SPIROU-4800-LAM-UM-00961

SPIRou Data Reduction Software

User Manual

0.0.8

For AT-4

N. Cook, F. Bouchy, E. Artigau, I. Boisse, M. Hobson, C. Moutou

2017-11-15

Abstract

This is the guide to installing, running, and using the SPIRou DRS.

Contents

Introduction

This manual describes how to install and run the current version of the spirou DRS (AT-4). The installation process will be simplied in future releases. Some of the chapters give information about indiviudual variables and custom python functions and modules that will not be needed to understand how to run the DRS but are very useful for developmental purposes. Please read the notes below very carefully.

Notes

- This installation assumes you are running bash on a Linux machine.
- There is no need for root access needed for any of these steps.
- The python modules required for this DRS to run (more up-to-date versions are NOT supported) will make current versions of python not work. It is currently recommended that an isolated version of python be used with this version of the DRS (see Section 1.5).
- Variables inside { } are defined in Section 1.2 and used throughout to mean "replace with valued defined in Section 1.2".
- The following denotes a line of text that is to be edited (in a file):

```
VARIABLE_NAME="Variable Value"
```

- The following denotes a console command to run:

```
echo "HelloWorld!"
```

- The following denotes a python command to run:

```
>>> print("HelloWorld!")
```

Chapter 1

Pre-Installation

This file is just for use with the current installation files (version 43), the author does not recommend that this is the final set-up procedure, nor that any of the 'modifications' to the original source code applied here should be used in any final version of the DRS (better solutions should be found).

1.1 Prerequisites

Before one can use the DRS pipeline one must follow the followings steps before installing the pipeline.

1. Add variables to the system environment (Section 1.2)
2. Setup and check the folder structure (Section 1.3)
3. Make sure Fortran and C back-ends are installed (Section 1.4)
4. Install an isolated, clean version of python (Section 1.5)
5. Install required python modules - exact version not older not newer (Section 1.6)

There is no need for root access for any of the following steps.

1.2 Adding variables to the system environment

This can be done one of two ways. The first root (Section 1.2.1) is via a setup file and will allow you to switch on and switch of the environmental variables (to avoid clashes with other programs and to keep ones environment clean). Note with this method you will have to source the setup script before installing or running this code (each time the environment changes) or add the activation to your bashrc. This route is recommended. Otherwise you can follow the second root (Section 1.2.2) and hard-code variables to your '.bashrc' file.

1.2.1 Using a setup script

Open up 'env_setup.sh' and edit the following lines. You will find a copy of 'env_setup.sh' in Appendix A.

1. Set the instrument name {INSTRUMENT_NAME}
2. Directory in which all SPIROU files go {DATA_ROOT}
3. Define installation folder name {INSTALL_ROOT}
4. Define data folder name {DATA_ROOT}

5. Define raw path {DATA_ROOT_RAW}
6. Define reduced path {DATA_ROOT_REDUCED}
7. Define calibDB path {DATA_ROOT_CALIB}
8. Define msg path {DATA_ROOT_MSG}
9. Define tmp path {DATA_ROOT_TMP}
10. Define python version {PYTHON_VERSION}
11. Define python directory (i.e. result of command "which python") {PYTHON_DIR}
12. Define GSL path (default paths if installed may be /usr/local/include/gsl or /opt/gsl) {GSL_DIR}

i.e. these lines in 'env_setup.sh'.

```
INSTRUMENT_NAME="SPIROU"
DIR="/data/spirou/drs"
INSTALL_ROOT="$DIR/INTROOT"
DATA_ROOT="$DIR/data"
DATA_RAW_ROOT="$DATA_ROOT/raw/"
DATA_ROOT_REDUCED="$DATA_ROOT/reduced/"
DATA_ROOT_CALIB="$DATA_ROOT/calibDB"
DATA_ROOT_MSG="$DATA_ROOT/msg/"
DATA_ROOT_TMP="$DATA_ROOT/tmp/"
PYTHON_VERSION="2.7"
PYTHON_DIR="$DIR/python/miniconda2/"
GSL_DIR="$DIR/c-libraries/gsl"
```

where any pre-defined variable can be used with a preceding $ sign (to avoid repetition - coloured above in red).

1.2.2 Adding variables to .bashrc file

Open '~.bashrc' in your favourite text editor.

Add the following to it:

```
export INSTRUMENT={INSTRUMENT_NAME}
export DRS_DATA_RAW={DATA_ROOT_RAW}
export DRS_DATA_REDUC={DATA_ROOT_REDUCED}
export DRS_CALIB_DB={DATA_ROOT_CALIB}
export DRS_DATA_MSG={DATA_ROOT_MSG}
export DRS_DATA_WORKING={DATA_ROOT_TMP}
export TDATA={DATA_ROOT}

export INTROOT={INSTALL_ROOT}
export PATH={INSTALL_ROOT}/bin:{PYTHON_DIR}/bin/:$PATH
export PYTHONPATH=.:{PYTHON_DIR}/lib/python{PYTHON_VERSION}/site-packages/:{INSTALL_ROOT}/bin
export DRS_LOG=1
export PYTHON_INCLUDE_DIR="{PYTHON_DIR}/lib/python{PYTHON_VERSION}/site-packages/numpy/core/include"
export GSL_INCLUDE_DIR={GSL_DIR}/include
export GSL_LIBRARY_DIR={GSL_DIR}/lib

chmod +x "$PYTHON_DIR/bin/conda"
alias conda="$PYTHON_DIR/bin/conda"
chmod +x "$PYTHON_DIR/bin/pip"
alias pip="$PYTHON_DIR/bin/pip"
chmod +x "$PYTHON_DIR/bin/f2py"
alias f2py="$PYTHON_DIR/bin/f2py"
```

where:

- {INSTRUMENT_NAME} = Set the instrument name

- {DATA_ROOT}= Directory in which all SPIROU files go
- {INSTALL_ROOT} = Define installation folder name
- {DATA_ROOT} = Define data folder name
- {DATA_ROOT_RAW} = Define raw path
- {DATA_ROOT_REDUCED} Define reduced path
- {DATA_ROOT_CALIB} = Define calibDB path
- {DATA_ROOT_MSG} = Define msg path
- {DATA_ROOT_TMP} = Define tmp path
- {PYTHON_VERSION} = Define python version
- {PYTHON_DIR} = Define python directory (i.e. result of command "which python")
- {GSL_DIR} = Define GSL path (default paths if installed may be /usr/local/include/gsl or /opt/gsl)

Note: you will have to remove all these environmental variables/aliases to use any other version of python on your system.

1.3 Setup up and check file structure

Make sure the following folders are created:

```
mkdir {DIR}
mkdir {DIR}/{INSTALL_ROOT}/bin
mkdir {DATA_ROOT}
mkdir {DATA_RAW_ROOT}
mkdir {DATA_ROOT_REDUCED}
mkdir {DATA_ROOT_CALIB}
mkdir {DATA_ROOT_MSG}
mkdir {DATA_ROOT_TMP}
```

i.e. for the above 'env_setup.sh' this would be

```
mkdir "/data/spirou/drs"
mkdir "/data/spirou/drs/INTROOT/bin"
mkdir "/data/spirou/drs/data"
mkdir "/data/spirou/drs/data/raw"
mkdir "/data/spirou/drs/data/reduced"
mkdir "/data/spirou/drs/data/calibDB"
mkdir "/data/spirou/drs/data/msg"
mkdir "/data/spirou/drs/data/tmp"
```

1.4 Fortran and C back-end installation

1.4.1 Checking Fortran/C back-ends

Fortran and C compiling is taken care of via python (f2py and distutils respectively).
f2py is installed with miniconda and thus needs to be aliased (as stated above). This is done automatically with the env_setup script or manaully with the following commands added to the '.bashrc' file:

```
chmod +x "$PYTHON_DIR/bin/f2py"
alias f2py="$PYTHON_DIR/bin/f2py"
```

In addition you will need the C package 'GSL', check that it is installed (default paths may be: /usr/local/include/gsl or /opt/gsl). If it is not installed please follow the instructions in Section 1.4.2. Note that the {GSL_DIR} environmental variable must be set and some of the installation files must be modified (currently) in order to point the installation scripts to the correct location (see Section 2.2).

1.4.2 Installing GSL back-ends without root access

To install GSL without root us the following steps:

1. download from http://ftpmirror.gnu.org/gsl/
   ```
   wget http://ftpmirror.gnu.org/gsl/gsl-1.16.tar.gz
   ```
2. create the {GSL_DIR} (from above)
   ```
   mkdir {GSL_DIR}
   ```
3. untar the GSL files
   ```
   tar -xvf gs1-1.16.tar.gz
   ```
4. change to untarred directory
   ```
   cd gs1-1.16/
   ```
5. configure the installation dir for GSL
   ```
   ./configure prefix={GSL_DIR}
   ```
6. build the GSL installation
   ```
   make
   ```
7. install the GSL installation
   ```
   make install
   ```

1.5 Install isolated version of Python

As the current DRS requires specific versions of modules to run we recommend a isolated version of python (as to not interfere with your system or running other python codes).
Python installation must meet the following specifications:

- numpy 1.8.2 (versions later than 1.8.2 are unsupported)
- scipy 0.14
- matplotlib 1.3.1
- pyfits 3.2.4

Hence installing Miniconda (a minimal version of the anaconda python distribution) is the easiest way to achieve this in an isolated environment (as not to destroy any current/system version of python). See section 1.5.1 for Miniconda install instructions.

1.5.1 Installing Miniconda

To install Miniconda follow the steps below:

1. download miniconda from here: https://conda.io/miniconda.html

```
wget https://repo.continuum.io/miniconda/Miniconda2-latest-Linux-x86_64.sh
```

2. run bash script

```
bash Miniconda2-latest-Linux-x86_64.sh
```

Note: choose Miniconda installation directory to match {PYTHON_DIR} above Note: if you wish to activate this environment/use other python installations do not add Miniconda2 install location to PATH in ~/.bashrc

3. add {PYTHON_DIR} to FRONT of PATH environment (temporarily).

```
export PATH={PYTHON_DIR}/bin:$PATH
```

4. check that we are using the correct version of python/conda/pip

```
which python
```

Should read:

```
{PYTHON_DIR}/bin/python
```

or

```
{PYTHON_DIR}/bin/python{PYTHON_VERSION}
```

similarly for

```
which conda
which pip
```

which should read:

```
{PYTHON_DIR}/bin/conda
```

and

```
{PYTHON_DIR}/bin/pip respectively
```

1.6 Install required python modules

1.6.1 Installation

1. install numpy with miniconda

```
conda install numpy==1.8.2
```

```
Fetching package metadata .........
Solving package specifications: .

Package plan for installation in environment /scratch/bin/miniconda2/test:

The following NEW packages will be INSTALLED:

    libgfortran: 1.0-0
    numpy:       1.8.2-py27_1

The following packages will be UPDATED:

    conda:       4.3.21-py27_0 --> 4.3.27-py27hff99c7a_0

          Proceed ([y]/n)? y
```

2. install scipy with miniconda

```
conda install scipy==0.14
```

```
Fetching package metadata ...........
Solving package specifications: .

Package plan for installation in environment /scratch/bin/miniconda2/test:

The following NEW packages will be INSTALLED:

    scipy:     0.14.0-np18py27_0

The following packages will be UPDATED:

    conda-env: 2.6.0-0           --> 2.6.0-h36134e3_1


          Proceed ([y]/n)? y
```

3. install matplotlib with miniconda

```
conda install matplotlib==1.3.1
```

```
Fetching package metadata ...........
Solving package specifications: .

Package plan for installation in environment /scratch/bin/miniconda2/test:

The following NEW packages will be INSTALLED:

    cairo:      1.12.18-0
    dateutil:   2.4.1-py27_0
    freetype:   2.4.10-0
    libpng:     1.5.13-1
    matplotlib: 1.3.1-np18py27_1
    pixman:     0.26.2-0
    py2cairo:   1.10.0-py27_2
    pyqt:       4.10.4-py27_0
    pytz:       2017.2-py27hcac29fa_1
    qt:         4.8.5-0
    sip:        4.15.5-py27_0

The following packages will be DOWNGRADED:

    pyparsing:  2.1.4-py27_0          --> 2.0.1-py27_0


        Proceed ([ y ]/ n ) ? y
```

4. download pyfits 3.2.4 from http://www.stsci.edu/institute/software_hardware/pyfits/Download

```
wget https://pypi.python.org/packages/source/p/pyfits/pyfits-3.2.4.tar.gz
```

5. install pyfits with pip

```
pip install pyfits-3.2.4.tar.gz
```

1.6.2 Checking installed module versions

Before we continue we should check python and module installation.

1. run python

```
python
```

Inside python run following commands

```
>>> import numpy
>>> import matplotlib
>>> import scipy
>>> import pyfits
```

Test the version with:

```
>>> numpy.__version__
1.8.2
>>> matplotlib.__version__
1.3.1
>>> scipy.__version__
0.14.0
>>> pyfits.__version__
3.2.4
```

Chapter 2

Installation

This is just for use with the current installation files (version 43), the author does not recommend that this is the final set-up procedure, nor that any of the 'modifications' to the original source code applied here should be used in any final version of the DRS (better solutions should be found).

2.1 Activate environment

If you followed the steps in 1.2.1 please follow this section, if however you followed the steps in 1.2.2 continue to Section 2.2.
Before running the installation (or before running the code) one must run the following:

```
source env_setup.sh
```

Output should look like this:

```
  ========================================
    Environmental setup for SPIRou Pipeline
  ========================================
    Setting up environment
    Set up for {INSTRUMENT}
      - Python located at: {PYTHON_DIR}
      - GLS located at: {GSL_DIR}
      - data located at:
            {DATA_ROOT}
            {DATA_RAW_ROOT}
            {DATA_ROOT_REDUCED}
            {DATA_ROOT_CALIB}
            {DATA_ROOT_MSG}
            {DATA_ROOT_TMP}

    Done
```

Note to deactivate type

```
source env_setup.sh --clean
```

2.2 Downloading the preparing the install scripts

Minor modifications need to be made to the code to allow a isolated version of GSL to be used. If and only if your GSL is installed to '/opt/gsl/' will the code install correctly with out this modification. The other steps are as in the original installation procedure.

1. change directory to {DIR}

   ```
   {DIR}
   ```

2. download source code for spirou, location should be {DIR}/spirou

   ```
   svn co https://svn.lam.fr/repos/spirou/trunk spirou
   ```

1. change directory to source files

```
cd {DIR}/spirou/src
```

2. find and change {DIR}/spirou/src/C/setup.py

   a) Below the line

```
python_include_dir = os.getenv('PYTHON\_INCLUDE\_DIR')
```

   add the following:

```
gsl_include_dir = os.getenv('GSL_INCLUDE_DIR')
gsl_library_dir = os.getenv('GSL_LIBRARY_DIR')
```

   b) Find and replace all instances of

```
include_dirs = [python_include_dir,'/opt/gsl/include'],
```

   with

```
include_dirs = [python_include_dir, gsl_include_dir],
```

   c) Find and replace all instances of

```
library_dirs = ['/opt/gsl/lib'],
```

   with

```
library_dirs = [gsl_library_dir],
```


2.3 Running installation scripts

run installation scripts

```
./hardrsInstall SPIROU 2.7
./scriptInstall SPIROU 2.7
```

Note: you may need to run chmod on these script in order to run them

```
chmod +x hardrsInstall
chmod +x scriptInstall
```

2.4 Subversion

2.4.1 Updating local version

The code currently uses 'Subversion' (SVN) to monitor the different versions of the code that is under development. To update a local version of the DRS:

1. Go inside the {DIR}/spirou folder:

```
cd \{DIR\}/spirou
```

2. Type:

```
svn update
```

   You will then receive information on the DRS version.

3. Then you need to re-install the DRS. This will rewrite all codes in the {INSTALL_ROOT} folder.

   a) change directory to source files

   ```
   cd {DIR}/spirou/src
   ```

   b) verify that the following lines are in {DIR}/spirou/src/C/setup.py

   ```
   gsl_include_dir = os.getenv('GSL_INCLUDE_DIR')    # default: '/opt/gsl/include'
   gsl_library_dir = os.getenv('GSL_LIBRARY_DIR')    # default: '/opt/gsl/lib'
   ```

   if they are not do the following steps:

   i. Below the line

   ```
   python_include_dir = os.getenv('PYTHON\_INCLUDE\_DIR')
   ```

   add the following:

   ```
   gsl_include_dir = os.getenv('GSL_INCLUDE_DIR')
   gsl_library_dir = os.getenv('GSL_LIBRARY_DIR')
   ```

   ii. Find and replace all instances of

   ```
   include_dirs = [python_include_dir,'/opt/gsl/include'],
   ```

   with

   ```
   include_dirs = [python_include_dir, gsl_include_dir],
   ```

   iii. Find and replace all instances of

   ```
   library_dirs = ['/opt/gsl/lib'],
   ```

   with

   ```
   library_dirs = [gsl_library_dir],
   ```

   c) Reinstall the DRS using:

   ```
   cd src
   ./hardrsInstall SPIROU 2.7
   ./scriptInstall SPIROU 2.7
   ```

You are now ready to work with the latest version.

2.4.2 Adding new files to the shared DRS

Note: this is only to be done if your new functions are running correctly.

1. Go inside the {DIR}/spirou folder:

   ```
   cd \{DIR\}/spirou
   ```

2. run the update on the svn

   ```
   svn update
   ```

3. copy your modifications into the source code:

   ```
   cp {modified file} src/{location of new file}
   ```

4. Add the file to the list of pending SVN files:

   ```
   svn add src/{location of new file}
   ```

5. once all new files are added, commit to releasing the modifications to the SVN:

   ```
   svn commit -m 'modification of {file name}'
   ```

2.4.3 Useful links about subversion

- resentation: http://multithread.org/files/presentations/subversion/subversion_tutorial.pdf
- Book: http://svnbook.red-bean.com/
- Introduction to subversion: https://dev.nozav.org/intro_svn.html

Chapter 3

Using the DRS

3.1 Running the code

To run the DRS python (assuming env_setup is activated, see Section 2.1) type:

```
{PROGRAM NAME} {FOLDER} {FILES}
```

i.e.

```
cal_DARK_spirou {YYMMDD} {Filenames*}
```

instead of python
Note: location should of script should be {DIR}/{INSTALL_ROOT}/bin/DRS{INSTRUMENT}
i.e. for 'cal_DARK_spirou' in the {DATA_RAW_ROOT}/YYMMDD directory one would result in something like the following:

```
cal_DARK_spirou YYMMDD Filenames*
```

```
19:44:52.5 -   || ****************************************
19:44:52.5 -   || * SPIROU (@) Geneva Observatory ()
19:44:52.5 -   || ****************************************
19:44:52.5 -   ||(dir_data_raw)      DRS_DATA_RAW=/data/spirou/drs/data/raw/
19:44:52.5 -   ||(dir_data_reduc)    DRS_DATA_REDUC=/data/spirou/drs/data/reduced/
19:44:52.5 -   ||(dir_drs_config)    DRS_CONFIG=/data/spirou/drs/INTROOT/DRS_SPIROU/config/
19:44:52.5 -   ||(dir_calib_db)      DRS_CALIB_DB=/data/spirou/drs/data/calibDB
19:44:52.5 -   ||(dir_data_msg)      DRS_DATA_MSG=/data/spirou/drs/data/msg/
19:44:52.5 -   ||(print_log)         DRS_LOG=1         %(0: minimum stdin-out logs)
19:44:52.5 -   ||(plot_graph)        DRS_PLOT=NONE            %(def/undef/trigger)
19:44:52.5 -   ||(used_date)         DRS_USED_DATE=undefined
19:44:52.5 -   ||(working_dir)       DRS_DATA_WORKING=/data/spirou/drs/data/tmp/
19:44:52.5 -   ||                    DRS_INTERACTIVE is not set, running on-line mode
19:44:52.5 -   |-c:+[...]|Now running : -c on file(s):  dark_dark...
19:44:52.5 -   |-c:+[...]|On directory /data/spirou/drs/data/raw/20170811
19:44:52.5 -   |-c:+[...]|ICDP loaded from: /data/spirou/drs/INTROOT/DRS_SPIROU/config/hadmrICDP_SPIROU.py
19:44:52.5 - * |-c:+[...]|Now processing Image TYPE DARK with -c recipe
19:44:52.5 -   |-c:+[...]|Reading Image /data/spirou/drs/data/raw/20170811/dark_dark02d.fits
19:44:52.5 -   |-c:+[...]|Image 2048x2048 loaded
```

3.2 Working example of the code for SPIRou

3.2.1 Overview

For this example all files are from:

```
spirou@10.102.14.81:/data/RawImages/H2RG-AT4/AT4-04/2017-07-10_15-36-18/ramps/
```

Will also need current WAVE file from here:

```
spirou@10.102.14.81:/data/reduced/DATA-CALIB/spirou_wave_ini3.fits
```

This assumes a correct setup using all variables as in previous sections i.e.:

```
INSTRUMENT_NAME="SPIROU"
DIR="/data/spirou/drs"
INSTALL_ROOT="$DIR/INTROOT"
DATA_ROOT="$DIR/data"
DATA_RAW_ROOT="$DATA_ROOT/raw/"
DATA_ROOT_REDUCED="$DATA_ROOT/reduced/"
DATA_ROOT_CALIB="$DATA_ROOT/calibDB"
DATA_ROOT_MSG="$DATA_ROOT/msg/"
DATA_ROOT_TMP="$DATA_ROOT/tmp/"
PYTHON_VERSION="2.7"
PYTHON_DIR="$DIR/python/miniconda2/"
GSL_DIR="$DIR/c-libraries/gsl"
```

All files from

```
spirou@10.102.14.81:/data/RawImages/H2RG-AT4/AT4-04/2017-07-10_15-36-18/ramps/
```

must be placed in a folder named '20170710' in {DATA_RAW_ROOT} (/data/spirou/drs/-data/raw).

