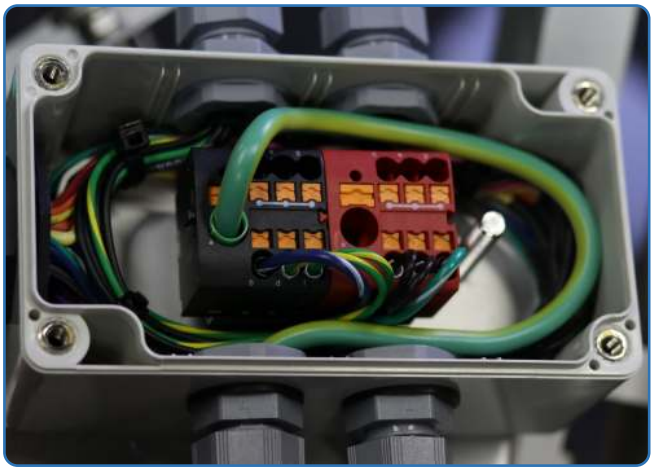

Figure 17: Fixing the Junction Box

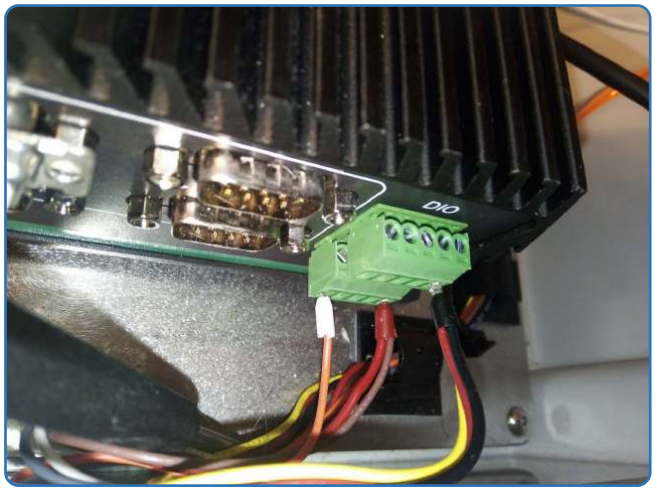

Figure 18: Wiring the DIO

2.4 Finalization

Instructions:

1. Carefully remove the Alignement Tool (Figure 19)
2. Place the anti-dust device (Figure 20)
3. Carefully respecting the color code, connect the cables
4. Make sure the rugged PC switch is on *AT*
5. Close the box

