**Programme:**

**Project/WP:** Electronic Engineering Department

CRISLER Monitoring USER MANUAL

**Document Number:**

**Document Version:** 0.3

**Document Type:**

**Released On:**

**Document Classification:**

|  |  |
| --- | --- |
| **Prepared by:** | Benoit Serra |
| **Validated by:** |  |
| **Approved by:** |  |
|  | Name |

Authors

|  |  |
| --- | --- |
| **Name** | **Affiliation** |
| Benoit Serra | ESO |
|  |  |
|  |  |
|  |  |
|  |  |

Change Record from previous Version

|  |  |
| --- | --- |
| **Affected Section(s)** | **Changes / Reason / Remarks** |
| All | Initial drafting |
| 5 Data Visu. | Adding section for visualisation of data logs from CRIMON |
| All | Adding the instrumentation config file |
| All | Adapted to version 1.0 of the monitoring |
|  |  |

Contents

[1. Introduction 4](#_Toc517793668)

[1.1 Scope 4](#_Toc517793669)

[1.2 Definitions, Acronyms and Abbreviations 4](#_Toc517793670)

[2. Related Documents 4](#_Toc517793671)

[2.1 Applicable Documents 4](#_Toc517793672)

[2.2 Reference Documents 5](#_Toc517793673)

[3. CRIMON description 6](#_Toc517793674)

[3.1 Development 6](#_Toc517793675)

[3.2 Requirements 6](#_Toc517793676)

[3.3 Description 7](#_Toc517793677)

[4. Operation 9](#_Toc517793678)

[4.1 Starting the monitoring 9](#_Toc517793679)

[Below is a step by step description of how to start: 9](#_Toc517793680)

[4.2 Interact with the instruments 11](#_Toc517793681)

[4.2.1 Using the GUI 11](#_Toc517793682)

[4.2.2 Using a Python script 12](#_Toc517793683)

[5. Data visualisation 14](#_Toc517793684)

[5.1 Step by step 14](#_Toc517793685)

[5.2 Notes 15](#_Toc517793686)

# Introduction

## Scope

This user manual covers the setup and operation of the monitoring tool CRIMON used for the CRISLER cryostat. CRISLER is a NIR detector test cryostat located in lab E.0.38 at ESO Garching. The cryostat is intended for use by expert technical staff in the detector systems group. This manual is to aid in understanding the monitoring tool as well as how to start it and how to interact with it while in operation.

## Definitions, Acronyms and Abbreviations

This document employs several abbreviations and acronyms to refer concisely to an item, after it has been introduced. The following list is aimed to help the reader in recalling the extended meaning of each short expression:

|  |  |
| --- | --- |
| TBC | To Be Clarified |
| TBD | To Be Defined |
| CRIMON | CRIsler MONitoring tool |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Related Documents

## Applicable Documents

The following documents, of the exact version shown, form part of this document to the extent specified herein. In the event of conflict between the documents referenced herein and the content of this document, *the* content of this document shall be considered as superseding.

AD references shall be specific about which part of the target document is the subject of the reference.

1. [Lakeshore Model 336 User Manual](https://www.lakeshore.com/Documents/336_Manual.pdf)

1. [DP800 Series Programmable Linear DC Power Supply](https://www.batronix.com/pdf/Rigol/ProgrammingGuide/DP800_ProgrammingGuide_EN.pdf)
2. [R&S HMC8012 Digital Multimeter SCPI Programmers Manual](https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_common_library/dl_manuals/gb_1/h/hmc8012_1/HMC8012_SCPI_ProgrammersManual_en_01.pdf)
3. [Agilent InfiniiVision 5000 Series Oscilloscopes Programmer’s Guide](https://www.keysight.com/upload/cmc_upload/All/5000_series_prog_guide.pdf)
4. [Log file example 26-06-2018\_13-53-09.xlsx](file:///\\dsraid2\raid\projects\CRISLER\Manuals\UserManual\26-06-2018_13-53-09.xlsx)
5. [InstructionServer\_Client.py](file:///\\dsraid2\raid\projects\CRISLER\Software\Monitoring\test_scripts\InstructionServer_Client.py)
6. [ResultsCooldown.py](file:///\\dsraid2\raid\projects\CRISLER\Software\Monitoring\test_scripts\ResultsCooldown.py)

## Reference Documents

The following documents, of the exact version shown herein, are listed as background references only. They are not to be construed as a binding complement to the present document.

1. <Document name>;

<Document number Version X>

1. <Document name>;

<Document number Version X>

1. <Document name>;

<Document number Version X>

# CRIMON description

This section describe the software CRIMON, the packages requires and how to set it up.