Starting with RAMP files and ending with extracted orders and calculated drifts we need to run six codes:

1. cal_DARK_spirou.py (See Section 6.2)
2. cal_loc_RAW_spirou.py (×2) (See Section 6.3)
3. cal_SLIT_spirou.py (See Section 6.4)
4. cal_FF_RAW_spirou.py (×2) (See Section 6.5)
5. (add spirou_wave_ini3.fits to calibDB)
6. cal_extract_RAW_spirouAB.py and cal_extract_RAW_spirouC.py (many times) (See Section 6.6)
7. cal_DRIFT_RAW_spirou (See Section 6.7)

3.2.2 Command-by-command run through

We next list the exact commands used to go from the RAMPs to the extract orders and calculated drifts.

1. Change to the {DIR} directory

   ```
   cd DIR
   ```

2. Activate the environment (in bash only if not using the '.bashrc' method)

   ```
   source env_setup.sh
   ```

3. Check that environment is active (only if using 'env_setup.sh' and if not using the '.bashrc' method).

```
echo $DRS_ACTIVE
```

result should be = 1

4. run the dark extraction on the 'dark_dark' file:

```
cal_DARK_spirou.py 20170710 dark_dark02d406.fits
```

5. run the order localisation on the 'dark_flat' files:

```
cal_loc_RAW_spirou.py 20170710 dark_flat02f10.fits dark_flat03f10.fits dark_flat04f10.fits dark_flat05f10.
    fits dark_flat06f10.fits
```

6. run the order localisation on the 'flat_dark' files:

```
cal_loc_RAW_spirou.py 20170710 flat_dark02f10.fits flat_dark03f10.fits flat_dark04f10.fits flat_dark05f10.
    fits flat_dark06f10.fits
```

7. run the slit calibration on the 'fp_fp' files.

```
cal_SLIT_spirou.py 20170710 fp_fp02a203.fits fp_fp03a203.fits fp_fp04a203.fits
```

8. run the flat field creation on the 'dark_flat' files:

Note: if using same files as above you will get an error message when running the file. To solve this open the 'master_calib_SPIROU.txt' file located in {DATA_ROOT_CALIB}. Edit the unix date in the line that begins 'TILT' so that it is less than the unix date on rwos 'ORDER_PROFIL_AB' (i.e. change it from 1499707515.0 to 1499705515.0).

i.e. the 'master_calib_SPIROU.txt' file should look go from

```
DARK 20170710 dark_dark02d406.fits 07/10/17/16:37:48 1499704668.0
ORDER_PROFIL_C 20170710 dark_flat02f10_order_profil_C.fits 07/10/17/17:03:50 1499706230.0
LOC_C 20170710 dark_flat02f10_loco_C.fits 07/10/17/17:03:50 1499706230.0
ORDER_PROFIL_AB 20170710 flat_dark02f10_order_profil_AB.fits 07/10/17/17:07:08 1499706428.0
LOC_AB 20170710 flat_dark02f10_loco_AB.fits 07/10/17/17:07:08 1499706428.0
TILT 20170710 fp_fp02a203_tilt.fits 07/10/17/17:25:15 1499707515.0
```

to this:

```
DARK 20170710 dark_dark02d406.fits 07/10/17/16:37:48 1499704668.0
ORDER_PROFIL_C 20170710 dark_flat02f10_order_profil_C.fits 07/10/17/17:03:50 1499706230.0
LOC_C 20170710 dark_flat02f10_loco_C.fits 07/10/17/17:03:50 1499706230.0
ORDER_PROFIL_AB 20170710 flat_dark02f10_order_profil_AB.fits 07/10/17/17:07:08 1499706428.0
LOC_AB 20170710 flat_dark02f10_loco_AB.fits 07/10/17/17:07:08 1499706428.0
TILT 20170710 fp_fp02a203_tilt.fits 07/10/17/17:25:15 1499705515.0
```

```
cal_FF_RAW_spirou.py 20170710 dark_flat02f10.fits dark_flat03f10.fits dark_flat04f10.fits dark_flat05f10.
    fits dark_flat06f10.fits
```

9. run the flat field creation on the 'flat_dark' files:

```
cal_FF_RAW_spirou.py 20170710 flat_dark02f10.fits flat_dark03f10.fits flat_dark04f10.fits flat_dark05f10.
    fits flat_dark06f10.fits
```

10. Currently we do not create a new wavelength calibration file for this run. Therefore we need to get one from here:

```
spirou@10.102.14.81:/data/reduced/DATA-CALIB/spirou_wave_ini3.fits
```

then place it in the {DATA_ROOT_CALIB} folder. You will also need to edit the 'master_calib_SPIROU.txt' file located in {DATA_ROOT_CALIB}. Add the folloing line to 'master_calib_SPIROU.txt'

```
WAVE 20170710 spirou_wave_ini3.fits 07/10/17/17:03:50 1499706230.0
```

and the 'master_calib_SPIROU.txt' should look like this:

```
DARK 20170710 dark_dark02d406.fits 07/10/17/16:37:48 1499704668.0
ORDER_PROFIL_C 20170710 dark_flat02f10_order_profil_C.fits 07/10/17/17:03:50 1499706230.0
LOC_C 20170710 dark_flat02f10_loco_C.fits 07/10/17/17:03:50 1499706230.0
ORDER_PROFIL_AB 20170710 flat_dark02f10_order_profil_AB.fits 07/10/17/17:07:08 1499706428.0
LOC_AB 20170710 flat_dark02f10_loco_AB.fits 07/10/17/17:07:08 1499706428.0
TILT 20170710 fp_fp02a203_tilt.fits 07/10/17/17:25:15 1499705515.0
FLAT_C 20170710 dark_flat02f10_flat_C.fits 07/10/17/17:03:50 1499706230.0
WAVE 20170710 spirou_wave_ini3.fits 07/10/17/17:03:50 1499706230.0
```

11. run the extraction files on the 'hcone_dark', 'dark_hcone', 'hcone_hcone', 'dark_dark_AHC1', 'hctwo_dark', 'dark_hctwo', 'hctwo-hctwo', 'dark_dark_AHC2' and 'fp_fp' files:

```
cal_extract_RAW_spirouAB.py 20170710 hcone_dark02c61.fits  hcone_dark03c61.fits  hcone_dark04c61.fits
     hcone_dark05c61.fits  hcone_dark06c61.fits

cal_extract_RAW_spirouC.py 20170710 dark_hcone02c61.fits  dark_hcone03c61.fits  dark_hcone04c61.fits
     dark_hcone05c61.fits  dark_hcone06c61.fits

cal_extract_RAW_spirouAB.py 20170710 hcone_hcone02c61.fits  hcone_hcone03c61.fits  hcone_hcone04c61.fits
      hcone_hcone05c61.fits  hcone_hcone06c61.fits

cal_extract_RAW_spirouC.py 20170710 hcone_hcone02c61.fits  hcone_hcone03c61.fits  hcone_hcone04c61.fits
     hcone_hcone05c61.fits  hcone_hcone06c61.fits

cal_extract_RAW_spirouAB.py 20170710 dark_dark_AHC102d61.fits  dark_dark_AHC103d61.fits
     dark_dark_AHC104d61.fits  dark_dark_AHC105d61.fits  dark_dark_AHC106d61.fits

cal_extract_RAW_spirouC.py 20170710 dark_dark_AHC102d61.fits  dark_dark_AHC103d61.fits
     dark_dark_AHC104d61.fits  dark_dark_AHC105d61.fits  dark_dark_AHC106d61.fits

cal_extract_RAW_spirouAB.py 20170710 hctwo_dark02c61.fits  hctwo_dark03c61.fits  hctwo_dark04c61.fits
     hctwo_dark05c61.fits  hctwo_dark06c61.fits

cal_extract_RAW_spirouC.py 20170710 hctwo_dark02c61.fits  hctwo_dark03c61.fits  hctwo_dark04c61.fits
     hctwo_dark05c61.fits  hctwo_dark06c61.fits

cal_extract_RAW_spirouAB.py 20170710 dark_hctwo02c61.fits  dark_hctwo03c61.fits  dark_hctwo04c61.fits
     dark_hctwo05c61.fits  dark_hctwo06c61.fits

cal_extract_RAW_spirouC.py 20170710 dark_hctwo02c61.fits  dark_hctwo03c61.fits  dark_hctwo04c61.fits
     dark_hctwo05c61.fits  dark_hctwo06c61.fits

cal_extract_RAW_spirouAB.py 20170710 hctwo-hctwo02c61.fits  hctwo-hctwo03c61.fits  hctwo-hctwo04c61.fits
      hctwo-hctwo05c61.fits  hctwo-hctwo06c61.fits

cal_extract_RAW_spirouC.py 20170710 hctwo-hctwo02c61.fits  hctwo-hctwo03c61.fits  hctwo-hctwo04c61.fits
     hctwo-hctwo05c61.fits  hctwo-hctwo06c61.fits

cal_extract_RAW_spirouAB.py 20170710 dark_dark_AHC202d61.fits  dark_dark_AHC203d61.fits
     dark_dark_AHC204d61.fits  dark_dark_AHC205d61.fits  dark_dark_AHC206d61.fits

cal_extract_RAW_spirouC.py 20170710 dark_dark_AHC202d61.fits  dark_dark_AHC203d61.fits
     dark_dark_AHC204d61.fits  dark_dark_AHC205d61.fits  dark_dark_AHC206d61.fits

cal_extract_RAW_spirouAB.py 20170710 fp_fp02a203.fits fp_fp03a203.fits fp_fp04a203.fits

cal_extract_RAW_spirouC.py 20170710 fp_fp02a203.fits fp_fp03a203.fits fp_fp04a203.fits
```

12. run the drift calculation on the 'fp_fp' files:

```
cal_DRIFT_RAW_spirou.py 20170710 fp_fp02a203.fits fp_fp03a203.fits fp_fp04a203.fits
```

Chapter 4

Data Architecture

Below we describe the data architecture, first by providing the installed file structure in a tree diagram (4.1), and then describing individual directories.

4.1 Installed file structure

The file structure should look as follows:

```
{DIR}
├── {INSTALL_ROOT}
│   ├── bin
│   │   └── ..........................................Recipe symbolic-link
│   └── DRS_{INSTRUMENT_NAME}
│       └── ..........................................Worker functions
├── spirou
│   ├── read_harps.py
│   └── src
│       └── ..........................................source installation files
└── {DATA_ROOT}*
    ├── raw
    │   └── YYYYMMDD ..................................Observation directory
    │       └── ......................................Raw observation files
    ├── reduced
    ├── calibDB
    ├── msg
    └── tmp
```

* Note that this is the recommended file structure and raw, reduced, calibDB, msg and tmp can be changed using the {DATA_ROOT_RAW}, {DATA_ROOT_REDUCED}, {DATA_ROOT_CALIB}, {DATA_ROOT_MSG}, and {DATA_ROOT_TMP} variables in 1.2.

i.e. for the given 'env_setup.sh' (Appendix A) this would be:

```
/data/spirou/drs
├── INTROOT/
│   ├── bin
│   │   └── ..................................................Recipe symbolic-link
│   └── DRS_SPIROU
│       └── ..........................................................Worker functions
├── data
│   ├── raw
│   │   └── YYYYMMDD ..................................................night_repository
│   │       └── ..................................................Raw observation files
│   ├── reduced
│   ├── calibDB
│   ├── msg
│   └── tmp
└── spirou
    ├── read_harps.py
    └── src
        └── ..................................................source installation files
```

4.2 The Installation root directory

The {INSTALL_ROOT} contains all the installed recipes, worker functions and configuration files needed to run the DRS. The file structure is set up as below:

```
/data/spirou/drs
├── {INSTALL_ROOT}
│   ├── bin
│   │   └── ..........................................................................Recipes
│   └── DRS_SPIROU
```

4.2.1 The bin directory

The bin directory is located in the {INSTALL_ROOT} directory. This contains symbolic links to the DRS recipes.

4.2.2 The DRS_SPIROU directory

The DRS_SPIROU directory contains all the worker functions and configuration files for the DRS. The file structure is as follows:

```
DRS_SPIROU
├── C
│   └── ..................................................The C programs and setup.py
├── config
│   └── ..........................................................The config files
├── docs
├── fortran
│   └── ........................................The fortran programs and install.csh
├── man
├── python
│   ├── C-modules
│   ├── f2pymodules
│   ├── Modules
│   └── Recipes
└── util
```

The C and Fortran functions are found in their own directories. The configuration files are explained in Section 5.1.

The python directory contains the translation of the C and fortran functions into python (C-modules and f2pymodules directories).

The functions 'startup.py' and 'startup_recipes.py' are called automatically by the recipes.

'python.csh' is the custom python environment for the DRS.

The python modules are explained in Section 7 and the recipes are explained in Section 6.

4.3 The data root directory

This is the directory where all the data should be stored. The default and recommended design is to have {DATA_ROOT_RAW}, {DATA_ROOT_REDUCED}, {DATA_ROOT_CALIB}, {DATA_ROOT_MSG}, and {DATA_ROOT_TMP} as sub-directories of {DATA_ROOT}. However as in Section 1.2 these sub-directories can be defined elsewhere.

4.3.1 The raw and reduced data directories

The raw observed data is stored under the {DATA_ROOT_RAW} path, the files are stored by night in the form YYYYMMDD.

The file structure can be seen below:

```
{DATA_ROOT_RAW}
├── YYYYMMDD ..........................................night_repository
│   ├── ..............................................Raw observation files
│   ├── dark_dark{name}.fits
│   ├── dark_flat{name}.fits
│   ├── flat_dark{name}.fits
│   └── fp_fp{name}.fits
```

4.4 The calibration database directory

```
{DATA_ROOT}
├── calibDB or {DATA_ROOT_CALIB}
│   ├── master_calib_SPIROU.txt
│   └── ..........................................The calibration fits files
```

The calibDB contains all the calibration files that pass the qulaity tests and a test file 'master_calib.txt'. It is located at {DATA_ROOT_CALIB} or if this is not defined is located by default at the {DATA_ROOT} directory.

Each line in this file is a uniqie calibration file and lines are formatted in the following manner:

```
{key} {night_repository} {filename} {human readable date} {unix time}
```

where

- {key} is a code assigned for each type of calibration file. Currently accepted keys are:
  - DARK - Created from cal_DARK_spirou
  - ORDER_PROFIL_{fiber} - Created in cal_loc_RAW_spirou
  - LOC_C - Created in cal_loc_RAW_spirou
  - TILT - Created in cal_SLIT_spirou
  - FLAT_{fiber} - Created in cal_FF_RAW_spirou
  - WAVE - Currently manually added
- {night_repository} is the raw data observation directory (in {DATA_ROOT_RAW}) normally in the form YYYYMMDD.
- {filename} is the filename of the calibration file (located in the calibDB).
- {human readable date} is the date in DD/MM/YY/HH:MM:SS format taken from the header of the file that created the calibration file (using the header keyword 'ACQTIME1').
- {unix time} is the time (as in {human readable date}) but in unix time (in seconds).

An example working master_calib_SPIROU.txt is shown below (assuming the listed files are present in {DATA_ROOT_CALIB})

```
DARK 20170710 dark_dark02d406.fits 07/10/17/16:37:48 1499704668.0
ORDER_PROFIL_C 20170710 dark_flat02f10_order_profil_C.fits 07/10/17/17:03:50 1499706230.0
LOC_C 20170710 dark_flat02f10_loco_C.fits 07/10/17/17:03:50 1499706230.0
ORDER_PROFIL_AB 20170710 flat_dark02f10_order_profil_AB.fits 07/10/17/17:07:08 1499706428.0
LOC_AB 20170710 flat_dark02f10_loco_AB.fits 07/10/17/17:07:08 1499706428.0
TILT 20170710 fp_fp02a203_tilt.fits 07/10/17/17:25:15 1499705515.0
FLAT_C 20170710 dark_flat02f10_flat_C.fits 07/10/17/17:03:50 1499706230.0
WAVE 20170710 spirou_wave_ini3.fits 07/10/17/17:03:50 1499706230.0
```

Chapter 5

Constants and Keywords

5.1 The configuration files

The configurations files are found in the {INSTALL_ROOT}/DRS_{INSTRUMENT_NAME}/config folder i.e. for our example in Section 1.2 this would be '/data/spirou/drs/INTROOT/DRS_SPIROU/config/'

The file structure should look as follows:

```
{DIR}
└── {INSTALL_ROOT}
    └── config
        ├── compute_CaII_HARPS.py
        ├── correct_geom_coralie.py
        ├── correct_geom_elodie.py
        ├── correct_geom_harpn.py
        ├── correct_meanpondref_CORALIE.py
        ├── correct_meanpondref_EGGS.py
        ├── correct_meanpondref_ELODIE.py
        ├── correct_meanpondref_HARPS.py
        ├── db_filter_CORALIE.py
        ├── db_filter_EGGS.py
        ├── db_filter_HARPN.py
        ├── db_filter_HARPS.py
        ├── dic_recipes_CORALIE.py
        ├── dic_recipes_EGGS.py
        ├── dic_recipes_ELODIE-S.py
        ├── dic_recipes_HARPN.py
        ├── dic_recipes_HARPS.py
        ├── hadmrICDP_CORALIE07.py
        ├── hadmrICDP_CORALIE14.py
        ├── hadmrICDP_CORALIE98_dec04.py
        ├── hadmrICDP_CORALIE98_jun99.py
        ├── hadmrICDP_CORALIE98_oct98.py
        ├── hadmrICDP_CORALIE98_sep02.py
        ├── hadmrICDP_EGGS.py
        ├── hadmrICDP_ELODIE.py
        ├── hadmrICDP_ELODIE-S.py
        ├── hadmrICDP_HARPN_may12.py
        ├── hadmrICDP_HARPN.py
        ├── hadmrICDP_HARPN_sep12.py
        ├── hadmrICDP_HARPS.py
        ├── hadmrICDP_SOPHIE.py
        └── hadmrICDP_SPIROU_H2RG.py
```

```
├── hadmrICDP_SPIROU_H4RG.py
├── hadmrICDP_SPIROU.py
├── kw_allocate_CORALIE.py
├── kw_allocate_ELODIE.py
├── kw_allocate_HARPN.py
├── kw_allocate_HARPS.py
├── kw_allocate_SOPHIE.py
├── kw_allocate_SPIROU.py
├── KW_CORALIE14.py
├── KW_CORALIE.py
├── kw_display_CORALIE.py
├── kw_display_ELODIE.py
├── kw_display_HARPN.py
├── kw_display_HARPS.py
├── kw_display_SOPHIE.py
├── kw_display_SPIROU.py
├── KW_DRS.py
├── KW_ELODIE.py
├── KW_HARPN.py
├── KW_HARPS.py
├── kw_lamp_CORALIE.py
├── kw_list_CORALIE.py
├── kw_list_ELODIE.py
├── kw_list_HARPN.py
├── kw_list_HARPS.py
├── kw_list_SOPHIE.py
├── kw_list_SPIROU.py
├── kw_pol.py
├── KW_SOPHIE.py
├── KW_SPIROU.py
├── special_config_CORALIE98.py
├── special_config_ELODIE-S.py
├── special_config_HARPN.py
└── special_wave_CORALIE_initial.py
```

These files contain most the variables that are used throughout the DRS (unfortunately hard-coded into the program files). The descriptions of used constants and keywords, the recipe(s) that use them and their location (or locations) are defined below.