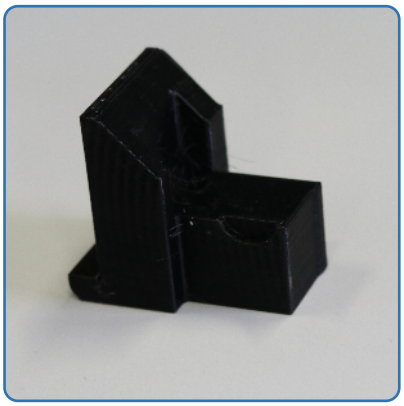

Figure 19: Remove the *Alignement Tool*

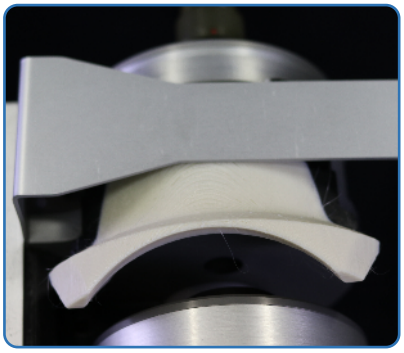

Figure 20: Put the anti-dust cap

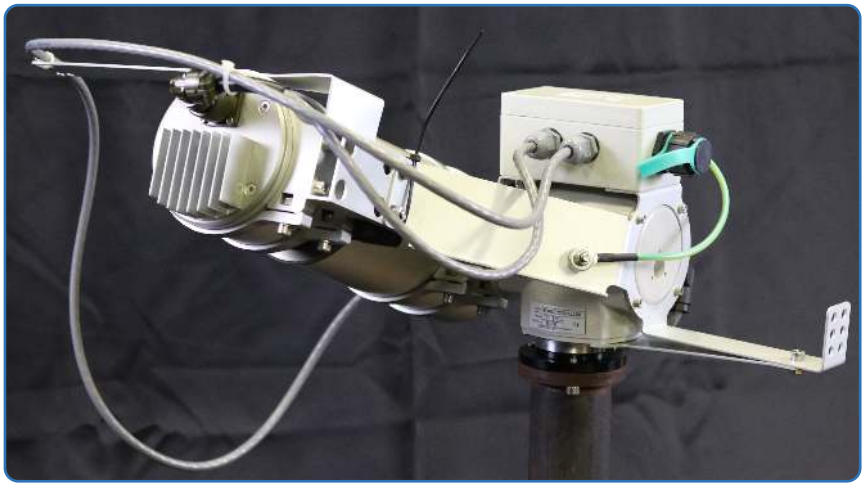

Figure 21: Fixing the cables

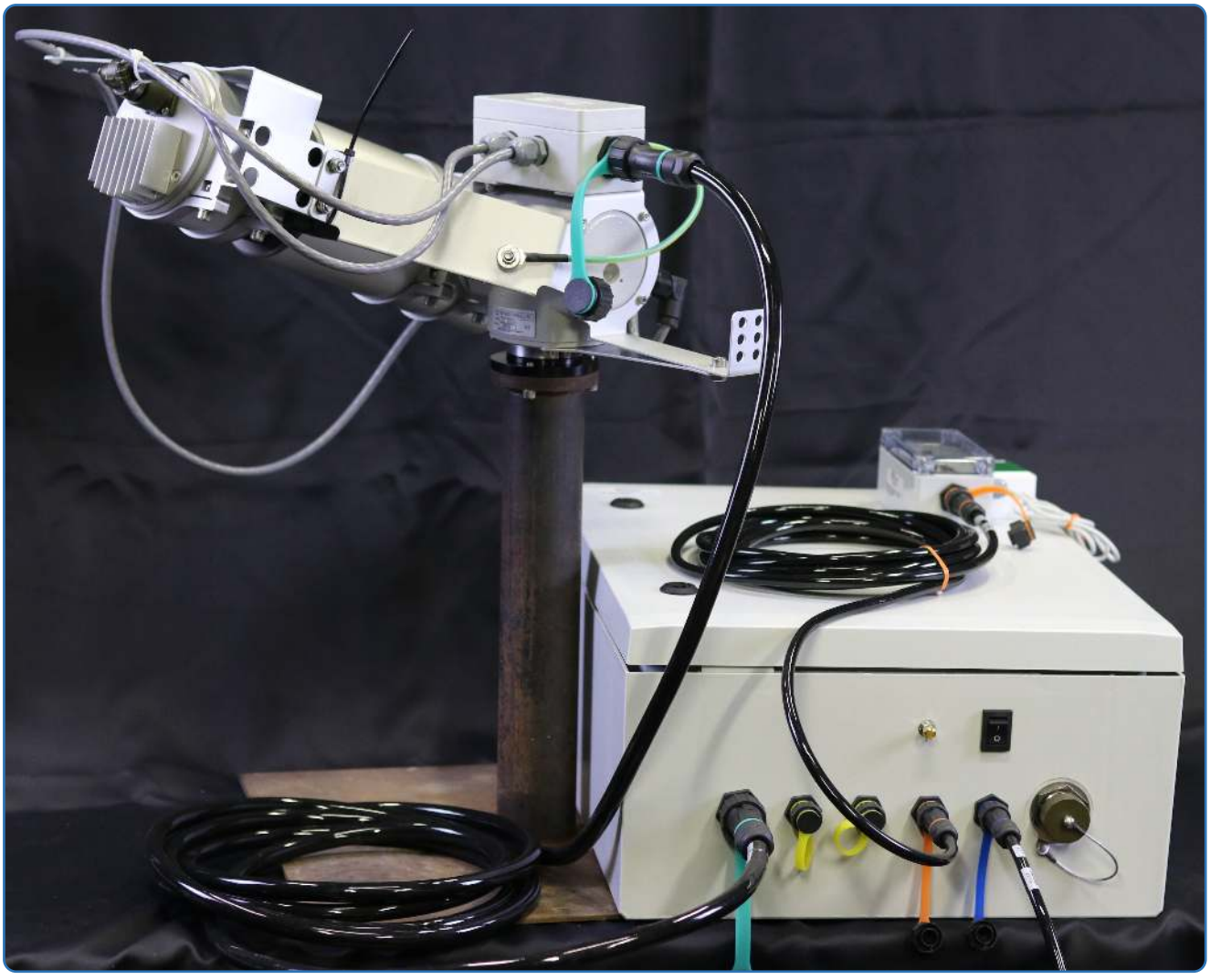

Figure 22: Overall Installation

3 Installing a Linux Distibution

Required time: 1 h 30

This part aims to install a fresh linux system on the rugged PC. Most of the first tests have been performed with *Manjaro* Linux. However, any UNIX system could replace this distribution. We also have experimented few updating issue with Manjaro as it is a *rolling release distribution*. Hence the last version of the software has been tested with *Debian* that might be the prefered solution.

Material List:

1. Rugged PC Cincoze
2. USB Stick up to 2GB
3. Internet connection
4. Screen, keyboard, and mouse

3.1 Debian Linux System

3.1.1 Creating an Installation USB of Debian Linux

Alternatively you can use a Debian System, visit this link and download those two files:

```
debian-11.7.0-amd64-netinst.iso
SHA256SUMS
```

Check-up the integrity of your downloaded file with:

```
sha512sum -c SHA512SUMS.txt
debian-11.7.0-amd64-netinst.iso: Success
```

The next step is to burn the image with:

```
umount /dev/sdX*
sudo dd if=/path/to/debian.iso of=/dev/sdX status="progress"
(replace sdX with the ID of your usb stick)
```

Double check your device name before writing onto it (please refer to the equivalent 8.1.1 in the Manjaro section).

3.1.2 Indicative settings for the Installation

1. Boot on the USB stick (see Annex 7.3 to configure the BIOS)
2. Choose the *Graphical Install* and the language *English*
3. Set *United Kingdom* as location and select your keyboard mapping

4. Configure a Network connection for the installation
5. Set up machine name, user and root passwords
6. Force UEFI installation: *yes*
7. Select the *guided partioning* with all files in one partition (recommended)
8. Participate in the package usage survey: *no*
9. Select the **Xfce** server X and add *standard system utilities*

Once done, remove the USB key stick and reboot. You should see a prompt asking you the password you previously setted-up. Open a terminal *(ctrl + alt + t)* and type the following commands:

```
rmdir Music/ Pictures/ Videos/ Public/ Templates/
su
<password>
/usr/sbin/usermod -aG sudo <username>
/usr/sbin/usermod -aG adm <username>
```

Logout, login and install git with:

```
sudo apt install git
```

4 Hypernets Tools

4.1 Presentation

hypernets_tools is the software developped for the Hypernets System. It consists in a *Git* repository, it is mainly developped with Bash and Python (3.7+) languages. It is fully compatible with any UNIX system which inclues *apt* or *pacman* and a functionnal *systemd* installation. It is also dependent on the *libhypstar*, the *driver* library of the Hypstar. *hypernets_tools* is designed in the form of a python package, containing several folders, grouped by thematic. Most of the scripts contained within those folders are *callable* from a terminal and are *self explanatory* thanks to the *help* command. In addition, a *Graphical User Interface* is built on top of those scripts, allowing to use some basic functionnalities of the system.

The installation of the *hypernets_tools* is made using the *distributed version control system Git*. If you aren't familiar with this tool, you can refer to the the wikipedia of Git or its documentation.