## Development

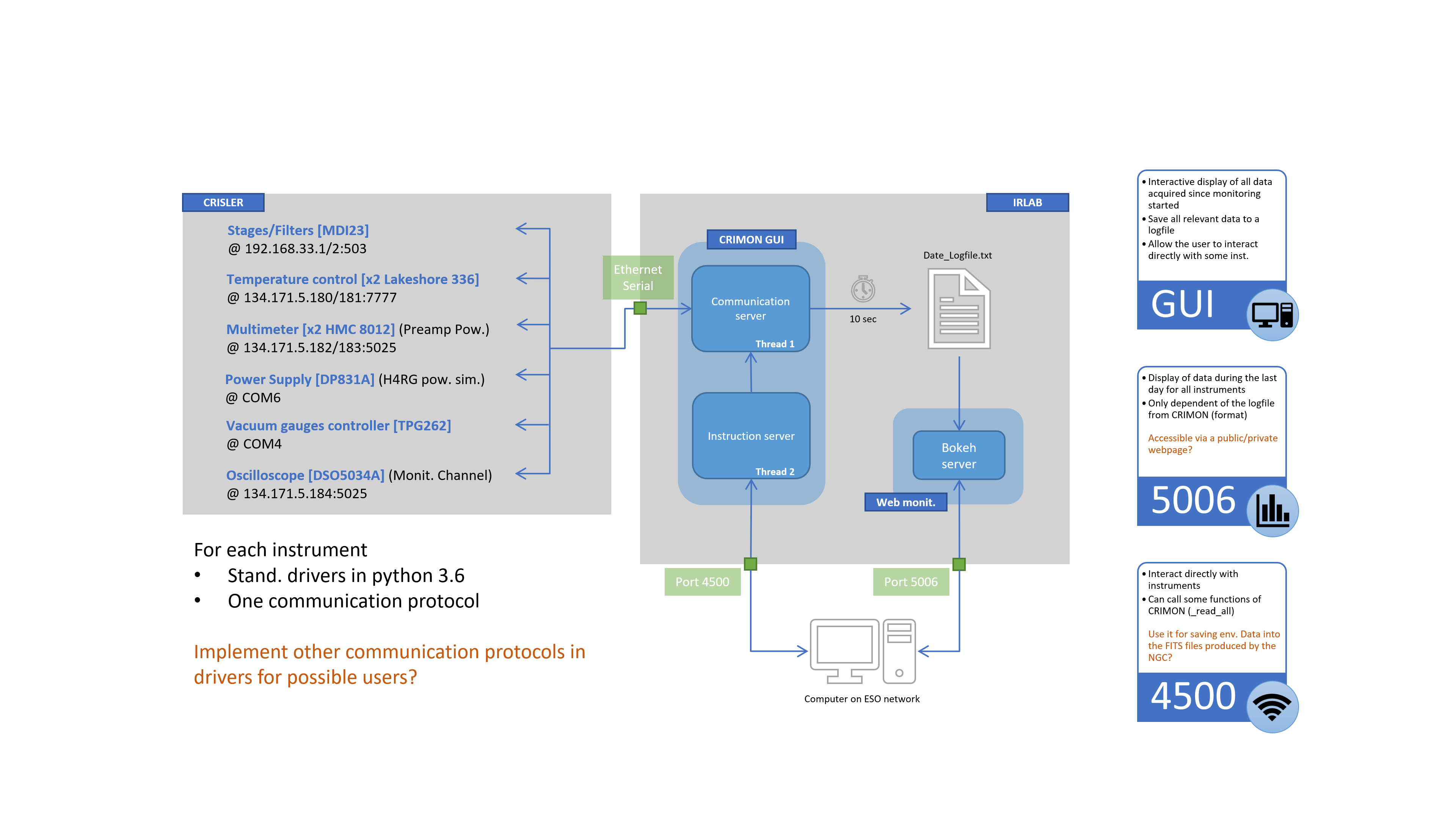
CRIMON is developed with Python 3.6 and is currently in version 0.4.4, below is a description of the version changes that the software went through.

|  |  |
| --- | --- |
| Version | Comments |
| 0.1 | Embedding pyqtgraph in the interface, communication tab with each  instrument |
| 0.2 | Changing layout to add a splitter in order to resize the graph/commands |
| 0.3 | Add support for second lakeshore controller  Added a configuration file (instrumentation\_config.json)  Added dynamical pen, linestyle change with type of instr., color change for different sensors |
| 0.3.1 | Pressure Y axis now in log scale  Line width set to 3  Adding a more appropriate message to the About info box  Status bar now display name+version of soft  X grid displayed on plots init |
| 0.4 | Adding UPS communication |
| 0.4.1 | Adding Multimeter rhode&schwarze communication (HMC8012)  Adding Power Supply rhode&schwarze communication (HMP4040) |
| 0.4.2 | Adding switching mechanism for choosing which instrument we want to use |
| 0.4.3 | Adding a last cooldown directory creation in the file menu for support of the bokeh server  Adding an instruction server which redirect request to instruments on a separate thread [class InstructionServer():] host='134.171.36.18', port=4500  Adding oscilloscope Agilent DSO5034A host='134.171.5.184', port=5025  Adding power supply Rigol DP831A (replacing R&S HMP4040) serial COMx? |
| 0.4.4 | Instruction server functions:  INST|READ: dict. of all instruments as define in \_dump\_sensors  INST|LIST: dict. of all instrument tags <TAG> and their resp. class  INST|<TAG>|<CMD>: raw output of the <CMD> for inst <TAG> |
| 0.4.5 | Configuration file instrumentation\_config\_2 json file with proper structure  Communication will loop over every entry and configure the communication to the instrument  Adding timestamp to the dictionary of the datapoint, switching reading # and time in log files  Only ONE real time process at a time. Either the monitoring from CRIMON and the only data available would be the last point  Or the instruction server is running and monitoring is not and full real time is available  On a longer time scale, when acquiring data with less than a second DIT will not give enough time to the instruction server to request a whole new set of data. With this solution the response time is much faster |