5.2 Environmental variables

Some constants are set via the environmental variables (Section 1.2), they are set either via sourcing a shell scripted (Section 1.2) or manually adding them (i.e. via the '.bashrc' file, see Section 1.2.2).

- {INSTRUMENT_NAME} = Set the instrument name
- {DATA_ROOT}= Directory in which all SPIROU files go

- {INSTALL_ROOT} = Define installation folder name
- {DATA_ROOT} = Define data folder name
- {DATA_ROOT_RAW} = Define raw path
- {DATA_ROOT_REDUCED} Define reduced path
- {DATA_ROOT_CALIB} = Define calibDB path
- {DATA_ROOT_MSG} = Define msg path
- {DATA_ROOT_TMP} = Define tmp path
- {PYTHON_VERSION} = Define python version
- {PYTHON_DIR} = Define python directory (i.e. result of command "which python")
- {GSL_DIR} = Define GSL path (default paths if installed may be /usr/local/include/gsl or /opt/gsl)

In addition some other environmental variables are set (or need to be set).

- {DRS_VERSION} = The current version of the DRS
- {DRS_PLOT} = If '1' then plots will be shown, if undefined then plot will not be shown, if 'trigger' will send commands to a file for later plotting.
- {DRS_DEBUG} = The current debug level, default = undefined or = 0, depending on the integer value will print/plot various parts of debugging code.
- {DRS_CONFIG} = Define the config path (defaults to {DATA_ROOT}/config)
- {DRS_MAN} = Define the manual path (defaults to {DATA_ROOT}/man)
- {DRS_DATA_WORKING} = Defines the current working directory (defaults to {DATA_ROOT_TMP}) ?

5.3 Constants

Below are the listed currently used constants in table format. The 'Name' column refers to the name of the constant (and what to search for), the 'Value' column is the default value, the 'Description' column provides a brief description of what the variable controls, the 'Used in' column shows which recipes use this variable and the 'Location' column defines the file or files in which this variable is located (they may be hardcoded in recipe files thus the location may not be a config file), order of precedence is indicated but may be wrong (probably best to change call instances). These are split into catagories for ease of reading.

Note: This section is currently updated for 'cal_DARK_spirou.py','cal_loc_RAW_spirou.py', 'cal_SLIT_spirou.py', 'cal_ff_RAW_spirou.py' and 'cal_DRIFT_RAW_spirou.py' only.

5.3.1 General constants

- **ic_disptimeout:**

  ic_disptimeout = 0.5 / Interval between plots (for certain iteractive graphs)
  Used in: cal_loc_RAW_spirou.py
  Defined in: hadmrICDP_SPIROU.py

5.3.2 Image Variables

- **ic_ccdx_blue:**

  ic_ccdx_blue = 2048-arange(200,1500) / Resize blue window x array
  Used in: cal_DARK_spirou.py
  Defined in: cal_DARK_spirou.py

- **ic_ccdy_blue:**

  ic_ccdy_blue = 2048-arange(20,350) / Resize blue window y array
  Used in: cal_DARK_spirou.py
  Defined in: cal_DARK_spirou.py

- **ic_ccdx_red:**

  ic_ccdx_red = 2048-arange(20,1750 / Resize red window x array
  Used in: cal_DARK_spirou.py
  Defined in: cal_DARK_spirou.py

- **ic_ccdy_red:**

  ic_ccdy_red = 2048-arange(1570,1910) / Resize red window y array
  Used in: cal_DARK_spirou.py
  Defined in: cal_DARK_spirou.py

- **ic_ccdx:**

  ic_ccdx = arange(5,2040) / Resize array for x
  Used in: cal_DARK_spirou.py, cal_loc_RAW_spirou.py, cal_SLIT_spirou.py, cal_ff_RAW_spirou.py, cal_extract_RAW_spirou{fiber}.py, cal_extract_RAW_spirouALL.py
  Defined in: cal_DARK_spirou.py, cal_loc_RAW_spirou.py, cal_SLIT_spirou.py, cal_ff_RAW_spirou.py, cal_extract_RAW_spirou{fiber}.py, cal_extract_RAW_spirouALL.py, hadmrICDP_SPIROU.py

- **ic_ccdy:**

  ic_ccdy = arange(5,1935) / Resize array for y
  Used in: cal_DARK_spirou.py, cal_loc_RAW_spirou.py, cal_SLIT_spirou.py, cal_ff_RAW_spirou.py, cal_extract_RAW_spirou{fiber}.py, cal_extract_RAW_spirouALL.py
  Defined in: cal_DARK_spirou.py, cal_loc_RAW_spirou.py, cal_SLIT_spirou.py, cal_ff_RAW_spirou.py, cal_extract_RAW_spirou{fiber}.py, cal_extract_RAW_spirouALL.py, hadmrICDP_SPIROU.py

5.3.3 Fiber variables

Note: these variables depend on prefix of input file. First entry is for 'flat_dark' second is for 'dark_flat'

- **fiber:**
  fiber = AB, C / Fiber Name
  Used in: cal_loc_RAW_spirou.py
  Defined in: cal_loc_RAW_spirou.py

- **nbfiber:**
  nbfiber = 2, 1 / Number of fibers
  Used in: cal_loc_RAW_spirou.py
  Defined in: cal_loc_RAW_spirou.py

- **ic_first_ord_jump:**
  ic_first_ord_jump = 2, 0 / Number of orders to skip at start of image
  Used in: cal_loc_RAW_spirou.py
  Defined in: cal_loc_RAW_spirou.py, hadmrICDP_SPIROU.py

- **ic_locnbmaxo:**
  ic_locnbmaxo = 72, 36 / Maximum number of order to use
  Used in: cal_loc_RAW_spirou.py
  Defined in: cal_loc_RAW_spirou.py, hadmrICDP_SPIROU.py

- **qc_loc_nbo:**
  qc_loc_nbo = 72, 36 / Quality control "normal" number of orders on fiber
  Used in: cal_loc_RAW_spirou.py
  Defined in: cal_loc_RAW_spirou.py, hadmrICDP_SPIROU.py

5.3.4 Dark Variables

- **cutlimit:**
  cutlimit = 100 / Define a bad pixel cut limit (in ADU/s)
  Used in: cal_DARK_spirou.py
  Defined in: cal_DARK_spirou.py

- **qmin, qmin2, qmin3:**
  qmin, qmin2, qmin3 = 5 / The lower percentile (0 - 100) for dead pixels
  Used in: cal_DARK_spirou.py
  Defined in: cal_DARK_spirou.py

- **qmax, qmax2, qmax3:**
  qmax, qmax2, qmax3 = 95 / The upper percentile (0 - 100) for dead pixels
  Used in: cal_DARK_spirou.py
  Defined in: cal_DARK_spirou.py

5.3.5 Localisation parameters

- **size:**

  size = 10 / Size of the order_profile smoothed box (from pixel - size to pixel + size)
  Used in: cal_loc_RAW_spirou.py
  Defined in: cal_loc_RAW_spirou.py

- **ic_offset:**

  ic_offset = 40 / row number of image to start processing at
  Used in: cal_loc_RAW_spirou.py
  Defined in: cal_loc_RAW_spirou.py

- **ic_ccdcolc:**

  ic_ccdcolc = 1000 / Definition of the central column
  Used in: cal_loc_RAW_spirou.py, cal_SLIT_spirou.py
  Defined in: cal_loc_RAW_spirou.py, hadmrICDP_SPIROU.py

- **ic_ext_window:**

  ic_ext_window = 12 / Definition of the extraction window size (half size)
  Used in: cal_loc_RAW_spirou.py
  Defined in: cal_loc_RAW_spirou.py

- **ic_ccdgap:**

  ic_ccdgap = 0 / Definition of the gap index in the selected area
  Used in: cal_loc_RAW_spirou.py
  Defined in: hadmrICDP_SPIROU.py

- **ic_locstepc:**

  ic_locstepc = 20 / Define the column separation for fitting orders
  Used in: cal_loc_RAW_spirou.py
  Defined in: cal_loc_RAW_spirou.py, hadmrICDP_SPIROU.py

- **ic_widthmin:**

  ic_widthmin = 5 / Define minimum width of order to be accepted
  Used in: cal_loc_RAW_spirou.py
  Defined in: cal_loc_RAW_spirou.py

- **ic_locnbpix:**

  ic_locnbpix = 45 / Half spacing between orders
  Used in: cal_loc_RAW_spirou.py
  Defined in: cal_loc_RAW_spirou.py, hadmrICDP_SPIROU.py

- **ic_min_amplitude:**

  ic_min_amplitude = 100 / Minimum amplitude to accept (in e-)
  Used in: cal_loc_RAW_spirou.py
  Defined in: hadmrICDP_SPIROU.py

- **ic_locseuil:**
  ic_locseuil = 0.2 / Normalised amplitude threshold to accept pixels for background calculation
  Used in: cal_loc_RAW_spirou.py
  Defined in: cal_loc_RAW_spirou.py

- **satseuil:**
  satseuil = `64536 * ccdgain * nbframes` / Saturation threshold on order profile plot
  Used in: cal_loc_RAW_spirou.py
  Defined in: cal_loc_RAW_spirou.py

- **ic_locdfitc:**
  ic_locdfitc = 5 / Order of polynomial to fit for positions
  Used in: cal_loc_RAW_spirou.py,KW_DRS.py
  Defined in: cal_loc_RAW_spirou.py, hadmrICDP_SPIROU.py

- **ic_locdfitw:**
  ic_locdfitw = 5 / Order of polynomial to fit for widths
  Used in: cal_loc_RAW_spirou.py,KW_DRS.py
  Defined in: cal_loc_RAW_spirou.py, hadmrICDP_SPIROU.py

- **ic_locdfitp:**
  ic_locdfitp = 3 / Order of polynomial to fit for position error
  Used in: KW_DRS.py
  Defined in: hadmrICDP_SPIROU.py

- **ic_max_rms_center:**
  ic_max_rms_center = 0.2 / Maximum rms for sigma-clip order fit (center positions)
  Used in: cal_loc_RAW_spirou.py
  Defined in: hadmrICDP_SPIROU.py

- **ic_max_ptp_center:**
  ic_max_ptp_center = 0.200 / Maximum ptp for sigma-clip order fit (center positions)
  Used in: cal_loc_RAW_spirou.py
  Defined in: hadmrICDP_SPIROU.py

- **ic_ptporms_center:**
  ic_ptporms_center = 0.200 / Maximum frac ptp/rms for sigma-clip order fit
  Used in: cal_loc_RAW_spirou.py
  Defined in: hadmrICDP_SPIROU.py

- **ic_max_rms_width:**
  ic_max_rms_width = 1.0 / Maximum rms for sigma-clip order fit (width)
  Used in: cal_loc_RAW_spirou.py
  Defined in: hadmrICDP_SPIROU.py

- **ic_max_ptp_fracfwhm:**
  ic_max_ptp_fracfwhm = 10.0 / Maximum ptp for sigma-clip order fit (width)
  Used in: cal_loc_RAW_spirou.py
  Defined in: hadmrICDP_SPIROU.py

- **ic_loc_delta_width:**

  ic_loc_delta_width = 1.85 / Delta width (pix) for 3 convol shape model
  Used in: KW_DRS.py
  Defined in: hadmrICDP_SPIROU.py

- **ic_locopt1:**

  ic_locopt1 = 1 / Localisation option 1: Option for archiving the location image
  Used in: cal_loc_RAW_spirou.py
  Defined in: hadmrICDP_SPIROU.py

5.3.6 Tilt calculation parameters

- **ic_extopt:**

  ic_extopt = 0 / Extraction option in tilt calculation
  Used in: cal_SLIT_spirou.py
  Defined in: cal_SLIT_spirou.py, hadmrICDP_SPIROU.py

- **ic_extnbsig:**

  ic_extnbsig = 2.5 / Distance away from center to extract out to +/-
  Used in: cal_SLIT_spirou.py
  Defined in: cal_SLIT_spirou.py, hadmrICDP_SPIROU.py

- **ic_facdec:**

  ic_facdec = 1.6 / Offset multiplicative factor (for width)
  Used in: cal_SLIT_spirou.py
  Defined in: cal_SLIT_spirou.py

- **coi:**

  coi = 10 / Oversampling factor (for tilt finding)
  Used in: cal_SLIT_spirou.py
  Defined in: cal_SLIT_spirou.py

5.3.7 Flat-fielding parameters

- **ic_do_bkgr_subtraction:**

  ic_do_bkgr_subtraction = 0 / Do background measurement (True = 1, False = 0)
  Used in: cal_ff_RAW_spirou.py
  Defined in: hadmrICDP_SPIROU.py

- **ic_bkgr_window:**

  ic_bkgr_window = 100 / Half-size of window for background measurements
  Used in: cal_ff_RAW_spirou.py
  Defined in: hadmrICDP_SPIROU.py

- **nbo:**

  nbo = 36 / Number of orders in tilt file
  Used in: cal_ff_RAW_spirou.py
  Defined in: cal_ff_RAW_spirou.py

- **ccdsigdet:**

  ccdsigdet = 100 / Manually set the sigdet to use in weighted tilt extraction
  Used in: cal_ff_RAW_spirou.py, cal_extract_RAW_spirou{fiber}.py, cal_extract_RAW_spirouALL.py
  Defined in: cal_ff_RAW_spirou.py

- **ic_extfblaz:**

  ic_extfblaz = 50 / Half size blaze smoothing window
  Used in: cal_ff_RAW_spirou.py, cal_extract_RAW_spirou{fiber}.py, cal_extract_RAW_spirouALL.py
  Defined in: hadmrICDP_SPIROU.py

- **ic_order_plot:**

  ic_order_plot = 5 / Order to plot on ff image plot
  Used in: cal_ff_RAW_spirou.py, cal_extract_RAW_spirou{fiber}.py, cal_extract_RAW_spirouALL.py
  Defined in: cal_ff_RAW_spirou.py, hadmrICDP_SPIROU.py

5.3.8 Extraction parameters

- **ic_extmeanzone:**

  ic_extmeanzone = 16 / Mean window of extraction
  Used in: cal_extract_RAW_spirou{fiber}.py, cal_extract_RAW_spirouALL.py
  Defined in: hadmrICDP_SPIROU.py

- **satseuil:**

  satseuil = 64536 / Saturation threshold on order profile plot
  Used in: cal_extract_RAW_spirou{fiber}.py, cal_extract_RAW_spirouALL.py
  Defined in: cal_extract_RAW_spirou{fiber}.py, cal_extract_RAW_spirouALL.py

5.3.9 Drift parameters

- **noise:**

  noise = 100 / The value of the noise for drift calculation
  Used in: cal_DRIFT_RAW_spirou.py
  Defined in: cal_DRIFT_RAW_spirou.py

- **seuilmax:**
  seuilmax = 1.e9 / The maximum flux for good (unsaturated) pixels
  Used in: cal_DRIFT_RAW_spirou.py
  Defined in: cal_DRIFT_RAW_spirou.py

- **size:**
  size = 12 / The size around a saturated pixel to flag as unusable
  Used in: cal_DRIFT_RAW_spirou.py
  Defined in: spirouRV.py.deltavrms2d(), spirouRV.py.renorm_cosmics2d(), spirouRV.py.calcvrdrifts2d()

- **skip:**
  skip = 3 / Parameter to skip files in the sequence, only every "skip" file is used when loading 'fp_fp' files from {DATA_ROOT_RAW}.
  Used in: cal_DRIFT_RAW_spirou.py
  Defined in: cal_DRIFT_RAW_spirou.py

- **cut:**
  cut = 3 / Define the number of standard deviations cut at in cosmic renormalisation
  Used in: cal_DRIFT_RAW_spirou.py
  Defined in: cal_DRIFT_RAW_spirou.py, spirouRV.py.renorm_cosmics2d()

- **nomax:**
  nomax = 28 / Define the number of orders to use (starting from 0 to max) used to get median drift median(order0 –> order max)
  Used in: cal_DRIFT_RAW_spirou.py
  Defined in: cal_DRIFT_RAW_spirou.py

5.3.10 Quality Control Variables

- **qc_max_darklevel:**
  qc_max_darklevel = 1.0 / Max dark median level [ADU/s]
  Used in: cal_DARK_spirou.py
  Defined in: cal_DARK_spirou.py, hadmrICDP_SPIROU.py

- **qc_maxdead:**
  qc_maxdead = 25.0 / Max fraction of dead pixels
  Used in: cal_DARK_spirou.py
  Defined in: cal_DARK_spirou.py

- **qc_dark_time:**
  qc_dark_time = 1.0 / Min dark exposure time
  Used in: cal_DARK_spirou.py
  Defined in: hadmrICDP_SPIROU.py

- **qc_loc_maxlocfit_removed_ctr:**
  qc_loc_maxlocfit_removed_ctr = 1500 / Maximum points removed in location fit
  Used in: cal_loc_RAW_spirou.py
  Defined in: hadmrICDP_SPIROU.py

- **qc_loc_maxlocfit_removed_wid:**

  qc_loc_maxlocfit_removed_wid = 105 / Maximum points removed in width fit
  Used in: cal_loc_RAW_spirou.py
  Defined in: hadmrICDP_SPIROU.py

- **qc_loc_rmsmax_center:**

  qc_loc_rmsmax_center = 100 / Maximum rms allowed in fitting location
  Used in: cal_loc_RAW_spirou.py
  Defined in: hadmrICDP_SPIROU.py

- **qc_loc_rmsmax_fwhm :**

  qc_loc_rmsmax_fwhm = 500 / Maximum rms allowed in fitting width
  Used in: cal_loc_RAW_spirou.py
  Defined in: hadmrICDP_SPIROU.py

- **qc_max_signal:**

  qc_max_signal = 65500 / Maximum signal allowed (set saturation limit)
  Used in: cal_ff_RAW_spirou.py, cal_extract_RAW_spirou{fiber}.py, cal_extract_RAW_spirouALL.py
  Defined in: hadmrICDP_SPIROU.py

- **qc_loc_flumax:**

  qc_loc_flumax = 64500 / Saturation level reached warning
  Used in: cal_ff_RAW_spirou.py
  Defined in: hadmrICDP_SPIROU.py

5.3.11 Calib DB settings

- **ic_calib_db_master_file:**

  ic_calib_db_master_file = master_calib_SPIROU.txt / Define calibd DB master filename
  Used in: cal_DARK_spirou.py, cal_loc_RAW_spirou.py
  Defined in: hadmrICDP_SPIROU.py

5.4 Input Keywords

The following FITS descriptors of the 2D raw frames are required for the DRS. Last updated version 21 Nov 2014.

5.4.1 Standard FITS Keywords

BITPIX	=	16	/	16bit
NAXIS	=	2	/	Number of axes
NAXIS1	=	4096	/	Number of pixel columns
NAXIS2	=	4096	/	Number of pixel rows
BZERO	=	32768.0	/	Zero factor
BSCALE	=	1.0	/	Scale factor
DATE	=	'2013-11-26T09:06:14'	/	UTC Date of file creation
INSTRUME	=	'SPIROU'	/	Instrument Name

5.4.2 FITS keywords related to the detector

EXPTIME	=	800.0	/	Integration time (seconds)
DARKTIME	=	800.0	/	Dark current time (seconds)
GAIN	=	1.30	/	gain (electrons/ADU)
RDNOISE	=	4.20	/	read noise (electrons)
NSUBEXP	=	4	/	Total number of sub-exposures of 5.2s
OBSTYPE	=	'NORMAL'	/	Exposure type (DARK/NORMAL)
MIDEXPTM	=	400	/	mid-exposure time (seconds)
EMCNTS	=	444578	/	exposure meter counts at end

5.4.3 FITS keywords related to the target

OBJNAME	=	Gl9999	/	Target name
OBJRA	=	'5:35:09.87'	/	Target right ascension
OBJDEC	=	'-5:27:53.3'	/	Target declination
OBJRAPM	=	0.560	/	Target right ascension proper motion in as/yr
OBJDECPM	=	-0.33	/	Target declination proper motion in as/yr
OBJEQUIN	=	2000.0	/	Target equinox
OBJRV	=	-30.0	/	Target Radial velocity (km/s) (999 if unknown)
OBJTYPE	=	'M5'	/	Target spectral type
OBJJMAG	=	8.2	/	Target J magnitude
OBJHMAG	=	9.2	/	Target H magnitude
OBJKMAG	=	10.0	/	Target K magnitude

5.4.4 FITS keywords related to the telescope

DATE_OBS	=	'2013-11-26T09 :06 :14.858'	/	Date at start of observation (UTC)
EQUINOX	=	2000.0	/	Equinox of coordinates
EPOCH	=	2000.0	/	Epoch of coordinates
MJDATE	=	56622.3700212	/	Modified Julian Date at start of observation
MJEND	=	56622.3797593	/	Modified Julian Date at end of observation
AIRMASS	=	1.4	/	Airmass at start of observation
RA	=	'5:35:09.87'	/	Telescope right ascension
DEC	=	'-5:27:53.3'	/	Telescope declination
SEEING	=	1.0	/	Seeing at start of observation

5.4.5 FITS keywords related to the instrument

TPL_NAME	=	'SPIROU_POL_WAVE'	/	template Name
TPL_NEXP	=	1	/	# of exposure within template
TPL_EXPN	=	1	/	exposure # within template
INS_CAL	=	'WAVE'	/	Simultaneous calibration (WAVE/FP/NONE)
INS_LAMP	=	'UrAr'	/	Calibration lamp
INS_RHB1	=	90	/	SPIROU rhomb 1 position (deg)
INS_RHB2	=	180	/	SPIROU rhomb 2 position deg)

5.5 Output Keywords

Below are a list of keywords that are used in FITS rec header files. Most are defined in a few files, specific FITS rec files that these are used in are listed in []. These are split into catagories for ease of reading.