4.2 Installation

First clone the repository of the project with:

```
git clone https://github.com/hypernets/hypernets_tools
```

Check-out the appropriate branch with:

```
cd hypernets_tools/
git branch -r             # List all remote branches
git checkout main         # Checkout the branch you want
```

You should now be able to use the installation wizard:

```
sudo ./install/EE_wizard.sh
```

This wizard will prompt you a menu where you can:

1. Update hypernets_tools
2. Install Dependencies
3. Download and install YoctoHub
4. Run Yocto-Pictor auto-config
5. install / update libhypstar
6. Configure Hypstar port

Once every step is done, you can try to launch the *Graphical User Interface* (and refer to the section 5 for the first tests) with:

```
python -m hypernets.gui
```

4.3 Configuration Files

All parameters of the *hypernets_tools* can be found in text files called *configuration files*. Splitted in two different parts, the first one is called *static* (see annex 7.1) ; and the second one *dynamic* that will be synchronized over network allowing the user to perform remote setting-up of the system (see annex 7.2). You can edit it using any text editor (e.g. vi, nano, ...).

```
ls
config_static.ini  config_dynamic.ini
```

4.4 Common Examples of Command

As a python application, you can *call* most of the modules and submodules of *hypernets_tools* using the *python -m hypernets.module.submodule* syntax. Moreover, the command "–help" with a package name can provide informations about how to use the package on the *Command Line Interface*. Note also that arguments are following the GNU argeument syntax convention :
https://www.gnu.org/software/libc/manual/html_node/Argument-Syntax.html

4.4.1 Driving the Yocto-Pictor

```
python -m hypernets.yocto.relay --help
```

logs/yocto_relay_help.txt

```
usage: relay.py [-h] (-g | -s {on, off} | -r | -p {on, off, unchanged})
                -n {1..6} [-f]

optional arguments:
  -h, --help            show this help message and exit
  -g, --get             display relay's states
  -s {on, off}, --set {on, off}
                        set the state of the relay
  -r, --reset           reset relay (1 sec off, then on)
  -p {on, off, unchanged}, --set-at-power-on {on, off, unchanged}
                        schedule the state of the relay for next wakeup
                        (use --force [-f] to write in flash memory)
  -n {1..6}, --id-relay {1..6}
                        ID number of the relay
  -f, --force           forces relay #1 to switch off, and allows to
                        write in memory for the state at power-on option
```

Closing the relay number one:

```
python -m hypernets.yocto.relay --state on --id-relay 1  # long version
python -m hypernets.yocto.relay -son -n1                 # short version
```

4.4.2 Moving the Pan-Tilt

```
python -m hypernets.geometry.pan_tilt --help
```

logs/pan_tilt_help.txt

```
usage: pan_tilt.py [-h] -p {0..360} -t {0..360} [-w] [-v]

optional arguments:
  -h, --help            show this help message and exit
  -p {0..360}, --pan {0..360}
                        set pan (azimuth angle in degrees)
  -t {0..360}, --tilt {0..360}
                        set tilt (zenith angle in degrees)
  -w, --wait            wait for pan/tilt end of move and return real
                        position
  -v, --verbose         print extra information
```

```
python -m hypernets.geometry.pan_tilt --pan 90 --tilt 180 --wait # long
python -m hypernets.geometry.pan_tilt -w -p 90 -t 180            # short
```

4.5 Protocol Files

In order to process an automatic sequence file (i.e. a multitude of measurement from diverse geometry), you have to first define a *sequence file*. A bunch of example of sequence file are given in the hypernets/ressource/sequence_sample folder. Edit then the field *sequence_file* under the general section of *config_dynamic.ini*. Note that the path to your sequence file can be absolute or relative path (see Annex 7.2). Prior to any other instruction, the file has to start with this line:

```
HypernetsProtocol v2.0
```

The general syntax to make a measurement can be given as a *one line instruction*:

```
@[geometry1, #flag1, #flag2] + I.measurement1 + J.measurement2
```

or like this:

```
@[geometry2]
      + K.measurement3
      + L.measurement4
```

Whith *geometry1* and *geometry2* for two geometries; *I, J, K, L* are the counts (integers, number of measurement) associated respectively with *measurement1, measurement2, measurement3 and measurement4*; and *flag1* and *flag2* are two associated conditions for the geometry1.

4.5.1 Geometry

A geometry always starts with the symbol "@" and contains a *pan* (azimuth) coordinate follow by a pan reference, and a *tilt* (zenith) coordinate follow by a tilt reference. Optionally, one or several flags (conditionning flag) can be specified in a geometry.

3 types of geometry reference are understandable :

- *abs* : absolute (given as the pan-tilt reference)
- *sun* : sun azimuth and zenith as a reference
- *hyper*: north (pan) and nadir (tilt) as a reference

4.5.2 Measurements

Measurement can be of 3 different types:

- Radiometer: defined as a type (*vnir, swir, both)* and an entrance (*rad, irr, dark*), follow by two integration times concatenated by dots.
- Picture
- Validation

4.5.3 Examples

```
@[0, sun, 0, sun] + 1.picture           # Take a picture of the sun
@[0, hyper, 90, hyper]                  # Point to the horizon north
@[198, abs, hyper] + 20.validation      # Make 20 VM measurements
@[90, sun, 180, hyper] + 6.vnir.irr.0.0 # Make 6 irradiance measure
```

```
HypernetsProtocol v2.0

# Flag definitions :
~ #firstLessThanFourSec := $spectra_file1.it_vnir <= 4096
~ #sequenceWasFastEnough := $elapsed_time < 120

@[ 90.0, sun, 180.0, hyper ]
    + 1.vnir.irr.0.0
# Picture of the Sun only if sequence is < 2 minutes
@[ 0.0, sun, 0.0, sun, #firstLessThanFourSec ]
    + 1.picture
# Park only if first integration time is less than 4 second :
@[ 0.0, sun, 0.0, hyper, #sequenceWasFastEnough ]
```

5 First tests

Required time: 15 min

This section describes how to operate the functionalities of the Yocto-Pictor including relays functions, GPS, temperature, humidity, or light sensors values reading. The second part explains how to use the Yocto-Pictor to power the computer instead of a *classical* power supply unit. Finally, instrument pointing and Hypstar measurements thanks to the GUI will be described.