## Requirements

CRIMON requires a 3.6 Python version as well as the following packages [TBD]

## Description



CRIMON functions can be summarized in the following list:

1. Record the data for each instrument that is connected via Serial or Ethernet connexion (with a specific driver written in Python).
2. Display the data recorded in a responsive GUI
3. Centralize the communication for all instruments in a simple interface in the GUI
4. External communications with the instruments can also be done by the use of an instruction server with specific commands. Thus allowing the user to ‘script’ a full experiment.
5. Display the data externally using a bokeh server (separate).

# Configuration

The instrument configuration can be changed in the instrumentation\_config\_2.json file. For each type of instrument (temperature, pressure, power supply, multimeter…) two fields are mandatory:

* Ident: which will determine which driver it will use [TBD a summary of all instrument drivers]
* Comm: the communication process and the address (serial+com or TCP+IP address)

Other optional fields will be described for each category.

## Temperature

For the temperatures, only Lakeshores 336 are used.

"T": {

"ident":["Lakeshore336",

"Lakeshore336"],

"Comm": [

["TCP", "134.171.5.180"],

["TCP", "134.171.5.181"]

],

"LSCI,MODEL336,336AANG/#######,2.2": ["A", "B", "C"],

"LSCI,MODEL336,LSA163F/LSA15SI,2.7": ["B", "C", "D", "D2", "D3", "D4", "D5"],

"LSCI,MODEL336,LSA22LU/LSA217G,2.8": ["A", "B", "C"]

},

The optional fields are the IDN of the Lakeshores with the channels that we want to record.

## Multimeters

For the voltage and current, mutlmeters HMC8012 are used

"MM": {

"ident":["HMC8012",

"HMC8012"],

"Comm": [

["TCP", "134.171.5.182"],

["TCP", "134.171.5.183"]

],

"Opts": [

["DC\_I"],

["DC\_V"]

]

},

The optional field is Opts, which will determine in which mode the multimeter will be used (AC/DC I or V).

## Power supply

The power supply used are a Rigol DP831A and a waveform generator DG1062Z

"PSU": {

"ident":["DP831A",

"DG1062Z"],

"Comm": [

["TCP", "134.171.5.188"],

["TCP", "192.168.6.10"]

]

},

## Oscilloscope

For monitoring the bias of the preamp we use a DSO5034 oscilloscope

"OS": {

"ident":["DSO5034"],

"Comm": [

["TCP", "134.171.5.184"]

]

},

# Operation

This section describes the operation of CRIMON. How to start it to record the data and where it can be found. Communication examples will be provided [TBD].

## Benches supported by the monitoring interface

|  |
| --- |
| Mosaic Test Facility (MTF) Instruments in configuration: 2x Lakeshores 340, 1x Pfeiffer TPG 262 controller  Raspberry information: Model 2, login/pwd: bserra/rasptest. Address 134.171.12.57 |
| CRISLER (CRISLER) Instruments in configuration: 2x Lakeshores 336, 2x HMC 8012, 1x Pfeiffer TPG 262 controller, 1x Rigol DP831A, 1x Rigol DG1000z…  Raspberry information: Model 3+, login/pwd: pi/rasptest. Address 134.171.5.190 |

## Starting the monitoring

In order to start the monitoring the software must be started from the raspberry pi that is in charge.

# Data visualisation

After cooling down, the user should have one or several text files with the same format. For visualizing the data, the user can copy the data of each file to an excel spreadsheet. But the sheer volume of data that will be inside the spreadsheet renders excel slow for very simple operations.

Using python, a written script can be used to merge the files together and display the data. This display also allows to see the closest datapoint to the cursor (see python script referred as [AD7](#_Applicable_Documents)).

## Step by step

|  |  |
| --- | --- |
| 1 | Copy the python script **ResultsCooldown.py** (found in [**\\dsraid2\raid\projects\CRISLER\Software\Monitoring\test\_scripts**](file:///\\dsraid2\raid\projects\CRISLER\Software\Monitoring\test_scripts)). Where the text files are. |
| 2 | Execute the script. Two windows should be seen in the display, as they are below. The top one is the GUI, the bottom one is the python kernel. **Closing one or the other will close the program**. |
|  | |
| 3 | When the user click on one of the plots, in the python kernel, a series of values will be displayed. This is the closest data point to where the user clicked.  If the user clicks outside of the plot nothing happens. |
|  | |

## Notes

This script is compatible with data acquired with CRIMON v0.4.4.

**--- End of document ---**