Note: This section is currently updated for 'cal_DARK_spirou.py','cal_loc_RAW_spirou.py', 'cal_SLIT_spirou.py', 'cal_ff_RAW_spirou.py' and 'cal_DRIFT_RAW_spirou.py' only.

5.5.1 General keywords

- **kw_version:**

  VERSION = 0 / DRS version

  Used in: cal_loc_RAW_spirou.py
  Defined in: KW_DRS.py

- **kw_root_drs_loc:**

  LO = 0 / Used in keywords below

  Used in: KW_DRS.py
  Defined in: KW_DRS.py

- **kw_root_drs_flat:**

  FF = 0 / Used in keywords below

Used in: KW_DRS.py
Defined in: KW_DRS.py

5.5.2 cal_dark keywords

- **kw_DARK_dead:**

  DADEAD = 0 / Fraction dead pixels [%]

  Used in: cal_DARK_spirou.py[Dark Frame]
  Defined in: cal_DARK_spirou.py

- **kw_DARK_MED:**

  DAMED = 0 / median dark level [ADU/s]

  Used in: cal_DARK_spirou.py[Dark Frame]
  Defined in: cal_DARK_spirou.py

- **kw_DARKBLUE_dead:**

  DABDEAD = 0 / Fraction dead pixels blue part [%]

  Used in: cal_DARK_spirou.py[Dark Frame]
  Defined in: cal_DARK_spirou.py

- **kw_DARKBLUE_med:**

  DABMED = 0 / median dark level blue part [ADU/s]

  Used in: cal_DARK_spirou.py[Dark Frame]
  Defined in: cal_DARK_spirou.py

- **kw_DARKRED_dead:**

  DARDEAD = 0 / Fraction dead pixels red part [%]

  Used in: cal_DARK_spirou.py[Dark Frame]
  Defined in: cal_DARK_spirou.py

- **kw_DARKRED_med:**

  DARMED = 0 / median dark level red part [ADU/s]

  Used in: cal_DARK_spirou.py[Dark Frame]
  Defined in: cal_DARK_spirou.py

- **kw_DARK_cut:**

  DACUT = 0 / Threshold of dark level retain [ADU/s]

  Used in: cal_DARK_spirou.py[Dark Frame, Bad Pixel Map]
  Defined in: cal_DARK_spirou.py

5.5.3 cal_loc keywords

- **kw_LOCO_NBO:**

  {kw_root_drs_loc}NBO = 0 / nb orders localise

  Used in: cal_loc_RAW_spirou.py[*_LOCO_{fiber}*, *_fwhm-order_{fiber}*]
  Defined in: KW_DRS.py

- **kw_LOCO_BCKGRD:**

  {kw_root_drs_loc}BCKGRD = 0 / mean background [%]

  Used in: cal_loc_RAW_spirou.py[*_LOCO_{fiber}*, *_fwhm-order_{fiber}*]
  Defined in: KW_DRS.py

- **kw_LOCO_DEG_C:**

  {kw_root_drs_loc}DEGCTR = 0 / degree fit ctr ord

  Used in: cal_loc_RAW_spirou.py[*_LOCO_{fiber}*, *_fwhm-order_{fiber}*]
  Defined in: KW_DRS.py

- **kw_LOCO_DEGW:**

  {kw_root_drs_loc}DEGFWH = 0 / degree fit width ord

  Used in: cal_loc_RAW_spirou.py[*_LOCO_{fiber}*, *_fwhm-order_{fiber}*]
  Defined in: KW_DRS.py

- **kw_LOCO_DEGE:**

  {kw_root_drs_loc}DEGERR = 0 / degree fit profile error

  Used in: cal_loc_RAW_spirou.py[*_LOCO_{fiber}*, *_fwhm-order_{fiber}*] but not used
  Defined in: KW_DRS.py

- **kw_LOCO_DELTA:**

  {kw_root_drs_loc}PRODEL = 0 / param model 3gau

  Used in: cal_loc_RAW_spirou.py[*_LOCO_{fiber}*, *_fwhm-order_{fiber}*] but not used
  Defined in: KW_DRS.py

- **kw_LOCO_CTR_COEFF:**

  {kw_root_drs_loc}CTR[row, col] = 0 / Coeff center order

  Used in: cal_loc_RAW_spirou.py[*_LOCO_{fiber}*, *_fwhm-order_{fiber}*], used on a 2D list of coefficients size=(number of orders x number of fit coefficients)
  Defined in: KW_DRS.py

- **kw_LOCO_FWHM_COEFF):**

  {kw_root_drs_loc}FW[row, col] = 0 / Coeff fwhm order

  Used in: cal_loc_RAW_spirou.py[*_LOCO_{fiber}*, *_fwhm-order_{fiber}*], used on a 2D list of coefficients size=(number of orders x number of fit coefficients)
  Defined in: KW_DRS.py

- **kw_CCD_SIGDET:**

  SIGDET = 0 / CCD Readout Noise [e-]

  Used in: cal_loc_RAW_spirou.py[*_LOCO_{fiber}*, *_fwhm-order_{fiber}*]
  Defined in: KW_DRS.py

- **kw_CCD_CONAD:**

  CONAD = 0 / CCD conv factor [e-/ADU]

  Used in: cal_loc_RAW_spirou.py[*_LOCO_{fiber}*, *_fwhm-order_{fiber}*]
  Defined in: KW_DRS.py

- **kw_LOC_maxflx:**

  {kw_root_drs_loc}FLXMAX = 0 / max flux in order [ADU]

  Used in: cal_loc_RAW_spirou.py[*_LOCO_{fiber}*, *_fwhm-order_{fiber}*]
  Defined in: KW_DRS.py

- **kw_LOC_Smaxpts_ctr:**

  {kw_root_drs_loc}CTRMAX = 0 / max rm pts ctr

  Used in: cal_loc_RAW_spirou.py[*_LOCO_{fiber}*, *_fwhm-order_{fiber}*]
  Defined in: KW_DRS.py

- **kw_LOC_Smaxpts_width:**

  {kw_root_drs_loc}WIDMAX = 0 / max rm pts width

  Used in: cal_loc_RAW_spirou.py[*_LOCO_{fiber}*, *_fwhm-order_{fiber}*]
  Defined in: KW_DRS.py

- **kw_LOC_rms_ctr:**

  {kw_root_drs_loc}RMSCTR = 0 / max rms ctr

  Used in: cal_loc_RAW_spirou.py[*_LOCO_{fiber}*, *_fwhm-order_{fiber}*]
  Defined in: KW_DRS.py

- **kw_LOC_rms_fwhm:**

  {kw_root_drs_loc}RMSWID = 0 / max rms width

  Used in: cal_loc_RAW_spirou.py[*_LOCO_{fiber}*, *_fwhm-order_{fiber}*]
  Defined in: KW_DRS.py

- **kw_drs_QC):**

  QC = 0 / QCcontr

  Used in: cal_loc_RAW_spirou.py[*_LOCO_{fiber}*, *_fwhm-order_{fiber}*]
  Defined in: KW_DRS.py

5.5.4 cal_slit keywords

- **kw_TILT:**

  {kw_root_drs_loc}TILT = 0 / Tilt order

  Used in: cal_SLIT_spirou.py[*_tilt.fits]
  Defined in: KW_DRS.py

5.5.5 cal_ff keywords

- **kw_EXTRA_SN:**

  EXTSN = 0 / Signal to noise traio for order center

  Used in: cal_ff_RAW_spirou.py[*_blaze_{fiber}.fits, *_flat_{fiber}.fits]
  Defined in: KW_DRS.py

- **kw_FLAT_RMS:**

  {kw_root_drs_flat}RMS = 0 / FF RMS order

  Used in: cal_ff_RAW_spirou.py[*_flat_{fiber}.fits]
  Defined in: KW_DRS.py

5.5.6 cal_extract keywords

- **kw_LOCO_FILE:**

  {kw_root_drs_loc}FILE = 0 / Localization file used in extraction process

  Used in: cal_extract_RAW_spirou{fiber}.py, cal_extract_RAW_spirouALL.py[*_e2ds_{fiber}.fits
  Defined in: KW_DRS.py

Note all cal_loc keywords are also copied to "cal_extract_RAW_spirou{fiber}.py, cal_extract_RAW_spirouALL.py[*_
files

Chapter 6

The Recipes

6.1 The Recipe directory

These are symbolically linked under {INSTALL_ROOT}/bin. A list of all current recipes is below:

```
{DIR}
└── {INSTALL_ROOT}
    └── DRS_SPIROU
        └── python
            └── Recipes
                ├── cal_BIAS_spirou.py
                ├── cal_CONTAM_spirou.py
                ├── cal_DARK_spirou.py
                ├── cal_DRIFT_E2DS_spirou.py
                ├── cal_DRIFT_RAW_spirou.py
                ├── cal_extract_RAW_spirouAB.py
                ├── cal_extract_RAW_spirouALL.py
                ├── cal_extract_RAW_spirouC.py
                ├── cal_FFPOL_spirou.py
                ├── cal_FF_RAW_spirou.py
                ├── cal_FF_spirou.py
                ├── cal_loc_ONE_spirou.py
                ├── cal_loc_RAW_spirou.py
                ├── cal_SLIT_spirou.py
                ├── cal_TH_spirou.py
                ├── cal_WAVE_spirou.py
                ├── db_get_files_spirou.py
                ├── db_reduce_star_spirou.py
                ├── db_update_spirou.py
                ├── Install.csh
                ├── mai_cal_drift_spirou.py
                ├── mai_compute_drift_spirou.py
                ├── mai_config_wavecal_spirou.py
                ├── mai_make_flux_template_spirou.py
                ├── mai_plot_drift_spirou.py
                ├── obj_ONE_spirou.py
                ├── obj_TH_spirou.py
                ├── obj_TWO_spirou.py
                ├── obj_WAVE_spirou.py
                ├── off_make_bis_spirou.py
                └── off_make_ccf_spirou.py
```

```
├── off_make_execCAL_spirou.py
├── off_make_execOBJ_spirou.py
├── off_make_exec_spirou.py
├── off_make_S_spirou.py
├── off_visu_bis_spirou.py
├── off_visu_ccf_spirou.py
├── off_visu_dark_spirou.py
├── off_visu_e2ds_spirou.py
├── off_visu_rvo_spirou.py
├── off_visu_s1d_spirou.py
├── off_visu_SN_spirou.py
├── ske_recipe_spirou.py
├── test_cal_loc_ONE_spirou.py
└── visu_RAW_spirou.py
```

6.2 The cal_DARK_spirou recipe

Dark with short exposure time (5min, to be defined during AT-4) to check if read-out noise, dark current and hot pixel mask are consistent with the ones obtained during technical night. Quality control is done automaticaly by the pipeline (see Section 8.1).

File prefixes allowed:

- dark_dark

6.2.1 Summary of procedure

1. adds defined 'dark_dark' files together
2. resizes the image
3. calculates the fraction of dead pixels [full, blue part, red part]
4. calculates median dark level [full, blue part, red part]
5. calculates threshold of dark level to retain
6. removes dead pixels by setting them to 0
7. does some quality control
8. updates calibDB with key "DARK"

6.2.2 Running cal_DARK_spirou

To run cal_DARK_spirou type:

```
cal_DARK_spirou.py  night_repository  filenames
```

Note: filenames must start with 'dark_dark'

6.2.3 Example working run

An example run where everything worked is below:

```
cal_DARK_spirou.py 20170710 dark_dark02d406.fits


20:44:08.3 -   ||DRS  SPIROU   v   (interactive mode)
20:44:08.3 -   || ****************************************
20:44:08.3 -   || * SPIROU @(#) Geneva Observatory ()
20:44:08.3 -   || ****************************************
20:44:08.3 -   ||(dir_data_raw)      DRS_DATA_RAW=/scratch/Projects/SPIRou_Pipeline/data/raw/
20:44:08.3 -   ||(dir_data_reduc)    DRS_DATA_REDUC=/scratch/Projects/SPIRou_Pipeline/data/reduced/
20:44:08.3 -   ||(dir_drs_config)    DRS_CONFIG=/scratch/Projects/SPIRou_Pipeline/INTROOT/DRS_SPIROU/config/
20:44:08.3 -   ||(dir_calib_db)      DRS_CALIB_DB=/scratch/Projects/SPIRou_Pipeline/data/calibDB
20:44:08.3 -   ||(dir_data_msg)      DRS_DATA_MSG=/scratch/Projects/SPIRou_Pipeline/data/msg/
20:44:08.3 -   ||(print_log)         DRS_LOG=1         %(0: minimum stdin-out logs)
20:44:08.3 -   ||(plot_graph)        DRS_PLOT=NONE           %(def/undef/trigger)
20:44:08.3 -   ||(used_date)         DRS_USED_DATE=undefined
20:44:08.3 -   ||(working_dir)       DRS_DATA_WORKING=/scratch/Projects/SPIRou_Pipeline/data/tmp/
20:44:08.3 -   ||                    DRS_INTERACTIVE is  set
20:44:08.3 -   |-c:|Now running : -c on file(s):  dark_dark02d406.fits
20:44:08.3 -   |-c:|On directory /scratch/Projects/SPIRou_Pipeline/data/raw/20170710
20:44:08.3 -   |-c:|ICDP loaded from: /scratch/Projects/SPIRou_Pipeline/INTROOT/DRS_SPIROU/config/
     hadmrICDP_SPIROU.py
20:44:08.3 - * |-c:|Now processing Image TYPE DARK with -c recipe
20:44:08.3 -   |-c:|Reading Image /scratch/Projects/SPIRou_Pipeline/data/raw/20170710/dark_dark02d406.fits
20:44:08.4 -   |-c:|Image 2048x2048 loaded
```

```
20:44:08.6 - * |-c:|Dark Time = 597.489 [s]
20:44:08.8 -   |-c:|Doing Dark measurement
20:44:10.1 - * |-c:|Whole det   : Frac dead pixels= 14.7 % - Median= 0.35 ADU/s - Percent[5:95]= 0.08-99.57 ADU/
     s
20:44:10.2 - * |-c:|In Blue part: Frac dead pixels= 1.0 % - Median= 0.15 ADU/s - Percent[5:95]= 0.09-0.53 ADU/s
20:44:10.3 - * |-c:|In Red part : Frac dead pixels= 20.5 % - Median= 2.11 ADU/s - Percent[5:95]= 0.18-232.09 ADU
     /s
20:44:10.4 - * |-c:|Total Frac dead pixels (N.A.N) + DARK > 100.0 ADU/s = 18.9 %
20:44:11.1 - * |-c:|QUALITY CONTROL SUCCESSFUL - Well Done -
20:44:11.1 -   |-c:|Saving Dark frame in dark_dark02d406.fits
20:44:12.2 -   |-c:|Saving Bad Pixel Map in dark_dark02d406_badpixel.fits
20:44:13.0 - * |-c:|Updating Calib Data Base with DARK
20:44:13.0 - * |-c:|Recipe -c has been succesfully completed
```

6.2.4 Interactive mode

In interactive mode three figures will also appear (eee figures 6.1, 6.2, and 6.3).

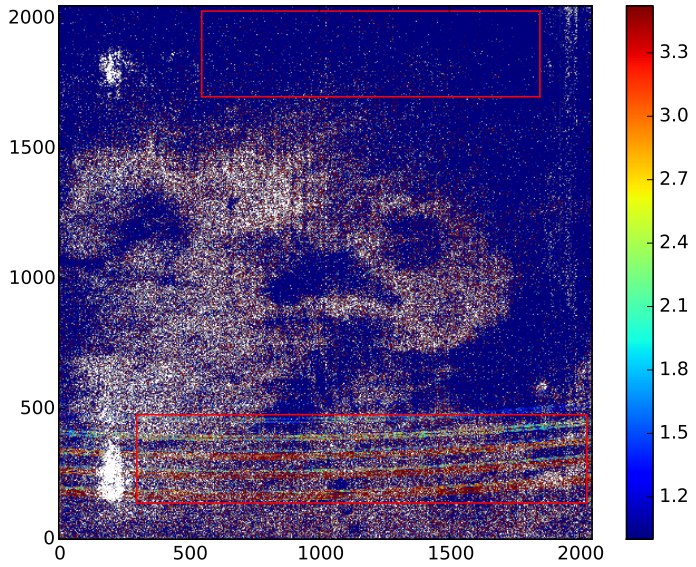

Figure 6.1 The image with overplot red and blue regions (red rectangles).

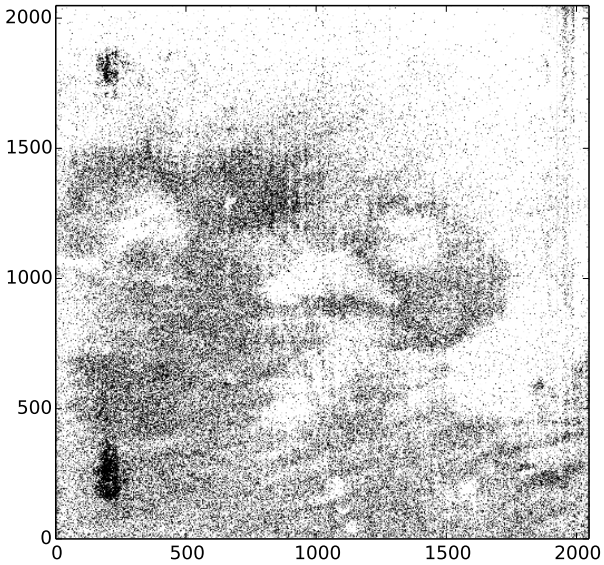

Figure 6.2 The bad pixel mask, bad pixels have a value=1 (in black) and good pixels have a value=0 (in white).

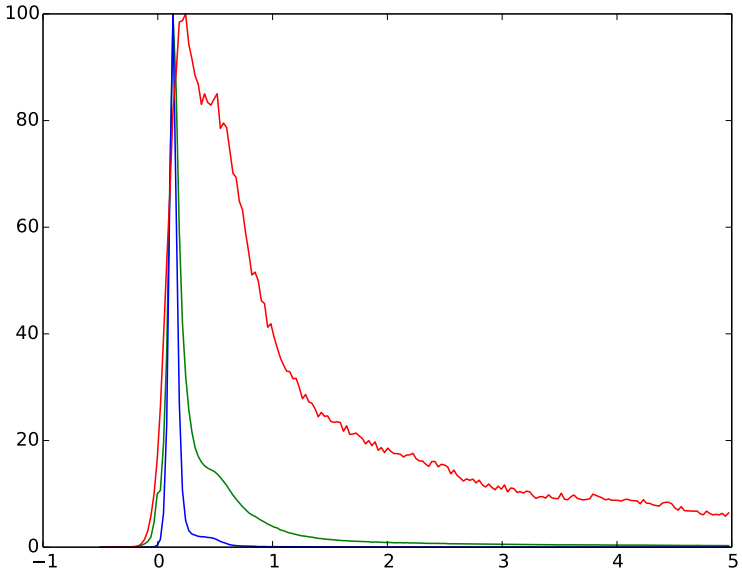

Figure 6.3 Histograms of the image regions, the full image (in green), the blue section (in blue) and the red section (in red).