5.1 Prior Set-up of the System and First tests

If necessary, get the *hotspot connection* working back (8.2.1) and start the GUI:

```
python -m hypernets.gui
```

Follow those steps to take a spectrum:

- Click on *Connection* and activate the relay #2 (figure 23)
- Wait for 20 seconds (instrument boot time) and click on *acquisition*
- Click on *Show plot* to see the taken spectrum (figure 25)

Note: in order to display the real axis unit (in nanometre) and to calibrate the spectrum. First *dump* the *linearity and calibration coefficients*, from the radiometer. Let the relay #2 opened, exit the GUI and type once command:

```
python -m hypernets.dump_current_config
```

5.2 Validation Module Testing

- Open the GUI and activate relay #2, and #3 (Figure 23).
- Wait for 20 seconds (instrument boot time)
- On the pan-tilt section: set-up the tilt to 198°, click on *move*
- Press on *Turn VM on*, and *Measure VM*
- Visually check if the Validation Module is lighting
- The spectrum graph of the measure should appears (Figure 25)

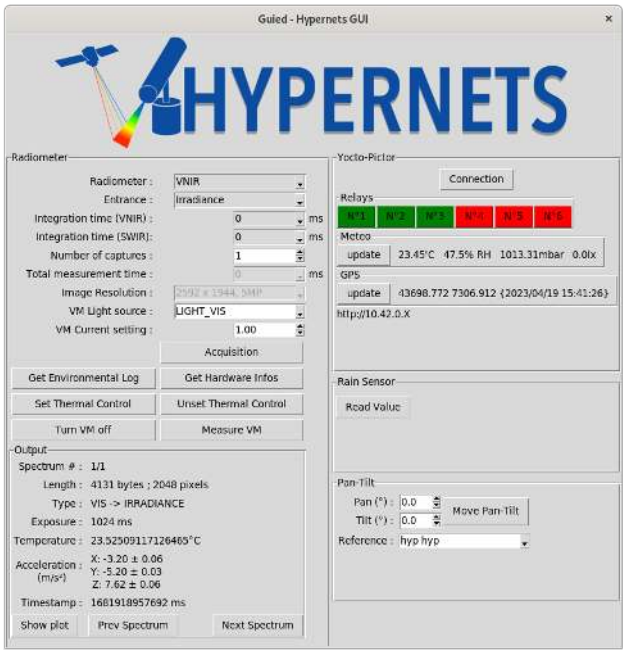

Figure 23: Graphical User Interface

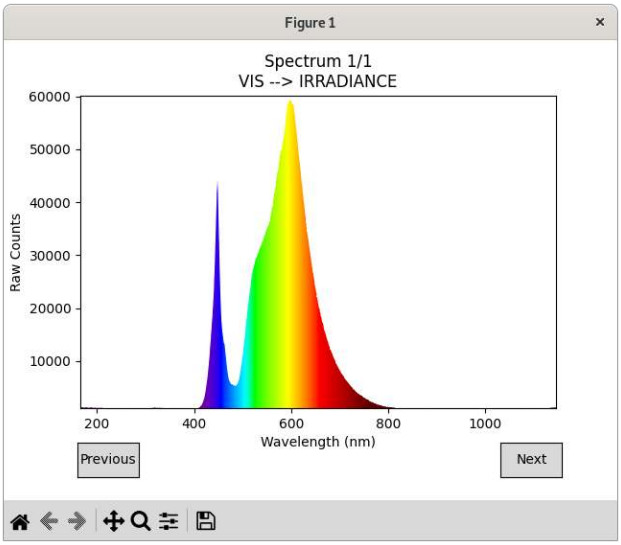

Figure 24: A Spectrum Plot

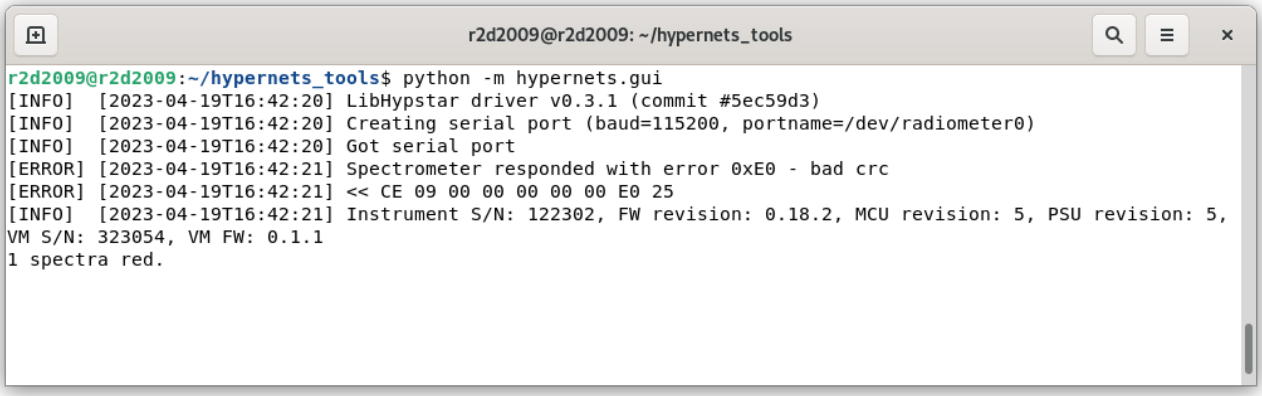

Figure 25: Some Outputs on the console

6 Autonomous System Mode

6.1 Set Relay n°1 ON at Wake-up

Before using the studiel-box, we need to set the Yocto-Pictor first relay default value as ON. This aims to power-on the rugged-PC at the wake-up time of the Yocto-Pictor. To achieve this, fire up a terminal and use those commands:

```
cd hypernets_tools/
python -m hypernets.yocto.relay -fpon -n1
```

In order to test if it worked, hold the sleep button (the middle one) on the Yocto-Pictor until it goes to sleep. Then wake it up by pressing the wake-up button (the upper one on the figure 26). The relay 1 should be closed and the associated green LED (i.e. the LED 1 ; on the right of the card in the figure 26) should light.

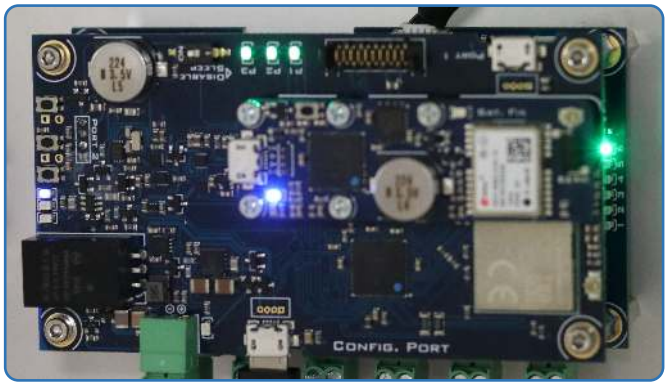

Figure 26: Yocto-Pictor: Three black buttons on the upper-left corner & Green Relay Light on the right side of the bottom card

6.1.1 Enable SSH access

Closing the Studiel-box will make our system *headless*. *i.e.* it won't have any keyboard, screen or mouse connected on it. As we still need to interact with it, a workaround solution is to connect to Wi-Fi and access it through a SSH, *a Secure Shell*. To enable this service, type those two lines in terminal:

```
sudo systemctl enable sshd    # For Manjaro
sudo systemctl start sshd
sudo systemctl enable ssh    # For Debian
sudo systemctl start ssh
```

Expected output:

```
Created symlink /etc/systemd/system/multi-user.target.wants [...]
```

Then, try to connect to the Wi-Fi with any laptop and SSH like this (you can use Putty on Windows):

```
ssh 'rugged_pc_user_name'@10.42.0.1
```

6.2 Try a full sequence acquisition

Now edit the *dynamic configuration* file and edit it with the following values:

- sequence_file = *path to the sequence of your choice*
- start_sequence = yes
- keep_pc = on

Then, run the command below and see if it works. The output *sequence directory* should be saved in the folder *DATA*. Its name is prefixed by *SEQ* and given according to the *current UTC date and time* with the following format: *SEQYYYYMMDDTHHMMSS*.

```
./utils/run_service.sh
```

If everything is working properly, you can refer to the yoctopuce wake-up scheduler documentation in order to configure the system to wake up autonomously. In addition, to enable the automatic run of the sequence every time the system wakes up, run the installation script:

```
sudo install/04_setup_script_at_boot.sh
```

6.3 Setup Server Communication

If not already done, generate a ssh key with:

```
ssh -t rsa
```

Edit the section *network* of the file *config_static.ini*, create an empty directory in the appropriate folder on the server and run:

```
sudo install/05_setup_server_communication.sh
```

Example: if your site name is

7 Annexes

7.1 Configuration file: static_config.ini

The *static configuration file* is the file dealing with the parameters that user will preferably not change. It includes the communication settings with the Yocto-Pictor and the server credentials. As it contains critical settings, this file will not be automaticaly synchronized over the network.

7.1.1 Section: yoctopuce

field	**definition**	values (**default**)
yoctopuce_ip	Yocto-Pictor-Wifi IP Address	10.42.0.X
yocto_prefix1	Yocto-Pictor Serial (relay board)	OBSVLFR1-XXXXXX
yocto_prefix2	Yocto-Pictor-Wifi Serial	OBSVLFR2-XXXXXX
yocto_gps	Yocto-GPS-V2 Serial	YGNSSMK2-XXXXXX
bypass_yocto	Bypass Yocto-Pictor during in a sequence (debug mode)	**no**

7.1.2 Section: network

field	**definition**	**default**
credentials	user account and server name	user@server
remote_dir	name of the remote directory where data/config/logs are pushed	~/XXYY
ssh_port	ssh port used to connect to the server	22
remote_ssh_port	ssh port used to connect from the server to the host system (use ssh -p 20213 rugged_pc_user@127.0.0.1 to connect)	20213

7.2 Configuration file : dynamic_config.ini

The *dynamic configuration file* is a communication tool allowing the user to remotely change settings of the system. Unlike the *static configuration file*, it is synchronized with an identical file on the server side. Synchronization over the network is performed using the one with the more recent modification.

7.2.1 Section: general

field	**definition**	values (**default**)
keep_pc	state of the system at the end of the sequence	on / **off**
start_sequence	start or no the sequence every time the system is turned on	**yes** / no
auto_update	check for update at start-up	yes / **no**
sequence_file	which protocol file to follow	*path to file*

7.2.2 Section: GPS

field	**definition**
latitude	Cartesian latitude of the location (float)
longitude	Cartesian longitude of the location (float)

7.2.3 Section: pantilt

field	**definition**	values (**default**)
offset_pan	Azimuth adjustement value (in degrees) to point to South or North	[0 - 360] **0**
offset_tilt	Zenith adjustement value (in degrees) to point to point the Nadir	[0 - 360] **60**
reverse_tilt	Change the sign of both Azimuth and Zenith	yes / **no**

7.2.4 Section: metadata

Metadatas are the information embedded in each Hypernets Sequence folder. In particular, each metadata file will have *a header*, that is, the top the file. This header contains some dynamic informations like the current date and time as well as static ones, for instance the site identification. This section of the *dynamic config file* is the exact content of each metadata header. Fields between brackets will be automatically processed by the host system.

field	**definition**	values
principal_invesigator	Name of the PI,	*Investigator name*
datetime	UTC date and time at the sequence start time (no need to edit)	{datetime}
hypstar_sn	Hypstar Serial Number	123456
site_id	A site ID with the naming convention: XXYY such as: – XX: two first letters of the site name – YY two first letters of the country partners-name	XXYY
protocol_file_name	Self-reference to the *protocol file name* from the *general section* of this file (no need to edit)	${general:sequence_file}
latitude	Self-reference to the latitude from the GPS section of this file (no need to edit)	${GPS:latitude}
longitude	Self-reference to the longitude from the GPS section of this file (no need to edit)	${GPS:longitude}
offset_pan	Self-reference to the *protocol file name* from the *pantilt* of this file (no need to edit)	${pantilt:offset_pan}
offset_tilt	Self-reference to the *protocol file name* from the *pantilt* of this file (no need to edit)	${pantilt:offset_tilt}
offset_tilt	Self-reference to the *protocol file name* from the *pantilt* of this file (no need to edit)	${pantilt:offset_tilt}
azimuth_switch	Self-reference to the *protocol file name* from the *pantilt* of this file (no need to edit)	${pantilt:azimuth_switch}