6.3 The cal_loc_RAW_spirou recipe

Locates the orders on the 'dark_flat' or 'flat_dark' images.

File prefixes allowed:

- dark_flat
- flat_dark

6.3.1 Summary of procedure

1. adds all defined 'dark_flat' or 'flat_dark' files together
2. corrects for darks
3. resizes the image
4. constructs 'order_profile' image
5. locates the central pixel of each order
6. steps out in large steps along the order (toward beginning and end)
7. fits the position of each order (using a small 2D box around each fit point)
    - includes a rejection of bad points (while loop)
8. fits the width of each order (using a small 2D box around each fit point)
    - includes a rejection of bad points (while loop)
9. saves the 'order_profile' image (with a superposition of the fit orders as zero values)
10. does some quality control
11. updates calibDB with key "LOC_{fiber}" where {fiber} = [AB, C] etc

6.3.2 Running cal_loc_RAW_spirou

To run cal_loc_RAW_spirou type:

```
cal_loc_RAW_spirou.py  night_repository  filenames
```

Note: filenames must start with 'dark_flat' or 'flat_dark'

6.3.3 Example working run

An example run where everything worked is below (for 'dark_flat' files):

```
cal_loc_RAW_spirou.py 20170710 dark_flat02f10.fits dark_flat03f10.fits

||DRS  SPIROU   v   (interactive mode)
16:15:58.5 -   || ****************************************
16:15:58.5 -   || * SPIROU @(#) Geneva Observatory ()
16:15:58.5 -   || ****************************************
16:15:58.5 -   ||(dir_data_raw)      DRS_DATA_RAW=/scratch/Projects/SPIRou_Pipeline/data/raw/
16:15:58.5 -   ||(dir_data_reduc)    DRS_DATA_REDUC=/scratch/Projects/SPIRou_Pipeline/data/reduced/
16:15:58.5 -   ||(dir_drs_config)    DRS_CONFIG=/scratch/Projects/SPIRou_Pipeline/INTROOT/DRS_SPIROU/config/
16:15:58.5 -   ||(dir_calib_db)      DRS_CALIB_DB=/scratch/Projects/SPIRou_Pipeline/data/calibDB
16:15:58.5 -   ||(dir_data_msg)      DRS_DATA_MSG=/scratch/Projects/SPIRou_Pipeline/data/msg/
16:15:58.5 -   ||(print_log)         DRS_LOG=1         %(0: minimum stdin-out logs)
```

```
16:15:58.5 -   ||(plot_graph)        DRS_PLOT=NONE            %(def/undef/trigger)
16:15:58.5 -   ||(used_date)         DRS_USED_DATE=undefined
16:15:58.5 -   ||(working_dir)       DRS_DATA_WORKING=/scratch/Projects/SPIRou_Pipeline/data/tmp/
16:15:58.5 -   ||                    DRS_INTERACTIVE is  set
16:15:58.5 -   |-c:+[...]|Now running : -c on file(s):  dark_flat02f10.fits dark_flat03f10.fits
16:15:58.5 -   |-c:+[...]|On directory /scratch/Projects/SPIRou_Pipeline/data/raw/20170710
16:15:58.5 -   |-c:+[...]|ICDP loaded from: /scratch/Projects/SPIRou_Pipeline/INTROOT/DRS_SPIROU/config/
     hadmrICDP_SPIROU.py
16:15:58.5 -   |-c:+[...]|Reading file: /scratch/Projects/SPIRou_Pipeline/data/raw/20170710/dark_flat02f10.fits
16:15:58.5 -   |-c:+[...]|Image 2048x2048 loaded
16:15:58.6 - * |-c:+[...]|Adding frames
16:15:58.6 -   |-c:+[...]|Reading File: /scratch/Projects/SPIRou_Pipeline/data/raw/20170710/dark_flat03f10.fits
16:15:59.0 -   |-c:+[...]|Doing Dark Correction using /scratch/Projects/SPIRou_Pipeline/data/calibDB/
     dark_dark02d406.fits
16:15:59.4 -   |-c:+[...]|Image format changed to 2035x1930
16:16:01.4 -   |-c:+[...]|Saving processed raw frame in dark_flat02f10_order_profil_C.fits
16:16:01.8 - * |-c:+[...]|Updating Calib Data Base with ORDER_PROFIL_C
16:16:01.9 - * |-c:+[...]|Maximum flux/pixel in the spectrum: 220526.1[e-]
16:16:01.9 - * |-c:+[...]|Average background level: 0.46[%]
16:16:02.3 -   |-c:+[...]|Searching order center on central column
16:16:02.3 - * |-c:+[...]|On fiber C 36 orders have been detected on 1 fiber
16:16:02.4 -   |-c:+[...]|ORDER: 0 center at pixel 88.0 width 11.8 rms 0.058
16:16:02.4 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.058/0.175/3.0 with 0 rejected points
16:16:02.4 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.393/0.889/8.081 with 0 rejected points
16:16:02.4 -   |-c:+[...]|ORDER: 1 center at pixel 135.3 width 11.7 rms 0.067
16:16:02.4 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.067/0.196/2.9 with 0 rejected points
16:16:02.4 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.393/0.908/8.256 with 0 rejected points
16:16:02.4 -   |-c:+[...]|ORDER: 2 center at pixel 182.1 width 11.9 rms 0.057
16:16:02.4 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.057/0.138/2.4 with 0 rejected points
16:16:02.4 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.376/0.939/8.156 with 0 rejected points
16:16:02.4 -   |-c:+[...]|ORDER: 3 center at pixel 228.5 width 11.9 rms 0.052
16:16:02.4 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.052/0.179/3.5 with 0 rejected points
16:16:02.4 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.386/0.921/8.376 with 0 rejected points
16:16:02.4 -   |-c:+[...]|ORDER: 4 center at pixel 274.3 width 11.7 rms 0.058
16:16:02.4 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.058/0.189/3.2 with 0 rejected points
16:16:02.4 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.385/0.908/8.256 with 0 rejected points
16:16:02.4 -   |-c:+[...]|ORDER: 5 center at pixel 319.8 width 11.7 rms 0.054
16:16:02.4 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.054/0.140/2.6 with 0 rejected points
16:16:02.4 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.409/0.796/7.237 with 0 rejected points
16:16:02.4 -   |-c:+[...]|ORDER: 6 center at pixel 364.8 width 11.7 rms 0.057
16:16:02.4 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.057/0.149/2.6 with 0 rejected points
16:16:02.4 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.357/0.926/7.595 with 0 rejected points
16:16:02.4 -   |-c:+[...]|ORDER: 7 center at pixel 409.6 width 11.6 rms 0.054
16:16:02.4 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.054/0.146/2.7 with 0 rejected points
16:16:02.4 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.414/0.882/8.015 with 0 rejected points
16:16:02.4 -   |-c:+[...]|ORDER: 8 center at pixel 454.1 width 11.5 rms 0.059
16:16:02.4 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.059/0.151/2.5 with 0 rejected points
16:16:02.4 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.434/0.805/7.319 with 0 rejected points
16:16:02.4 -   |-c:+[...]|ORDER: 9 center at pixel 498.2 width 11.6 rms 0.058
16:16:02.4 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.058/0.180/3.1 with 0 rejected points
16:16:02.4 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.429/0.833/7.575 with 0 rejected points
16:16:02.4 -   |-c:+[...]|ORDER: 10 center at pixel 542.2 width 11.3 rms 0.066
16:16:02.4 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.066/0.162/2.5 with 0 rejected points
16:16:02.4 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.437/0.886/7.761 with 0 rejected points
16:16:02.4 -   |-c:+[...]|ORDER: 11 center at pixel 586.0 width 11.4 rms 0.071
16:16:02.4 -   |-c:+[...]|      center fit converging with rms/ptp/sigrms: 0.071/0.210/3.0
16:16:02.4 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.068/0.150/2.2 with 1 rejected points
16:16:02.4 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.453/0.834/7.579 with 0 rejected points
16:16:02.5 -   |-c:+[...]|ORDER: 12 center at pixel 629.7 width 11.5 rms 0.065
16:16:02.5 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.065/0.159/2.4 with 0 rejected points
16:16:02.5 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.459/0.804/6.727 with 0 rejected points
16:16:02.5 -   |-c:+[...]|ORDER: 13 center at pixel 673.3 width 11.3 rms 0.065
16:16:02.5 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.065/0.177/2.7 with 0 rejected points
16:16:02.5 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.420/0.861/7.826 with 0 rejected points
16:16:02.5 -   |-c:+[...]|ORDER: 14 center at pixel 717.0 width 11.5 rms 0.065
16:16:02.5 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.065/0.192/3.0 with 0 rejected points
16:16:02.5 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.469/0.840/7.638 with 0 rejected points
16:16:02.5 -   |-c:+[...]|ORDER: 15 center at pixel 760.7 width 11.2 rms 0.069
16:16:02.5 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.069/0.185/2.7 with 0 rejected points
16:16:02.5 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.442/0.858/7.802 with 0 rejected points
16:16:02.5 -   |-c:+[...]|ORDER: 16 center at pixel 804.6 width 11.3 rms 0.056
16:16:02.5 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.056/0.158/2.8 with 0 rejected points
16:16:02.5 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.451/0.871/7.917 with 0 rejected points
16:16:02.5 -   |-c:+[...]|ORDER: 17 center at pixel 848.7 width 11.3 rms 0.062
16:16:02.5 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.062/0.185/3.0 with 0 rejected points
16:16:02.5 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.447/0.931/7.756 with 0 rejected points
```

```
16:16:02.5 -   |-c:+[...]|ORDER: 18 center at pixel 893.2 width 11.2 rms 0.067
16:16:02.5 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.067/0.174/2.6 with 0 rejected points
16:16:02.5 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.392/0.915/7.623 with 0 rejected points
16:16:02.5 -   |-c:+[...]|ORDER: 19 center at pixel 938.0 width 11.2 rms 0.063
16:16:02.5 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.063/0.189/3.0 with 0 rejected points
16:16:02.5 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.437/0.841/7.005 with 0 rejected points
16:16:02.5 -   |-c:+[...]|ORDER: 20 center at pixel 983.4 width 11.4 rms 0.076
16:16:02.5 -   |-c:+[...]|      center fit converging with rms/ptp/sigrms: 0.076/0.215/2.8
16:16:02.5 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.073/0.171/2.3 with 1 rejected points
16:16:02.5 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.423/0.954/7.949 with 0 rejected points
16:16:02.5 -   |-c:+[...]|ORDER: 21 center at pixel 1029.4 width 11.3 rms 0.075
16:16:02.5 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.075/0.193/2.6 with 0 rejected points
16:16:02.5 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.414/0.893/7.446 with 0 rejected points
16:16:02.5 -   |-c:+[...]|ORDER: 22 center at pixel 1076.2 width 11.0 rms 0.076
16:16:02.5 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.076/0.191/2.5 with 0 rejected points
16:16:02.5 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.372/0.951/7.927 with 0 rejected points
16:16:02.5 -   |-c:+[...]|ORDER: 23 center at pixel 1123.9 width 11.0 rms 0.076
16:16:02.5 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.076/0.171/2.3 with 0 rejected points
16:16:02.5 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.332/0.965/8.041 with 0 rejected points
16:16:02.5 -   |-c:+[...]|ORDER: 24 center at pixel 1172.7 width 11.0 rms 0.082
16:16:02.5 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.082/0.197/2.4 with 0 rejected points
16:16:02.5 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.322/0.931/9.309 with 0 rejected points
16:16:02.6 -   |-c:+[...]|ORDER: 25 center at pixel 1222.7 width 11.0 rms 0.083
16:16:02.6 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.083/0.170/2.1 with 0 rejected points
16:16:02.6 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.320/0.938/9.376 with 0 rejected points
16:16:02.6 -   |-c:+[...]|ORDER: 26 center at pixel 1274.2 width 11.0 rms 0.066
16:16:02.6 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.066/0.159/2.4 with 0 rejected points
16:16:02.6 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.331/0.982/9.815 with 0 rejected points
16:16:02.6 -   |-c:+[...]|ORDER: 27 center at pixel 1327.4 width 10.7 rms 0.075
16:16:02.6 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.075/0.185/2.5 with 0 rejected points
16:16:02.6 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.405/0.958/9.578 with 0 rejected points
16:16:02.6 -   |-c:+[...]|ORDER: 28 center at pixel 1382.6 width 10.9 rms 0.067
16:16:02.6 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.067/0.169/2.5 with 0 rejected points
16:16:02.6 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.297/0.876/8.760 with 0 rejected points
16:16:02.6 -   |-c:+[...]|ORDER: 29 center at pixel 1440.0 width 10.8 rms 0.062
16:16:02.6 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.062/0.180/2.9 with 0 rejected points
16:16:02.6 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.364/1.051/8.824 with 0 rejected points
16:16:02.6 -   |-c:+[...]|ORDER: 30 center at pixel 1500.1 width 10.7 rms 0.060
16:16:02.6 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.060/0.184/3.1 with 0 rejected points
16:16:02.6 -   |-c:+[...]|      width fit converging with rms/ptp/ptp%: 0.362/1.006/10.064
16:16:02.6 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.348/0.917/8.160 with 1 rejected points
16:16:02.6 -   |-c:+[...]|ORDER: 31 center at pixel 1563.2 width 10.6 rms 0.064
16:16:02.6 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.064/0.200/3.1 with 0 rejected points
16:16:02.6 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.407/0.799/7.990 with 0 rejected points
16:16:02.6 -   |-c:+[...]|ORDER: 32 center at pixel 1629.8 width 10.6 rms 0.066
16:16:02.6 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.066/0.176/2.7 with 0 rejected points
16:16:02.6 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.406/0.993/9.930 with 0 rejected points
16:16:02.6 -   |-c:+[...]|ORDER: 33 center at pixel 1700.3 width 10.6 rms 0.079
16:16:02.6 -   |-c:+[...]|      center fit converging with rms/ptp/sigrms: 0.079/0.233/3.0
16:16:02.6 -   |-c:+[...]|      center fit converging with rms/ptp/sigrms: 0.075/0.229/3.0
16:16:02.6 -   |-c:+[...]|      center fit converging with rms/ptp/sigrms: 0.072/0.205/2.9
16:16:02.6 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.069/0.193/2.8 with 3 rejected points
16:16:02.6 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.438/0.943/9.425 with 0 rejected points
16:16:02.6 -   |-c:+[...]|ORDER: 34 center at pixel 1775.7 width 10.4 rms 0.194
16:16:02.6 -   |-c:+[...]|      center fit converging with rms/ptp/sigrms: 0.194/1.728/8.9
16:16:02.6 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.067/0.167/2.5 with 1 rejected points
16:16:02.6 -   |-c:+[...]|      width fit converging with rms/ptp/ptp%: 0.566/3.326/47.516
16:16:02.6 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.447/0.921/7.754 with 1 rejected points
16:16:02.6 -   |-c:+[...]|ORDER: 35 center at pixel 1856.3 width 10.5 rms 0.297
16:16:02.6 -   |-c:+[...]|      center fit converging with rms/ptp/sigrms: 0.297/2.351/7.9
16:16:02.6 -   |-c:+[...]|      center fit converging with rms/ptp/sigrms: 0.168/1.475/8.8
16:16:02.6 -   |-c:+[...]|      center fit converging with rms/ptp/sigrms: 0.065/0.212/3.3
16:16:02.6 -   |-c:+[...]| - center fit rms/ptp/sigrms: 0.061/0.162/2.7 with 3 rejected points
16:16:02.6 -   |-c:+[...]|      width fit converging with rms/ptp/ptp%: 0.606/3.294/47.051
16:16:02.6 -   |-c:+[...]|      width fit converging with rms/ptp/ptp%: 0.501/2.459/30.733
16:16:02.6 -   |-c:+[...]| - width  fit rms/ptp/ptp%: 0.431/0.944/9.444 with 2 rejected points
16:16:02.6 - * |-c:+[...]|On fiber C 36 orders geometry have been measured
16:16:02.6 - * |-c:+[...]|Average uncertainty on position : 65.01 [mpix]
16:16:02.6 - * |-c:+[...]|Average uncertainty on width : 401.10 [mpix]
16:16:02.6 -   |-c:+[...]|Saving localization information in file: dark_flat02f10_loco_C.fits
16:16:03.3 -   |-c:+[...]|Saving FWHM information in dark_flat02f10_fwhm-order_C.fits
16:16:03.9 -   |-c:+[...]|Saving localization image with superposition of orders in file: dark_flat02f10_with-
     order_C.fits
16:16:04.2 - * |-c:+[...]|QUALITY CONTROL SUCCESSFUL - Well Done -
16:16:04.7 - * |-c:+[...]|Updating Calib Data Base with LOC_C
16:16:04.7 - * |-c:+[...]|Recipe -c has been successfully completed
```

6.3.4 Interactive mode

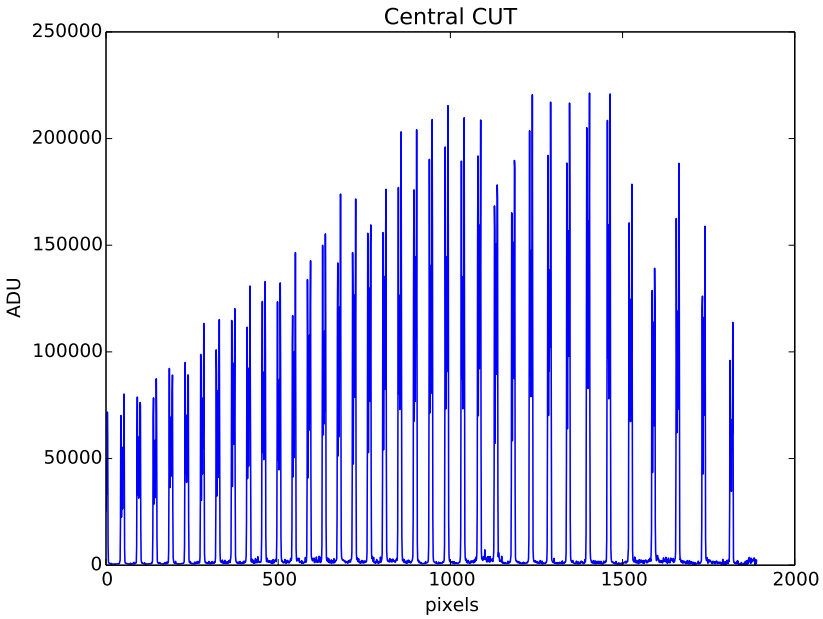

Figure 6.4 Pixel number (across order) against flux value of central pixel.

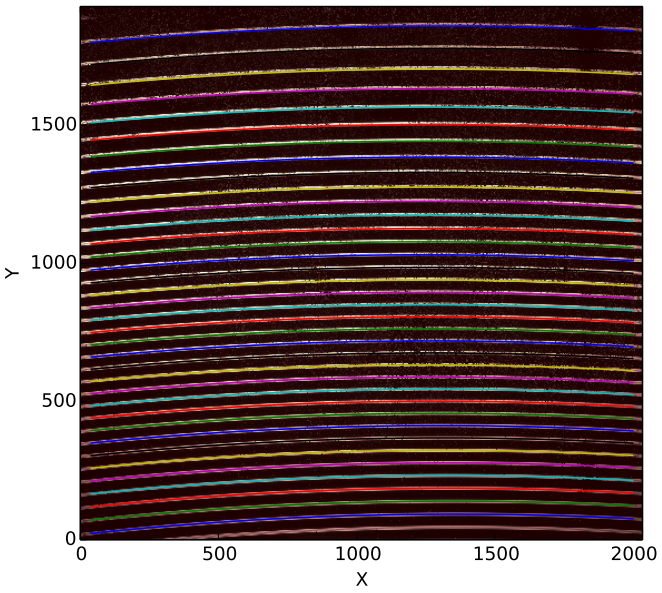

Figure 6.5 Image with fits to each order.

6.4 The cal_SLIT_spirou recipe

Fabry-Perot exposures in which the three fibres are simultaneously fed by light from the Fabry-Perot filter. Each exposure is used to build the slit orientation. Finds the tilt of the orders.

File prefixes allowed:

- fp_fp

6.4.1 Summary of procedure

1. adds all fp_fp files together
2. corrects for dark
3. resizes the image
4. extracts the orders (no weight no tilt)
5. works out the tilt for each order using the location and width
6. saves the tilt to file
7. should do some quality control
8. updates calibDB with key "TILT"

6.4.2 Running cal_

To run cal_ type:

```
cal_SLIT_spirou.p  night_repository  filenames
```

Note: filesnames must start with 'fp_fp'

6.4.3 Example working run

An example run where everything worked is below:

```
cal_SLIT_spirou.py 20170710 fp_fp02a203.fits fp_fp03a203.fits fp_fp04a203.fits

||DRS  SPIROU   v   (interactive mode)
16:26:55.3 -   || ****************************************
16:26:55.3 -   || * SPIROU @(#) Geneva Observatory ()
16:26:55.3 -   || ****************************************
16:26:55.3 -   ||(dir_data_raw)      DRS_DATA_RAW=/scratch/Projects/SPIRou_Pipeline/data/raw/
16:26:55.3 -   ||(dir_data_reduc)    DRS_DATA_REDUC=/scratch/Projects/SPIRou_Pipeline/data/reduced/
16:26:55.3 -   ||(dir_drs_config)    DRS_CONFIG=/scratch/Projects/SPIRou_Pipeline/INTROOT/DRS_SPIROU/config/
16:26:55.3 -   ||(dir_calib_db)      DRS_CALIB_DB=/scratch/Projects/SPIRou_Pipeline/data/calibDB
16:26:55.3 -   ||(dir_data_msg)      DRS_DATA_MSG=/scratch/Projects/SPIRou_Pipeline/data/msg/
16:26:55.3 -   ||(print_log)         DRS_LOG=1         %(0: minimum stdin-out logs)
16:26:55.3 -   ||(plot_graph)        DRS_PLOT=NONE            %(def/undef/trigger)
16:26:55.3 -   ||(used_date)         DRS_USED_DATE=undefined
16:26:55.3 -   ||(working_dir)       DRS_DATA_WORKING=/scratch/Projects/SPIRou_Pipeline/data/tmp/
16:26:55.3 -   ||                    DRS_INTERACTIVE is  set
16:26:55.3 -   |-c:+[...]|Now running : -c on file(s):  fp_fp02a203.fits fp_fp03a203.fits fp_fp04a203.fits
16:26:55.3 -   |-c:+[...]|On directory /scratch/Projects/SPIRou_Pipeline/data/raw/20170710
16:26:55.3 -   |-c:+[...]|ICDP loaded from: /scratch/Projects/SPIRou_Pipeline/INTROOT/DRS_SPIROU/config/
     hadmrICDP_SPIROU.py
16:26:55.4 -   |-c:+[...]|Calibration file: fp_fp02a203_tilt.fits already exists - not copied
16:26:55.4 -   |-c:+[...]|Calibration file: dark_flat02f10_loco_C.fits already exists - not copied
16:26:55.4 -   |-c:+[...]|Calibration file: dark_flat02f10_order_profil_C.fits already exists - not copied
16:26:55.5 -   |-c:+[...]|Calibration file: flat_dark02f10_order_profil_AB.fits already exists - not copied
16:26:55.5 -   |-c:+[...]|Calibration file: spirou_wave_ini3.fits already exists - not copied
16:26:55.5 -   |-c:+[...]|Calibration file: dark_dark02d406.fits already exists - not copied
```

```
16:26:55.5 -   |-c:+[...]|Calibration file: dark_flat02f10_flat_C.fits already exists - not copied
16:26:55.5 -   |-c:+[...]|Calibration file: flat_dark02f10_loco_AB.fits already exists - not copied
16:26:55.5 -   |-c:+[...]|Reading File: /scratch/Projects/SPIRou_Pipeline/data/raw/20170710/fp_fp02a203.fits
16:26:55.5 -   |-c:+[...]|Image 2048x2048 loaded
16:26:55.6 - * |-c:+[...]|Adding frames
16:26:55.6 -   |-c:+[...]|Reading File: /scratch/Projects/SPIRou_Pipeline/data/raw/20170710/fp_fp03a203.fits
16:26:55.6 -   |-c:+[...]|Reading File: /scratch/Projects/SPIRou_Pipeline/data/raw/20170710/fp_fp04a203.fits
16:26:55.9 -   |-c:+[...]|Doing Dark Correction using /scratch/Projects/SPIRou_Pipeline/data/calibDB/
     dark_dark02d406.fits
16:26:56.2 -   |-c:+[...]|Image format changed to 2035x1930
16:26:56.7 - * |-c:+[...]|Nb pixels morts = 611716 / 15.58 %
16:26:56.7 -   |-c:+[...]|Reading localization parameters of Fiber AB
16:26:57.0 -   |-c:+[...]|Order  0 : Tilt = 4.70 pixel sur 37.0 = -7.24 deg
16:26:57.2 -   |-c:+[...]|Order  1 : Tilt = 4.60 pixel sur 37.4 = -7.02 deg
16:26:57.4 -   |-c:+[...]|Order  2 : Tilt = 4.50 pixel sur 36.8 = -6.97 deg
16:26:57.6 -   |-c:+[...]|Order  3 : Tilt = 4.30 pixel sur 36.4 = -6.74 deg
16:26:57.8 -   |-c:+[...]|Order  4 : Tilt = 4.20 pixel sur 36.8 = -6.52 deg
16:26:58.0 -   |-c:+[...]|Order  5 : Tilt = 4.20 pixel sur 36.0 = -6.65 deg
16:26:58.2 -   |-c:+[...]|Order  6 : Tilt = 4.00 pixel sur 36.1 = -6.33 deg
16:26:58.4 -   |-c:+[...]|Order  7 : Tilt = 4.00 pixel sur 35.7 = -6.40 deg
16:26:58.6 -   |-c:+[...]|Order  8 : Tilt = 3.90 pixel sur 36.7 = -6.07 deg
16:26:58.8 -   |-c:+[...]|Order  9 : Tilt = 3.80 pixel sur 36.1 = -6.01 deg
16:26:59.0 -   |-c:+[...]|Order 10 : Tilt = 3.60 pixel sur 35.7 = -5.76 deg
16:26:59.3 -   |-c:+[...]|Order 11 : Tilt = 3.60 pixel sur 35.7 = -5.75 deg
16:26:59.5 -   |-c:+[...]|Order 12 : Tilt = 3.50 pixel sur 35.9 = -5.57 deg
16:26:59.7 -   |-c:+[...]|Order 13 : Tilt = 3.30 pixel sur 35.8 = -5.27 deg
16:26:59.8 -   |-c:+[...]|Order 14 : Tilt = 3.30 pixel sur 35.9 = -5.25 deg
16:27:00.0 -   |-c:+[...]|Order 15 : Tilt = 3.20 pixel sur 35.4 = -5.17 deg
16:27:00.2 -   |-c:+[...]|Order 16 : Tilt = 3.10 pixel sur 35.5 = -4.99 deg
16:27:00.3 -   |-c:+[...]|Order 17 : Tilt = 3.00 pixel sur 35.4 = -4.85 deg
16:27:00.5 -   |-c:+[...]|Order 18 : Tilt = 2.90 pixel sur 35.1 = -4.73 deg
16:27:00.7 -   |-c:+[...]|Order 19 : Tilt = 2.80 pixel sur 35.1 = -4.57 deg
16:27:00.8 -   |-c:+[...]|Order 20 : Tilt = 2.70 pixel sur 34.9 = -4.42 deg
16:27:01.0 -   |-c:+[...]|Order 21 : Tilt = 2.60 pixel sur 34.6 = -4.29 deg
16:27:01.2 -   |-c:+[...]|Order 22 : Tilt = 2.50 pixel sur 34.9 = -4.09 deg
16:27:01.3 -   |-c:+[...]|Order 23 : Tilt = 2.40 pixel sur 34.2 = -4.01 deg
16:27:01.5 -   |-c:+[...]|Order 24 : Tilt = 2.30 pixel sur 34.9 = -3.77 deg
16:27:01.7 -   |-c:+[...]|Order 25 : Tilt = 2.20 pixel sur 34.7 = -3.63 deg
16:27:01.8 -   |-c:+[...]|Order 26 : Tilt = 2.00 pixel sur 34.2 = -3.35 deg
16:27:02.0 -   |-c:+[...]|Order 27 : Tilt = 2.00 pixel sur 34.5 = -3.32 deg
16:27:02.2 -   |-c:+[...]|Order 28 : Tilt = 1.80 pixel sur 34.2 = -3.01 deg
16:27:02.3 -   |-c:+[...]|Order 29 : Tilt = 1.70 pixel sur 34.0 = -2.86 deg
16:27:02.5 -   |-c:+[...]|Order 30 : Tilt = 1.60 pixel sur 33.6 = -2.73 deg
16:27:02.7 -   |-c:+[...]|Order 31 : Tilt = 1.40 pixel sur 34.0 = -2.36 deg
16:27:02.8 -   |-c:+[...]|Order 32 : Tilt = 1.40 pixel sur 33.5 = -2.40 deg
16:27:03.0 -   |-c:+[...]|Order 33 : Tilt = 1.10 pixel sur 32.4 = -1.94 deg
16:27:03.2 -   |-c:+[...]|Order 34 : Tilt = 1.00 pixel sur 32.1 = -1.79 deg
16:27:03.3 -   |-c:+[...]|Order 35 : Tilt = 0.90 pixel sur 33.7 = -1.53 deg
16:27:03.3 - * |-c:+[...]AB|Tilt dispersion = 0.069 deg
16:27:03.6 -   |-c:+[...]|Saving tilt information in file: fp_fp02a203_tilt.fits
16:27:05.2 - * |-c:+[...]|Updating Calib Data Base with TILT
16:27:05.2 - * |-c:+[...]|Recipe -c has been successfully completed
```

6.4.4 Interactive mode

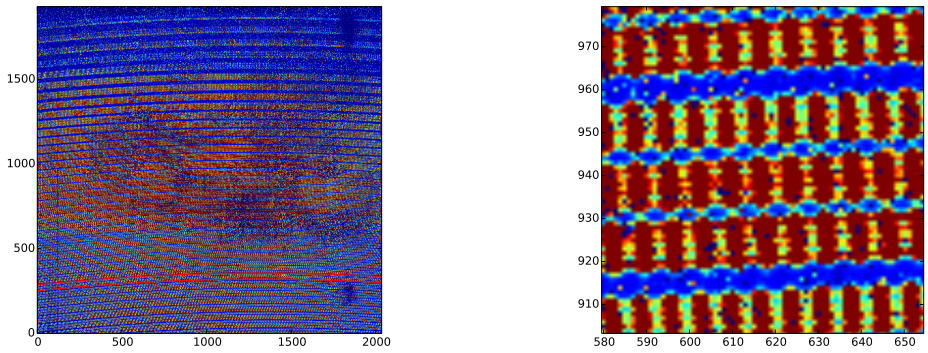

Figure 6.6 Left: The full 'fp_fp' image. Right: Zoom in on a section of the 'fp_fp' image showing the tilt.

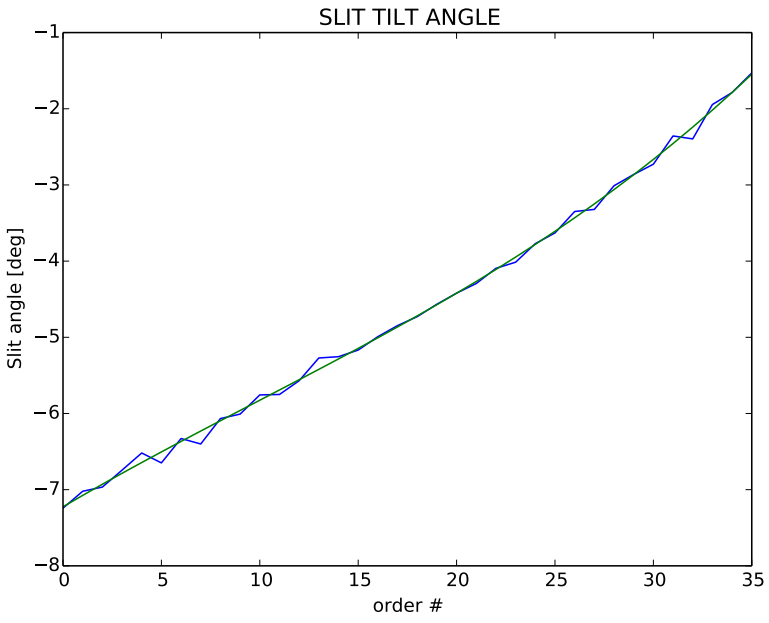

Figure 6.7 Slit aginle as a function of order number.

6.5 The cal_FF_RAW_spirou recipe

Creates the flat fields.

File prefixes allowed:

- dark_flat
- flat_dark

Note: if using same files in Section 3.2 you will get an error message when running the file. To solve this open the 'master_calib_SPIROU.txt' file located in {DATA_ROOT_CALIB}. Edit the unix date in the line that begins 'TILT' so that it is less than the unix date on rwos 'ORDER_PROFIL_AB' (i.e. change it from 1499707515.0 to 1499705515.0).
i.e. the 'master_calib_SPIROU.txt' file should look go from

```
DARK 20170710 dark_dark02d406.fits 07/10/17/16:37:48 1499704668.0
ORDER_PROFIL_C 20170710 dark_flat02f10_order_profil_C.fits 07/10/17/17:03:50 1499706230.0
LOC_C 20170710 dark_flat02f10_loco_C.fits 07/10/17/17:03:50 1499706230.0
ORDER_PROFIL_AB 20170710 flat_dark02f10_order_profil_AB.fits 07/10/17/17:07:08 1499706428.0
LOC_AB 20170710 flat_dark02f10_loco_AB.fits 07/10/17/17:07:08 1499706428.0
TILT 20170710 fp_fp02a203_tilt.fits 07/10/17/17:25:15 1499707515.0
```

to this:

```
DARK 20170710 dark_dark02d406.fits 07/10/17/16:37:48 1499704668.0
ORDER_PROFIL_C 20170710 dark_flat02f10_order_profil_C.fits 07/10/17/17:03:50 1499706230.0
LOC_C 20170710 dark_flat02f10_loco_C.fits 07/10/17/17:03:50 1499706230.0
ORDER_PROFIL_AB 20170710 flat_dark02f10_order_profil_AB.fits 07/10/17/17:07:08 1499706428.0
LOC_AB 20170710 flat_dark02f10_loco_AB.fits 07/10/17/17:07:08 1499706428.0
TILT 20170710 fp_fp02a203_tilt.fits 07/10/17/17:25:15 1499705515.0
```

6.5.1 Summary of procedure

1. adds all 'dark_flat' or 'flat_dark' files together
2. corrects for darks
3. resizes the image
4. possible background subtraction?
5. extracts the orders using tilt and weight
6. calculates the blaze
7. calculates the flat field, (flat = extraction / blaze)
8. stores the flat fields
9. does some quality control
10. updates calibDB with key "FLAT_{fiber}" where {fiber} = [AB, C] etc

6.5.2 Running cal_FF_RAW_spirou

To run cal_FF_RAW_spirou type:

```
cal_FF_RAW_spirou.py  night_repository  filenames
```

Note: filenames must start with 'dark_flat' or 'flat_dark'

6.5.3 Example working run

An example run where everything worked is below:

```
cal_FF_RAW_spirou.py 20170710 dark_flat02f10.fits dark_flat03f10.fits
dark_flat04f10.fits dark_flat05f10.fits dark_flat06f10.fits

||DRS  SPIROU   v   (interactive mode)
16:49:40.5 -   || ****************************************
16:49:40.5 -   || * SPIROU @(#) Geneva Observatory ()
16:49:40.5 -   || ****************************************
16:49:40.5 -   ||(dir_data_raw)      DRS_DATA_RAW=/scratch/Projects/SPIRou_Pipeline/data/raw/
16:49:40.5 -   ||(dir_data_reduc)    DRS_DATA_REDUC=/scratch/Projects/SPIRou_Pipeline/data/reduced/
16:49:40.5 -   ||(dir_drs_config)    DRS_CONFIG=/scratch/Projects/SPIRou_Pipeline/INTROOT/DRS_SPIROU/config/
16:49:40.5 -   ||(dir_calib_db)      DRS_CALIB_DB=/scratch/Projects/SPIRou_Pipeline/data/calibDB
16:49:40.5 -   ||(dir_data_msg)      DRS_DATA_MSG=/scratch/Projects/SPIRou_Pipeline/data/msg/
16:49:40.5 -   ||(print_log)         DRS_LOG=1         %(0: minimum stdin-out logs)
16:49:40.5 -   ||(plot_graph)        DRS_PLOT=NONE           %(def/undef/trigger)
16:49:40.5 -   ||(used_date)         DRS_USED_DATE=undefined
16:49:40.5 -   ||(working_dir)       DRS_DATA_WORKING=/scratch/Projects/SPIRou_Pipeline/data/tmp/
16:49:40.5 -   ||                    DRS_INTERACTIVE is  set
16:49:40.5 -   |-c:+[...]|Now running : -c on file(s):  dark_flat02f10.fits dark_flat03f10.fits
16:49:40.5 -   |-c:+[...]|On directory /scratch/Projects/SPIRou_Pipeline/data/raw/20170710
16:49:40.5 -   |-c:+[...]|ICDP loaded from: /scratch/Projects/SPIRou_Pipeline/INTROOT/DRS_SPIROU/config/
     hadmrICDP_SPIROU.py
16:49:40.5 - * |-c:+[...]|Now processing Image TYPE: UNKNOWN  with -c recipe
16:49:40.6 -   |-c:+[...]|Calibration file: dark_dark02d406.fits already exists - not copied
16:49:40.6 -   |-c:+[...]|Calibration file: fp_fp02a203_tilt.fits already exists - not copied
16:49:40.6 -   |-c:+[...]|Reading File: /scratch/Projects/SPIRou_Pipeline/data/raw/20170710/dark_flat02f10.fits
16:49:40.6 -   |-c:+[...]|Image 2048x2048 loaded
16:49:40.6 - * |-c:+[...]|Adding frames
16:49:40.6 -   |-c:+[...]|Reading File: /scratch/Projects/SPIRou_Pipeline/data/raw/20170710/dark_flat03f10.fits
16:49:40.7 -   |-c:+[...]|Doing Dark Correction using /scratch/Projects/SPIRou_Pipeline/data/calibDB/
     dark_dark02d406.fits
16:49:41.0 -   |-c:+[...]|Image format changed to 2035x1930
16:49:41.5 - * |-c:+[...]|Nb pixels morts = 568533 / 14.48 %
16:49:41.5 - * |-c:+[...]|Maximum average flux/pixel in the spectrum: 73264.1[ADU]
16:49:41.5 -   |-c:+[...]|Reading tilt slit
16:49:41.5 -   |-c:+[...]C|Reading localization parameters of Fiber C
16:49:41.6 -   |-c:+[...]C|Reading order profil of Fiber C
16:49:42.1 -   |-c:+[...]|On fiber C  order  0: S/N= 685.6  - FF rms= 4.68 %
16:49:42.7 -   |-c:+[...]|On fiber C  order  1: S/N= 709.3  - FF rms= 4.81 %
16:49:43.3 -   |-c:+[...]|On fiber C  order  2: S/N= 733.7  - FF rms= 4.75 %
16:49:43.8 -   |-c:+[...]|On fiber C  order  3: S/N= 759.2  - FF rms= 4.86 %
16:49:44.4 -   |-c:+[...]|On fiber C  order  4: S/N= 780.7  - FF rms= 4.94 %
16:49:44.9 -   |-c:+[...]|On fiber C  order  5: S/N= 811.4  - FF rms= 5.03 %
16:49:45.5 -   |-c:+[...]|On fiber C  order  6: S/N= 842.1  - FF rms= 5.11 %
16:49:46.0 -   |-c:+[...]|On fiber C  order  7: S/N= 875.4  - FF rms= 5.23 %
16:49:46.6 -   |-c:+[...]|On fiber C  order  8: S/N= 893.9  - FF rms= 5.37 %
16:49:47.2 -   |-c:+[...]|On fiber C  order  9: S/N= 916.4  - FF rms= 6.07 %
16:49:47.7 -   |-c:+[...]|On fiber C  order 10: S/N= 928.3  - FF rms= 8.02 %
16:49:48.3 -   |-c:+[...]|On fiber C  order 11: S/N= 912.5  - FF rms= 6.84 %
16:49:48.8 -   |-c:+[...]|On fiber C  order 12: S/N= 961.8  - FF rms= 5.94 %
16:49:49.4 -   |-c:+[...]|On fiber C  order 13: S/N= 1004.9  - FF rms= 5.97 %
16:49:49.9 -   |-c:+[...]|On fiber C  order 14: S/N= 1022.5  - FF rms= 6.15 %
16:49:50.5 -   |-c:+[...]|On fiber C  order 15: S/N= 1045.5  - FF rms= 6.18 %
16:49:51.1 -   |-c:+[...]|On fiber C  order 16: S/N= 1053.4  - FF rms= 6.14 %
16:49:51.6 -   |-c:+[...]|On fiber C  order 17: S/N= 1085.8  - FF rms= 6.23 %
16:49:52.2 -   |-c:+[...]|On fiber C  order 18: S/N= 1103.0  - FF rms= 6.28 %
16:49:52.7 -   |-c:+[...]|On fiber C  order 19: S/N= 1134.8  - FF rms= 6.77 %
16:49:53.3 -   |-c:+[...]|On fiber C  order 20: S/N= 1150.4  - FF rms= 6.43 %
16:49:53.9 -   |-c:+[...]|On fiber C  order 21: S/N= 1144.2  - FF rms= 6.42 %
16:49:54.4 -   |-c:+[...]|On fiber C  order 22: S/N= 1139.6  - FF rms= 6.39 %
16:49:55.0 -   |-c:+[...]|On fiber C  order 23: S/N= 1161.3  - FF rms= 6.52 %
16:49:55.5 -   |-c:+[...]|On fiber C  order 24: S/N= 1137.8  - FF rms= 9.11 %
16:49:56.1 -   |-c:+[...]|On fiber C  order 25: S/N= 1131.4  - FF rms= 9.39 %
16:49:56.6 -   |-c:+[...]|On fiber C  order 26: S/N= 1113.7  - FF rms= 7.97 %
16:49:57.2 -   |-c:+[...]|On fiber C  order 27: S/N= 1161.7  - FF rms= 6.04 %
16:49:57.7 -   |-c:+[...]|On fiber C  order 28: S/N= 1153.9  - FF rms= 6.07 %
16:49:58.3 -   |-c:+[...]|On fiber C  order 29: S/N= 1178.2  - FF rms= 5.89 %
16:49:58.9 -   |-c:+[...]|On fiber C  order 30: S/N= 1150.4  - FF rms= 5.92 %
16:49:59.6 -   |-c:+[...]|On fiber C  order 31: S/N= 1024.1  - FF rms= 5.60 %
16:50:00.3 -   |-c:+[...]|On fiber C  order 32: S/N= 945.1  - FF rms= 5.60 %
16:50:01.0 -   |-c:+[...]|On fiber C  order 33: S/N= 1030.4  - FF rms= 5.70 %
16:50:01.6 -   |-c:+[...]|On fiber C  order 34: S/N= 957.7  - FF rms= 8.21 %
16:50:02.2 -   |-c:+[...]|On fiber C  order 35: S/N= 753.1  - FF rms= 8.11 %
```

```
16:50:02.4 -   |-c:+[...]C|Saving blaze spectrum for fiber: C in dark_flat02f10_blaze_C.fits
16:50:03.1 -   |-c:+[...]C|Saving FF  spectrum of Fiber C in dark_flat02f10_flat_C.fits
16:50:04.2 - * |-c:+[...]|Updating Calib Data Base with FLAT_C
16:50:04.2 - * |-c:+[...]|Recipe -c has been  successfully completed
```