Note : if you want to see what your metadata header will look like, you can use this command :

```
python -m hypernets.abstract.create_metadata
```

7.2.5 Section: hypstar

field	**definition**	values (**default**)
hypstar_port	path to the hypstar *device file*	**/dev/radiometer0** /dev/ttyUSB5
baudrate	speed communication (in bauds) with the instrument	**115200**, 460800, 921600, 3000000, 6000000, 8000000
loglevel	Set the log verbosity of the communication with the instrument	**ERROR**, INFO, DEBUG, TRACE
swir_tec	Thermoelectric Cooler setting point (in degree Celsius) the SWIR module	[-15 ; +40] **0**

Log Level	**Definition**
ERROR	only errors are reported on stderr
INFO	stdout + stderr
DEBUG	driver command execution printout to stdout
TRACE	low level communication bytes are printed to stdout

7.3 How To Access to the BIOS of the Cincoze

You will need to access to the *Basic Input Ouput System* menu (or BIOS CMOS Setup Utility) to perform some configuration tasks like set-up the serial port or boot from USB stick. To get this acces, turn *OFF* and *ON* the computer, then immediately press **<del>** or **<ESC>**. You should see a screen similar to the figure 27. If not, you may refer to the User's Manual of the computer.

Figure 27: Screenshot of BIOS

7.3.1 Pan-Tilt: Serial Port Configuration

Go to the BIOS (see annex 7.3). Under the tab *Advanced* go to the menu *Super IO Configuration*, select the serial port for the pan-tilt (usually the n°4) and select *RS485 Half Duplex* on the line *Onboard Serial Port 4 Mode*. Save and exit.

7.4 Yoctopuce Tips

7.4.1 Upgrade Yocto-Pictor firmwares

Please refer to the yoctopuce documentation in order to upgrade the firmwares.

7.4.2 How to catch debug informations

The Yocto-Pictor is still a tool under development and some bugs may sometimes show up. So when something wrong happens. For example the software is reporting an error relative to yocto or the behaviour is not the one expected. It could be helpful to join a *memory dump* of the card to the issue report. In order to get this file, here is the most common scenario:

1. Something goes wrong with the card.
2. Don't touch anything but hold on the Ybutton during 5 seconds (see figure 28).
3. Now, restart the system in order to get back the Wi-Fi connection working.
4. Go to the yoctopuce webpage (by typing : 10.42.0.X:4444 in the address bar of a web browser).
5. Click successively on *Configure* (on the row of *Yocto-Pictor-Wifi*), then *manage files* (see 29).
6. You should see a row with *dump.bin* under the column *Path/name*. Download this file, using *right click* and *Download Linked File*. (see 30).
7. Finally click on *remove* to delete the file in order to avoid eventual future confusion.

For convenience, you can name the file with your initials, the date and an error reference if there is one. Also, please mind to always send this file with an explanation of the issue or the error logs.

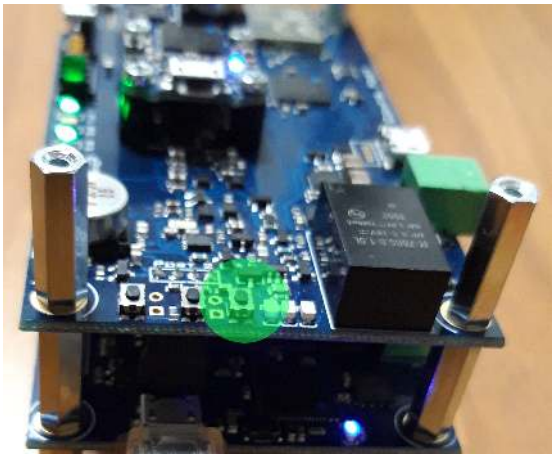

Figure 28: Highlighted *YButton*

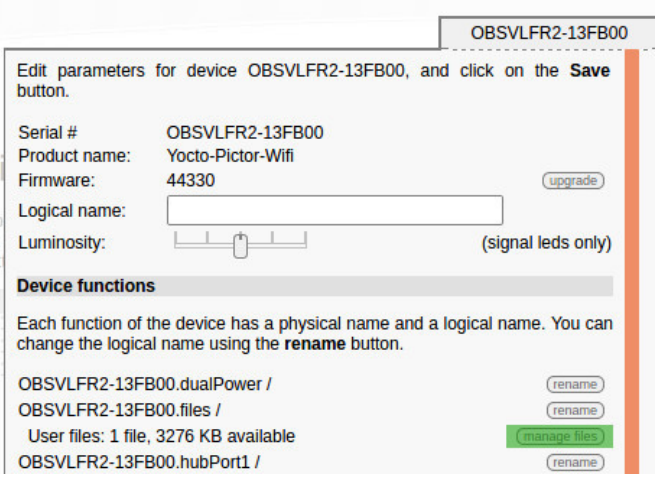

Figure 29: Hightlighted *file manager*

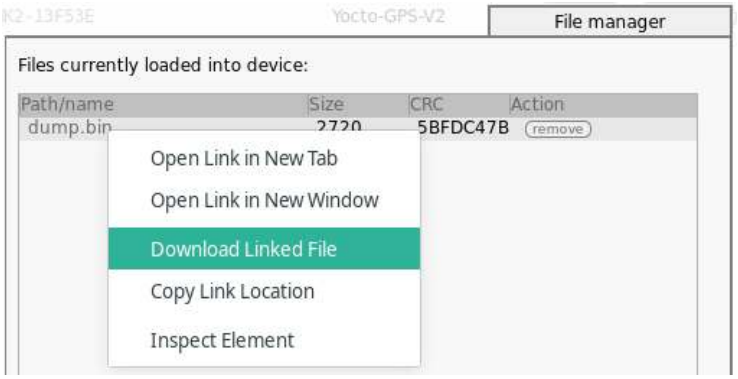

Figure 30: Download *dump.bin*

8 Old Versions Documentation

8.1 Manjaro Linux System

8.1.1 Creating A Live USB Of Manjaro Linux

First, please go to manjaro archive download page and download the "minimal" version ISO of Manjaro linux 21.0. Then, make the ISO file Bootable on USB drive. Under Windows, you can do it by using easy2boot for example. Under Linux and Mac, you can use the following instructions.