6.5.4 Interactive mode

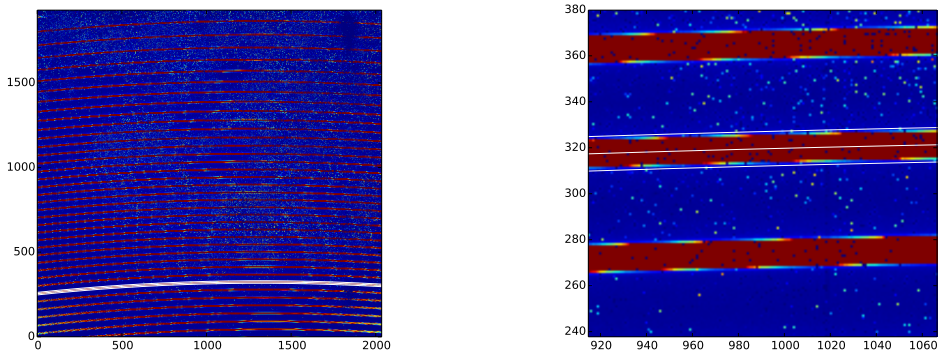

Figure 6.8 Left: the full processed image with one order fit highlighted. Right: Zoom in of the highlighted order fit.

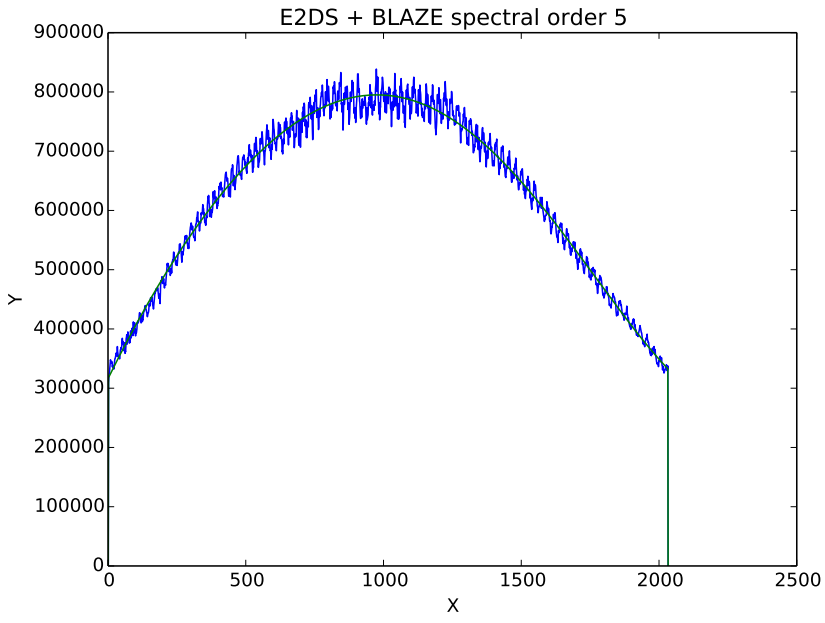

Figure 6.9 The extracted highlighted order from Figure 6.8. Overplotted is the blaze fit (in cyan).

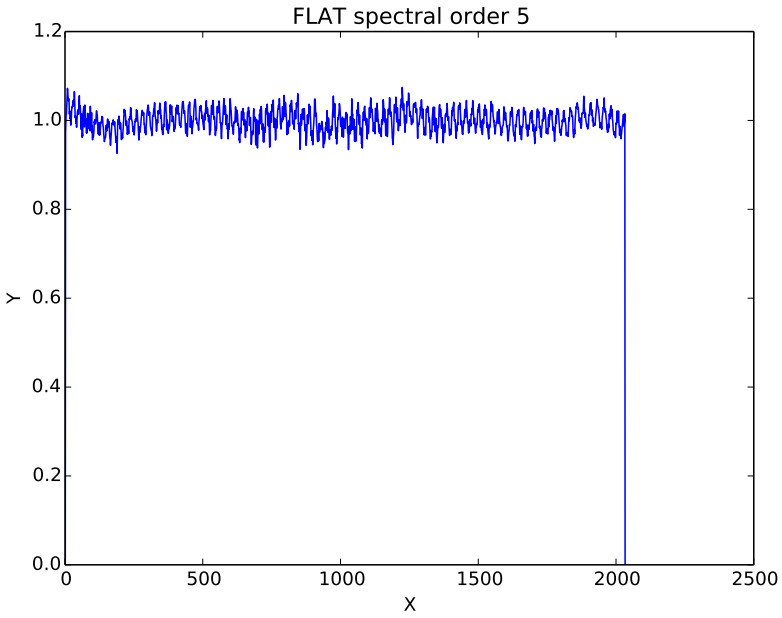

Figure 6.10 The flattened order for highlighted order from Figure 6.8.

6.6 The cal_extract_RAW_spirou recipe

Extracts orders for specific fibers and files.

File prefixes allowed:

- fp_fp
- hcone_dark
- dark_hcone
- hcone_hcone
- dark_dark_AHC1
- dark_hctwo
- hctwo_hctwo
- dark_dark_AHC2

6.6.1 Summary of procedure

1. adds all files together
2. corrects for darks
3. resizes the image
4. checks for saturation
5. possible background subtraction?
6. extracts orders
   - without tilt/weight fortran
   - without tilt/weight python
   - with tilt (no weight)
   - with tilt and weight
   - with weight (no tilt)
7. saves extraction with weight (no tilt) to e2ds file

6.6.2 Running cal_extract_RAW_spirou

To run cal_extract_RAW_spirou on fiber=AB:

```
cal_extract_RAW_spirouAB.py  night_repository  filenames
```

To run cal_extract_RAW_spirou on fiber=C:

```
cal_extract_RAW_spirouC.py  night_repository  filenames
```

Note: Filenames must start with 'fp_fp', 'hcone_dark', 'dark_hcone', 'hcone_hcone', 'dark_dark_AHC1', 'dark_hctwo', 'hctwo_hctwo', or 'dark_dark_AHC2'.

6.6.3 Example working run

An example run where everything worked is below:

```
cal_extract_RAW_spirouC.py 20170710 fp_fp02a203.fits fp_fp03a203.fits
fp_fp04a203.fits

||DRS  SPIROU   v   (interactive mode)
17:38:04.5 -   || ****************************************
17:38:04.5 -   || * SPIROU @(#) Geneva Observatory ()
17:38:04.5 -   || ****************************************
17:38:04.5 -   ||(dir_data_raw)      DRS_DATA_RAW=/scratch/Projects/SPIRou_Pipeline/data/raw/
17:38:04.5 -   ||(dir_data_reduc)    DRS_DATA_REDUC=/scratch/Projects/SPIRou_Pipeline/data/reduced/
17:38:04.5 -   ||(dir_drs_config)    DRS_CONFIG=/scratch/Projects/SPIRou_Pipeline/INTROOT/DRS_SPIROU/config/
17:38:04.5 -   ||(dir_calib_db)      DRS_CALIB_DB=/scratch/Projects/SPIRou_Pipeline/data/calibDB
17:38:04.5 -   ||(dir_data_msg)      DRS_DATA_MSG=/scratch/Projects/SPIRou_Pipeline/data/msg/
17:38:04.5 -   ||(print_log)         DRS_LOG=1         %(0: minimum stdin-out logs)
17:38:04.5 -   ||(plot_graph)        DRS_PLOT=NONE            %(def/undef/trigger)
17:38:04.5 -   ||(used_date)         DRS_USED_DATE=undefined
17:38:04.5 -   ||(working_dir)       DRS_DATA_WORKING=/scratch/Projects/SPIRou_Pipeline/data/tmp/
17:38:04.5 -   ||                    DRS_INTERACTIVE is  set
17:38:04.5 -   |-c:+[...]|Now running : -c on file(s):  fp_fp02a203.fits fp_fp03a203.fits
17:38:04.5 -   |-c:+[...]|On directory /scratch/Projects/SPIRou_Pipeline/data/raw/20170710
17:38:04.5 -   |-c:+[...]|ICDP loaded from: /scratch/Projects/SPIRou_Pipeline/INTROOT/DRS_SPIROU/config/
     hadmrICDP_SPIROU.py
17:38:04.5 - * |-c:+[...]|Now processing Image TYPE: UNKNOWN  with -c recipe
17:38:04.6 -   |-c:+[...]|Calibration file: fp_fp02a203_tilt.fits already exists - not copied
17:38:04.6 -   |-c:+[...]|Calibration file: dark_flat02f10_loco_C.fits already exists - not copied
17:38:04.6 -   |-c:+[...]|Calibration file: flat_dark02f10_order_profil_AB.fits already exists - not copied
17:38:04.6 -   |-c:+[...]|Calibration file: spirou_wave_ini3.fits already exists - not copied
17:38:04.7 -   |-c:+[...]|Calibration file: dark_dark02d406.fits already exists - not copied
17:38:04.7 -   |-c:+[...]|Calibration file: dark_flat02f10_flat_C.fits already exists - not copied
17:38:04.7 -   |-c:+[...]|Calibration file: dark_flat02f10_order_profil_C.fits already exists - not copied
17:38:04.7 -   |-c:+[...]|Calibration file: flat_dark02f10_loco_AB.fits already exists - not copied
17:38:04.7 -   |-c:+[...]|Reading File: /scratch/Projects/SPIRou_Pipeline/data/raw/20170710/fp_fp02a203.fits
17:38:04.7 -   |-c:+[...]|Image 2048x2048 loaded
17:38:04.8 - * |-c:+[...]|Adding frames
17:38:04.8 -   |-c:+[...]|Reading File: /scratch/Projects/SPIRou_Pipeline/data/raw/20170710/fp_fp03a203.fits
17:38:04.9 -   |-c:+[...]|Doing Dark Correction using /scratch/Projects/SPIRou_Pipeline/data/calibDB/
     dark_dark02d406.fits
17:38:05.1 -   |-c:+[...]|Image format changed to 2035x1930
17:38:05.6 - * |-c:+[...]|Nb pixels morts = 568485 / 14.47 %
17:38:05.6 - * |-c:+[...]|Maximum average flux/pixel in the spectrum: 110446.0[ADU]
17:38:05.6 -   |-c:+[...]|Reading tilt slit
17:38:05.7 -   |-c:+[...]|Reading wavelength solution
17:38:05.7 -   |-c:+[...]C|Reading localization parameters of Fiber C
17:38:05.7 -   |-c:+[...]C|Reading order profil of Fiber C
17:38:07.1 -   |-c:+[...]|On fiber C  order  0: S/N= 712.0
17:38:08.9 -   |-c:+[...]|On fiber C  order  1: S/N= 780.2
17:38:10.3 -   |-c:+[...]|On fiber C  order  2: S/N= 726.0
17:38:12.0 -   |-c:+[...]|On fiber C  order  3: S/N= 700.3
17:38:13.5 -   |-c:+[...]|On fiber C  order  4: S/N= 702.3
17:38:14.9 -   |-c:+[...]|On fiber C  order  5: S/N= 740.0
17:38:16.3 -   |-c:+[...]|On fiber C  order  6: S/N= 773.4
17:38:17.7 -   |-c:+[...]|On fiber C  order  7: S/N= 758.5
17:38:19.2 -   |-c:+[...]|On fiber C  order  8: S/N= 752.0
17:38:20.7 -   |-c:+[...]|On fiber C  order  9: S/N= 763.0
17:38:22.1 -   |-c:+[...]|On fiber C  order 10: S/N= 794.5
17:38:23.5 -   |-c:+[...]|On fiber C  order 11: S/N= 671.5
17:38:25.2 -   |-c:+[...]|On fiber C  order 12: S/N= 871.4
17:38:26.7 -   |-c:+[...]|On fiber C  order 13: S/N= 883.0
17:38:28.4 -   |-c:+[...]|On fiber C  order 14: S/N= 1070.7
17:38:29.9 -   |-c:+[...]|On fiber C  order 15: S/N= 954.4
17:38:31.6 -   |-c:+[...]|On fiber C  order 16: S/N= 909.4
17:38:33.3 -   |-c:+[...]|On fiber C  order 17: S/N= 935.8
17:38:34.9 -   |-c:+[...]|On fiber C  order 18: S/N= 905.3
17:38:36.4 -   |-c:+[...]|On fiber C  order 19: S/N= 888.5
17:38:37.8 -   |-c:+[...]|On fiber C  order 20: S/N= 896.3
17:38:39.6 -   |-c:+[...]|On fiber C  order 21: S/N= 888.1
17:38:41.3 -   |-c:+[...]|On fiber C  order 22: S/N= 848.4
17:38:43.0 -   |-c:+[...]|On fiber C  order 23: S/N= 860.3
17:38:44.7 -   |-c:+[...]|On fiber C  order 24: S/N= 817.9
17:38:46.1 -   |-c:+[...]|On fiber C  order 25: S/N= 813.0
17:38:47.5 -   |-c:+[...]|On fiber C  order 26: S/N= 792.2
17:38:49.0 -   |-c:+[...]|On fiber C  order 27: S/N= 836.9
17:38:50.7 -   |-c:+[...]|On fiber C  order 28: S/N= 882.4
```

```
17:38:52.3 -   |-c:+[...]|On fiber C  order 29: S/N= 837.6
17:38:53.7 -   |-c:+[...]|On fiber C  order 30: S/N= 785.7
17:38:55.1 -   |-c:+[...]|On fiber C  order 31: S/N= 434.9
17:38:56.5 -   |-c:+[...]|On fiber C  order 32: S/N= 376.3
17:38:58.0 -   |-c:+[...]|On fiber C  order 33: S/N= 614.3
17:38:59.7 -   |-c:+[...]|On fiber C  order 34: S/N= 523.1
17:39:01.2 -   |-c:+[...]|On fiber C  order 35: S/N= 343.2
17:39:01.5 -   |-c:+[...]|Saving E2DS spectrum of Fiber C in fp_fp02a203_e2ds_C.fits
17:39:02.1 - * |-c:+[...]|Recipe -c has been  successfully completed
```


6.6.4 Interactive mode

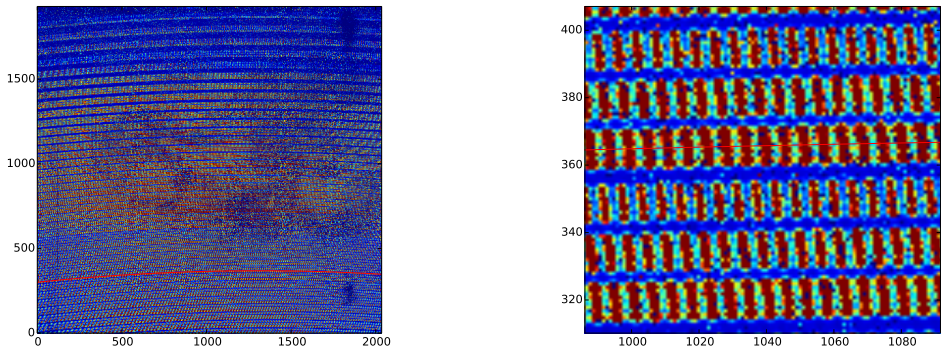

Figure 6.11 Left: The full process image. Right: Zoom in on the highlighted spectral order fit.

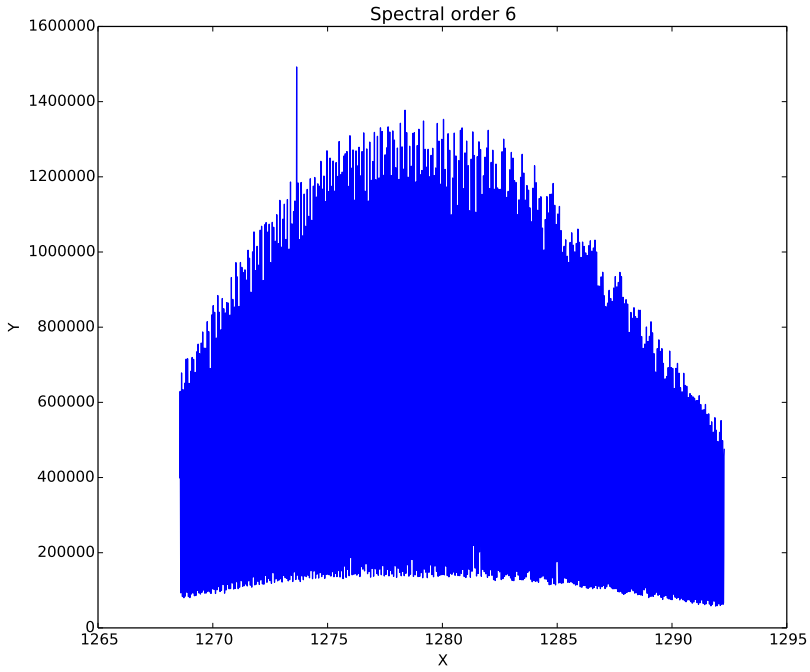

Figure 6.12 The extract spectrum from the hightlighted spectral order in 6.11.

6.7 The cal_DRIFT_RAW_spirou recipe

Calculates relative (radial velocity) drift between all files and individual files.