Open a terminal, type the following command:

```
sudo dmesg -w
```

Now connect your USB stick memory, you should get output similar to this:

```
 1 [10576.130389] usb-storage 2-6:1.0: USB Mass Storage device detected
 2 [10576.130756] scsi host5: usb-storage 2-6:1.0
 3 [10577.144722] scsi 5:0:0:0: Direct-Access     SanDisk  Ultra USB 3.0
 4 1.00 PQ: 0 ANSI: 6
 5 [10577.145229] sd 5:0:0:0: Attached scsi generic sg6 type 0
 6 [10577.145493] sd 5:0:0:0: [sdf] 30031872 512-byte logical blocks:
 7 (15.4 GB/14.3 GiB)
 8 [10577.146230] sd 5:0:0:0: [sdf] Write Protect is off
 9 [10577.146233] sd 5:0:0:0: [sdf] Mode Sense: 43 00 00 00
10 [10577.146485] sd 5:0:0:0: [sdf] Write cache: disabled, read cache:
11 enabled, doesn't support DPO or FUA
12 [10577.153805]  sdf: sdf1
13 [10577.155586] sd 5:0:0:0: [sdf] Attached SCSI removable disk
```

Here, the linux device naming for the USB key is */dev/sdf* (line 12). Next step is to *unmount* all file systems of this USB key from the file hierarchy in order to prevent any access by other application. Then we need to format the unmounted drive in Linux jounaling file system *ext4*. And finally copy the Manjaro Linux ISO file to the USB drive. Adapt the following commands with the name of your device and the path to the ISO file you just downloaded. Note the ending star for the first command.

Warning: it is critically important not to copy the ISO onto the wrong device! Please double check before going ahead with the next.

```
umount /dev/sdf*
sudo dd if=/path/to/manjaro.iso of=/dev/sdf status="progress"
```

If ISO is properly copied to bootable USB, you should see the new name MJRO1815 for the USB key.

8.1.2 Install and Update Manjaro Linux

Insert the Live USB flash drive into the computer, power it, and get access to the BIOS (see annex 7.3). Go under the tab named *Boot* and press enter on the menu *Hard Drive BBS Priorities*. You should see the name of your hard-drive on the first row and the name of your usb drive on the second one. If you don't see your USB key in here this may mean your bootable USB key is not made properly. Use <+/-> controls to put your flash drive on the first, then press <**Esc**> to quit the menu, go under *Save & Exit* and select *Save Changes and Reset*.

Once Manjaro has started, double-click on the icon *Install Manjaro Linux* (lower left corner)

- Use American English (US)
- Choose Europe as Region and London as Zone
- Choose ERASE DISK
- Login and password (use same for administrator)

Then reboot again without the USB stick. After getting your desktop environment started, set up an Internet connection, open a terminal (press <**ctrl + alt + t** >) and type the following commands in the terminal in order to clean your *home directory* update the system and install *git*:

```
rmdir Music/ Pictures/ Videos/ Public/ Templates/
sudo pacman -Syu
sudo pacman -S git
sudo pacman -S git
```

Say yes/or default [ENTER] to all questions. Note that once done, the computer may reboot.

8.2 Wi-Fi Connection Between the Yocto-Pictor and the rugged PC

Required time: 10 min

The Yocto-Pictor, provided as the electronic card of Hypernets system is controlled through Wi-Fi by the rugged PC. The three following subsections describe how to set-up this communication.

8.2.1 Wi-Fi Hotspot Set-up

This part aims to configure the Wi-Fi of the rugged PC. It will mainly be used to control the Yocto-Pictor. But will also be useful to access the rugged PC after having installed it in the main box of the system.

- First, right click on the the connection logo in the bottom right corner of the screen and click *Edit Connections...* (see figure 31).
- Click on + to add a new connection and select *Wi-Fi* for the Connection Type.

  On the tab *General*:

  - Check *Connect automatically with priority 1*
  - Check *All users may connect to this network*

  On the tab *Wi-Fi*:

  - Set the mode to *Hotspot*
  - Choose a *SSID*: the name that will appear as Wi-Fi network.

  On the tab *Wi-Fi Security* (see figure 32).

  - Set the security to *WPA & WPA2 Personal*,
  - Choose a valid password
- Choose a connection name (on the very top)
- Save
- Close the current window and finally left click on the connection logo, *Create New Wi-Fi Network* (see figure 33) and select your connection in the drop-down list to enable it.

Note: further, you may want to use Wi-Fi connection to get an internet acces. Keep in mind that you can switch the Wi-Fi connection and get back to the previous one by following this last step.

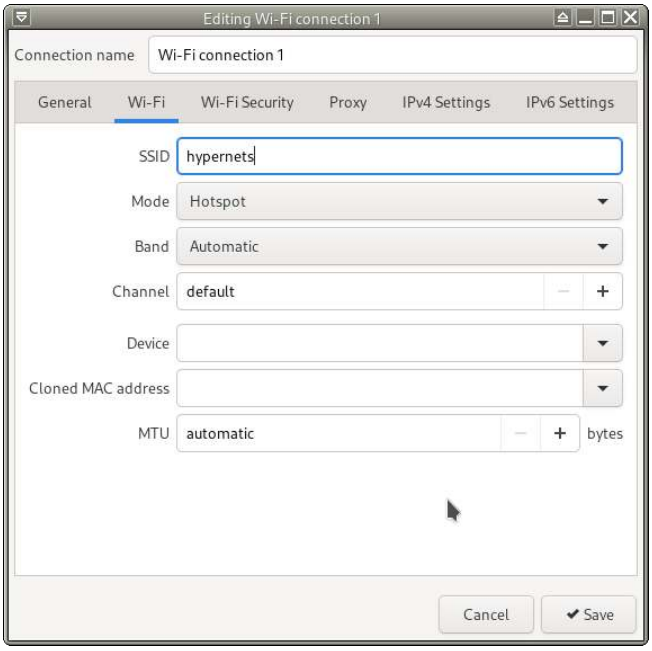

Figure 31: Edit Connections

Figure 32: Wi-Fi Connections Access

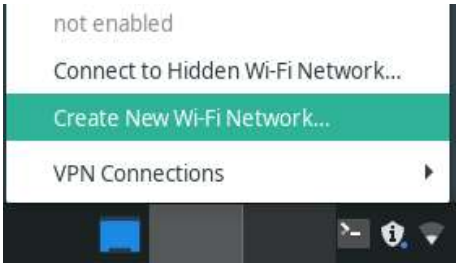

Figure 33: Retrieve Wi-Fi Hotspot

8.2.2 Setting up the Yoctopuce VirtualHub

According to the Yoctopuce documentation:

> The VirtualHub is a toolbox for Yoctopuce USB devices. It will allow you to:
>
> - configure and test Yoctopuce devices
> - remotely control Yoctopuce devices through network

Here, you should have an updated linux distribution on your rugged PC and a working internet connection wired on it. Open a terminal, clone the installation script repository on your home folder, and run the script with *sudo priveleges*:

```
<ctrl + alt + t>  # open a terminal
cd
git clone https://github.com/hypernets/hypernets_tools
git checkout beta
cd hypernets_tools/
sudo ./install/00_install_yoctohub.sh
```

You should now be able to run the VirtualHub by typing:

```
/usr/sbin/VirtualHub
```

8.2.3 Yocto-Pictor Wi-Fi Connection

Run the VirtualHub (see last command in the previous section), open the web browser and visit the following address:

```
127.0.0.1:4444
```

If you don't see a screen like figure 35, please retry instructions from the above section 36.

If you were using Wi-Fi to get Internet, enable the hotspot (see last note from section 35)

Plug the Wi-Fi antenna (see figure 34) and connect the Yocto-Pictor-Wifi with your usb cable on the config port (see figure 36). Click on the button *configure* and edit the *WLAN settings* in the section *Network Configuration*:

1. Select *Infrastructure (choose an existing WLAN).*
2. Enter the password you choose in section 35.
3. Click on *Test Config* and note the displayed Current IP.
4. Save and exit.

Now, you can come back to your terminal with **<ctrl + alt + t >** and quit the VirtualHub **<ctrl + c >**.

Type the IP address followed by *":4444"* in the address bar of the web browser and you should see a screen similar to the one before with 3 lines and serials (see figure 37). Note the 3 serials of these devices and the IP address. It will be required later (in the section 38).

Note: at this point you should consider to update the firmwares (see the annex 7.4.1)

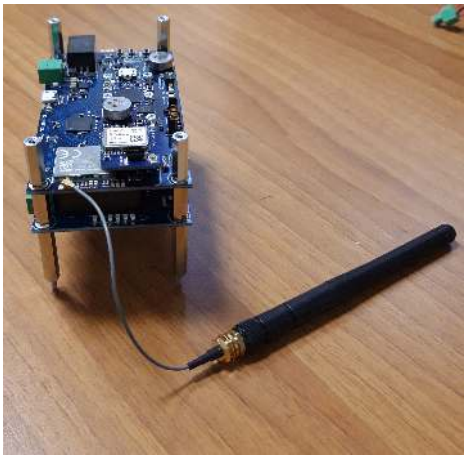

Figure 34: Wi-Fi Antenna

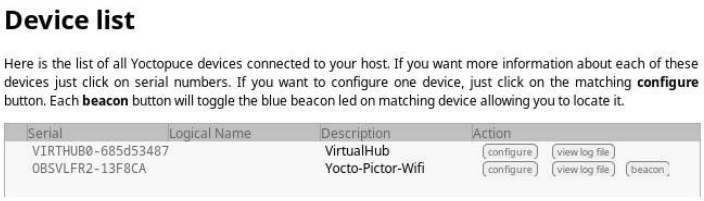

Figure 35: Web Interface: Virtual Hub

Figure 36: Yocto-Pictor: top floor USB connection

Figure 37: Web Interface: serial

8.3 Linking Hypernets Tools to the Yocto-Pictor

If not already done, first copy the two configuration file templates in the hypernets_tools folder:

```
cd hypernets_tools/
cp hypernets/resources/config_dynamic.ini.template config_dynamic.ini
cp hypernets/resources/config_static.ini.template config_static.ini
```

Connect the PC to the internet, go to the folder *hypernets_tools* and install required dependencies:

```
cd hypernets_tools/
sudo ./install/01_dependencies.sh
```

Now copy the template of the static configuration file (see description in annex 7.1) and edit the Yoctopuce section according to the serial that you previously noted in the section 37:

```
cp hypernets/resources/static_config.ini.template static_config.ini
mousepad static_config.ini
```

If necessary, mind to switch back the Wi-Fi connection to the hotspot mode (8.2.1) and try to run the Graphical User Interface (GUI) with:

```
python -m hypernets.gui.frame_yoctopuce
```

You should see some meteo and GPS data after clicking on *Connection*. Also you can try playing with relays to check if everything is working (see figure 38).

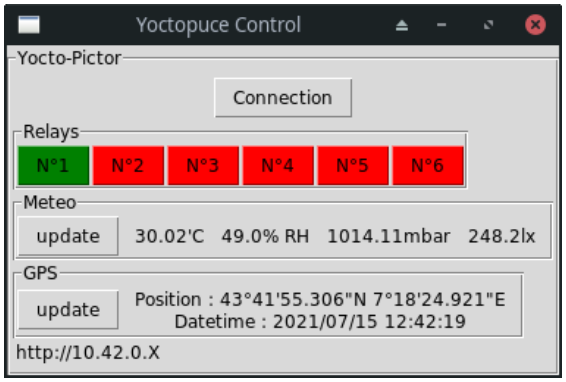

Figure 38: GUI Yoctopuce Control

8.3.1 Radiometer Configuration and First tests with the GUI

- Yocto-Pictor
- USB 2.0 cable of type A-micro B (data + power)

Open a terminal and type the following command in order to link the radiometer to a *device file*:

```
sudo ./install/02_configure_ports.sh
```

This should output with something like:

```
lrwxrwxrwx 1 root root 7 Jul 15 15:04 /dev/radiometer0 -> ttyUSB5
```

Type the following command (required internet connection) to update the *libhypstar driver*:

```
sudo ./install/03_update_libhypstar.sh
```