File prefixes allowed:

- fp_fp

6.7.1 Summary of procedure

1. first file is reference image
2. resizes the image
3. extracts with weight (no tilt)
4. loops around all other 'fp_fp' files in directory
5. calculates photon noise uncertainty and estimated RV uncertainty on spectrum
    - uses wave file
6. calculates RV drift and mean RV drift between reference (mean of files) and other 'fp_fp' files
7. saves drift values to file

6.7.2 Running cal_DRIFT_RAW_spirou

To run cal_DRIFT_RAW_spirou type:

```
cal_DRIFT_RAW_spirou.py  night_repository  filenames
```

Note: Filenames must start with 'fp_fp'.

6.7.3 Example working run

An example run where everything worked is below:

```
cal_DRIFT_RAW_spirou.py 20170710 fp_fp02a203.fits fp_fp03a203.fits
fp_fp04a203.fits

||DRS  SPIROU   v   (interactive mode)
17:58:26.1 -   || ****************************************
17:58:26.1 -   || * SPIROU @(#) Geneva Observatory ()
17:58:26.1 -   || ****************************************
17:58:26.1 -   ||(dir_data_raw)      DRS_DATA_RAW=/scratch/Projects/SPIRou_Pipeline/data/raw/
17:58:26.1 -   ||(dir_data_reduc)    DRS_DATA_REDUC=/scratch/Projects/SPIRou_Pipeline/data/reduced/
17:58:26.1 -   ||(dir_drs_config)    DRS_CONFIG=/scratch/Projects/SPIRou_Pipeline/INTROOT/DRS_SPIROU/config/
17:58:26.1 -   ||(dir_calib_db)      DRS_CALIB_DB=/scratch/Projects/SPIRou_Pipeline/data/calibDB
17:58:26.1 -   ||(dir_data_msg)      DRS_DATA_MSG=/scratch/Projects/SPIRou_Pipeline/data/msg/
17:58:26.1 -   ||(print_log)         DRS_LOG=1         %(0: minimum stdin-out logs)
17:58:26.1 -   ||(plot_graph)        DRS_PLOT=NONE           %(def/undef/trigger)
17:58:26.1 -   ||(used_date)         DRS_USED_DATE=undefined
17:58:26.1 -   ||(working_dir)       DRS_DATA_WORKING=/scratch/Projects/SPIRou_Pipeline/data/tmp/
17:58:26.1 -   ||                    DRS_INTERACTIVE is  set
17:58:26.1 -   |-c:+[...]|Now running : -c on file(s):  fp_fp02a203.fits fp_fp03a203.fits
17:58:26.1 -   |-c:+[...]|On directory /scratch/Projects/SPIRou_Pipeline/data/raw/20170710
17:58:26.1 -   |-c:+[...]|ICDP loaded from: /scratch/Projects/SPIRou_Pipeline/INTROOT/DRS_SPIROU/config/
     hadmrICDP_SPIROU.py
17:58:26.1 - * |-c:+[...]|Now processing Image TYPE: UNKNOWN  with -c recipe
17:58:26.2 -   |-c:+[...]|Calibration file: fp_fp02a203_tilt.fits already exists - not copied
17:58:26.2 -   |-c:+[...]|Calibration file: dark_flat02f10_loco_C.fits already exists - not copied
17:58:26.2 -   |-c:+[...]|Calibration file: flat_dark02f10_order_profil_AB.fits already exists - not copied
```

```
17:58:26.2 -   |-c:+[...]|Calibration file: spirou_wave_ini3.fits already exists - not copied
17:58:26.3 -   |-c:+[...]|Calibration file: dark_dark02d406.fits already exists - not copied
17:58:26.3 -   |-c:+[...]|Calibration file: dark_flat02f10_flat_C.fits already exists - not copied
17:58:26.3 -   |-c:+[...]|Calibration file: dark_flat02f10_order_profil_C.fits already exists - not copied
17:58:26.3 -   |-c:+[...]|Calibration file: flat_dark02f10_loco_AB.fits already exists - not copied
17:58:26.3 -   |-c:+[...]|Reading File: /scratch/Projects/SPIRou_Pipeline/data/raw/20170710/fp_fp02a203.fits
17:58:26.3 -   |-c:+[...]|Image 2048x2048 loaded
17:58:26.4 -   |-c:+[...]|Doing Dark Correction using /scratch/Projects/SPIRou_Pipeline/data/calibDB/
     dark_dark02d406.fits
17:58:26.8 -   |-c:+[...]|Image format changed to 2035x1930
17:58:27.2 - * |-c:+[...]|Nb pixels morts = 600284 / 15.28 %
17:58:27.2 -   |-c:+[...]|Reading tilt slit
17:58:27.2 -   |-c:+[...]AB|Reading localization parameters of Fiber AB
17:58:27.3 -   |-c:+[...]AB|Reading order profil of Fiber AB
17:58:27.3 -   |-c:+[...]|Reading wavelength solution
17:58:27.3 -   |-c:+[...]|Extraction Reference file  /scratch/Projects/SPIRou_Pipeline/data/raw/20170710/
     fp_fp02a203.fits
17:58:27.4 -   |-c:+[...]|On fiber AB  order  0: S/N= 513.5
17:58:27.6 -   |-c:+[...]|On fiber AB  order  1: S/N= 561.6
17:58:27.7 -   |-c:+[...]|On fiber AB  order  2: S/N= 525.8
17:58:27.9 -   |-c:+[...]|On fiber AB  order  3: S/N= 520.4
17:58:28.0 -   |-c:+[...]|On fiber AB  order  4: S/N= 518.1
17:58:28.1 -   |-c:+[...]|On fiber AB  order  5: S/N= 549.3
17:58:28.3 -   |-c:+[...]|On fiber AB  order  6: S/N= 590.9
17:58:28.4 -   |-c:+[...]|On fiber AB  order  7: S/N= 574.5
17:58:28.6 -   |-c:+[...]|On fiber AB  order  8: S/N= 582.1
17:58:28.7 -   |-c:+[...]|On fiber AB  order  9: S/N= 592.4
17:58:28.9 -   |-c:+[...]|On fiber AB  order 10: S/N= 615.4
17:58:29.0 -   |-c:+[...]|On fiber AB  order 11: S/N= 525.9
17:58:29.2 -   |-c:+[...]|On fiber AB  order 12: S/N= 697.1
17:58:29.3 -   |-c:+[...]|On fiber AB  order 13: S/N= 735.1
17:58:29.5 -   |-c:+[...]|On fiber AB  order 14: S/N= 920.8
17:58:29.6 -   |-c:+[...]|On fiber AB  order 15: S/N= 830.5
17:58:29.8 -   |-c:+[...]|On fiber AB  order 16: S/N= 818.3
17:58:29.9 -   |-c:+[...]|On fiber AB  order 17: S/N= 840.0
17:58:30.1 -   |-c:+[...]|On fiber AB  order 18: S/N= 814.2
17:58:30.2 -   |-c:+[...]|On fiber AB  order 19: S/N= 829.1
17:58:30.4 -   |-c:+[...]|On fiber AB  order 20: S/N= 836.1
17:58:30.5 -   |-c:+[...]|On fiber AB  order 21: S/N= 826.4
17:58:30.7 -   |-c:+[...]|On fiber AB  order 22: S/N= 812.6
17:58:30.8 -   |-c:+[...]|On fiber AB  order 23: S/N= 828.2
17:58:31.0 -   |-c:+[...]|On fiber AB  order 24: S/N= 785.1
17:58:31.1 -   |-c:+[...]|On fiber AB  order 25: S/N= 770.6
17:58:31.3 -   |-c:+[...]|On fiber AB  order 26: S/N= 743.6
17:58:31.4 -   |-c:+[...]|On fiber AB  order 27: S/N= 758.9
17:58:31.6 -   |-c:+[...]|On fiber AB  order 28: S/N= 765.8
17:58:31.7 -   |-c:+[...]|On fiber AB  order 29: S/N= 685.6
17:58:31.9 -   |-c:+[...]|On fiber AB  order 30: S/N= 612.8
17:58:32.0 -   |-c:+[...]|On fiber AB  order 31: S/N= 307.0
17:58:32.2 -   |-c:+[...]|On fiber AB  order 32: S/N= 255.6
17:58:32.3 -   |-c:+[...]|On fiber AB  order 33: S/N= 355.5
17:58:32.5 -   |-c:+[...]|On fiber AB  order 34: S/N= 283.3
17:58:32.6 -   |-c:+[...]|On fiber AB  order 35: S/N= 149.9
17:58:32.9 - * |-c:+[...]|On fiber AB estimated RV uncertainty on spectrum is 0.028 m/s
17:58:32.9 - * |-c:+[...]|Nb fp_fp files found on directory =  2
17:58:32.9 -   |-c:+[...]|Reading File: /scratch/Projects/SPIRou_Pipeline/data/raw/20170710/fp_fp03a203.fits
17:58:39.3 -   |-c:+[...]|Time from ref= 0.09 h  - Drift mean= 2.53 m/s - Flux ratio= 0.99 - Nb Cosmics= 1218
17:58:39.3 -   |-c:+[...]|Reading File: /scratch/Projects/SPIRou_Pipeline/data/raw/20170710/fp_fp04a203.fits
17:58:45.6 -   |-c:+[...]|Time from ref= 0.18 h  - Drift mean= -10.14 m/s - Flux ratio= 0.98 - Nb Cosmics= 1246
17:58:45.6 -   |-c:+[...]|Total drift Peak-To-Peak= 12.607 m/s RMS= 5.201 m/s in 0.00 hour
17:58:45.6 -   |-c:+[...]|Saving drift values of Fiber AB in fp_fp02a203_drift_AB.fits
17:58:45.8 - * |-c:+[...]|Recipe -c has been  successfully completed
```

6.7.4 Interactive mode

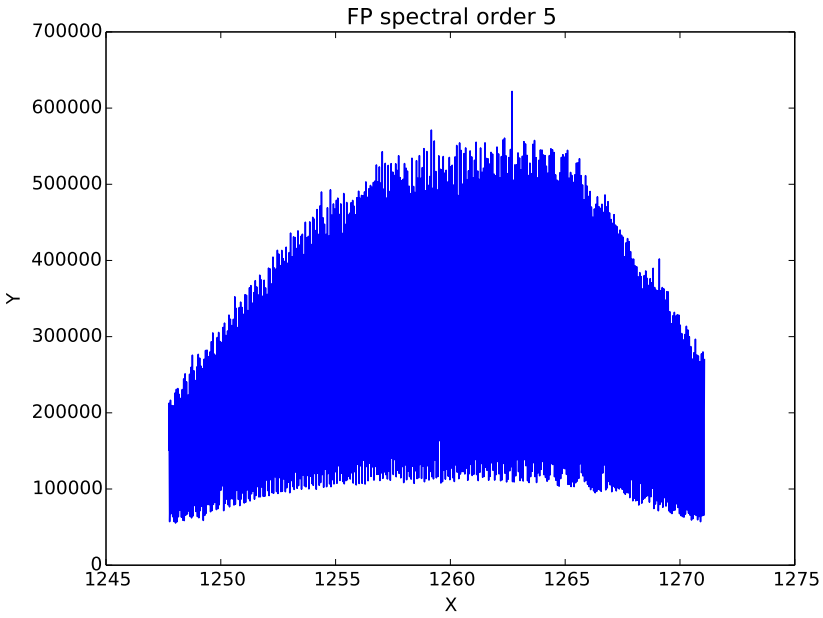

Figure 6.13 Extract FP spectral order.

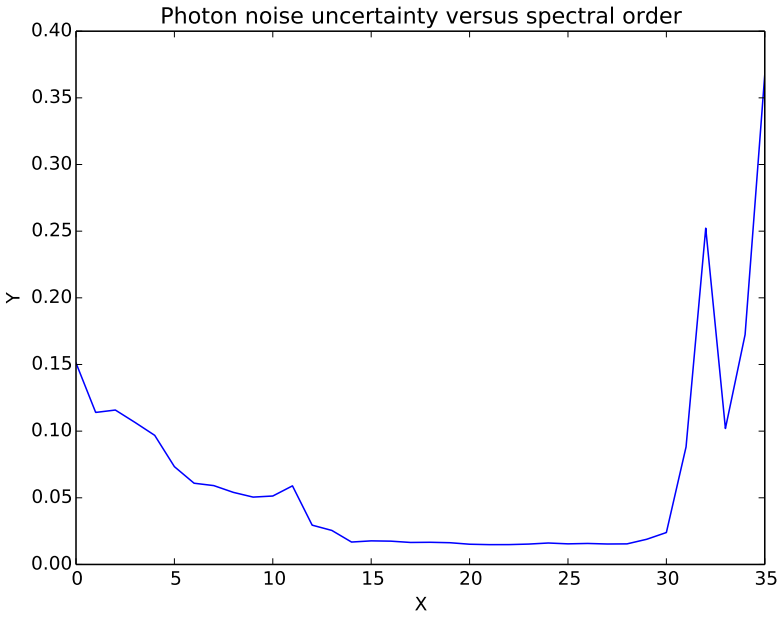

Figure 6.14 Photon noise uncertain against spectral order.

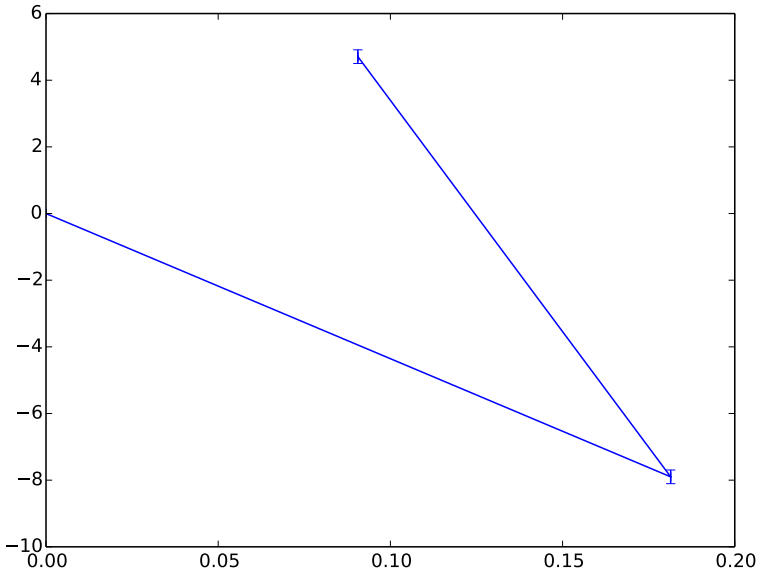

Figure 6.15 Time interval against median drift.

6.8 The cal_HC_E2DS_spirou recipe

Wavelength calibration.

File prefixes allowed:

- hc_hc*AB.fits
- hc_hc*C.fits

6.8.1 Summary of procedure

1. Reads lamp line file
2. Tries to identify lines using guess solution
3. Cleans list of identified lines
4. Fits wavelength solution on identified lines
5. Performs a Littrow test
6. Writes wavelength solution for fibers
7. Computes CCF

6.8.2 Running cal_HC_E2DS_spirou

To run cal_HC_E2DS_spirou type:

```
cal_HC_E2DS_spirou.py  night_repository  hc_hc*.fits
```

Note: Filenames must start with hc_hc Note: Filenames must include either AB or C

6.9 The cal_DRIFT-PEAK_E2DS_spirou recipe

Calculates relative (radial velocity) drift between all files and individual files using a gaussian fitting process to fit the FP peaks.

File prefixes allowed:

- fp_fp

6.9.1 Summary of procedure

1. first file is reference image
2. resizes the image
3. background correction
4. Identifies FP peaks in reference file
5. Creates a reference ascii file that contains the positions of the FP peaks
6. Removes lines with suspicious widths
7. loops around all other *_e2ds_{fiber}.fits files
8. Gets the centroid of all peaks using Gaussian fitting
9. Performs a Pearson R test to look for issues with extraction and/or illumination
10. Performs sigma clipping on measured drift
11. Saves drifts to file

6.9.2 Running cal_DRIFT-PEAK_E2DS_spirou

To run cal_DRIFT-PEAK_E2DS_spirou type:

```
cal_DRIFT-PEAK_E2DS_spirou.py  night_repository fp_fp*.fits
```

Note: Filenames must start with fp_fp.

6.10 The cal_WAVE_E2DS_spirou recipe

New wavelength solution combining Hollow-Cathode + FP E2DS spectra.

File prefixes allowed:

- hc_hc (first file)
- fp_fp (second file)

6.10.1 Summary of procedure

1. Reads lamp line file
2. Performs instrumental drift computation
3. Writes drift to e2ds fits file
4. Calculates initial wavelength solution
5. Tries to identify lines using guess solution (x2)
6. Cleans list of identified lines (x2)
7. Fits wavelength solution on identified lines (x2)
8. Performs a Littrow test (x2)
9. Writes wavelength solution for fibers
10. Computes CCF

6.10.2 Running cal_WAVE_E2DS_spirou

To run cal_WAVE_E2DS_spirou type:

```
cal_WAVE_E2DS_spirou.py night_repository  hc_hc_*_e2ds_AB.fits    fp_fp_*_e2ds_AB.fits
```

Note: First file must start with hc_hc and contain AB or C Note: Second file must start with fp_fp and contain AB or C

6.11 The cal_BADPIX_spirou recipe

Recipe to generate the bad pixel map.

File prefixes allowed:

- flat_flat (first file)
- dark_dark (second file)

6.11.1 Summary of procedure

1. Normalise the flats
2. Look for isolated hot pixels
3. Calculate how much pixels deviate compared to expected values
4. Select hot pixels compared to neighbours
5. Combine bad pixel map
6. Save bad pixel mask to file

6.11.2 Running cal_DRIFT_RAW_spirou

To run cal_DRIFT_RAW_spirou type:

```
cal_BADPIX_spirou.py night_repository flat_flat*.fits dark_dark*.fits
```

Note: Filenames must start with flat_flat (first file) and dark_dark (second file)
Note: only two files allowed (one flat plus one dark)

6.11.3 Example working run

An example run where everything worked is below:

```
cal_BADPIX_spirou.py 20170710 flat_flat02f10.fits dark_dark02d406.fits
||DRS  SPIROU   v   (interactive mode)
15:49:54.7 -   || ****************************************
15:49:54.7 -   || * SPIROU @(#) Geneva Observatory ()
15:49:54.7 -   || ****************************************
15:49:54.7 -   ||(dir_data_raw)      DRS_DATA_RAW=/scratch/Projects/SPIRou_Pipeline/data/raw/
15:49:54.7 -   ||(dir_data_reduc)    DRS_DATA_REDUC=/scratch/Projects/SPIRou_Pipeline/data/reduced/
15:49:54.7 -   ||(dir_drs_config)    DRS_CONFIG=/scratch/Projects/SPIRou_Pipeline/INTROOT/DRS_SPIROU/config/
15:49:54.7 -   ||(dir_calib_db)      DRS_CALIB_DB=/scratch/Projects/SPIRou_Pipeline/data/calibDB
15:49:54.7 -   ||(dir_data_msg)      DRS_DATA_MSG=/scratch/Projects/SPIRou_Pipeline/data/msg/
15:49:54.7 -   ||(print_log)         DRS_LOG=1         %(0: minimum stdin-out logs)
15:49:54.7 -   ||(plot_graph)        DRS_PLOT=1            %(def/undef/trigger)
15:49:54.7 -   ||(used_date)         DRS_USED_DATE=undefined
15:49:54.7 -   ||(working_dir)       DRS_DATA_WORKING=/scratch/Projects/SPIRou_Pipeline/data/tmp/
15:49:54.7 -   ||                    DRS_INTERACTIVE is  set
15:49:54.7 -   |cal_BADPIX_spirou:+[...]|Now running : cal_BADPIX_spirou on file(s):  flat_flat02f10.fits
     dark_dark02d406.fits
15:49:54.7 -   |cal_BADPIX_spirou:+[...]|On directory /scratch/Projects/SPIRou_Pipeline/data/raw/20170710
15:49:54.7 -   |cal_BADPIX_spirou:+[...]|ICDP loaded from: /scratch/Projects/SPIRou_Pipeline/INTROOT/DRS_SPIROU/
     config/hadmrICDP_SPIROU.py
15:49:54.7 - * |cal_BADPIX_spirou:+[...]|Now processing Image TYPE FLAT with cal_BADPIX_spirou recipe
15:49:54.7 - * |cal_BADPIX_spirou:+[...]|Now processing Image TYPE DARK with cal_BADPIX_spirou recipe
15:49:54.7 -   |cal_BADPIX_spirou:+[...]|Reading Flat Image /scratch/Projects/SPIRou_Pipeline/data/raw/20170710/
     flat_flat02f10.fits
15:49:54.7 -   |cal_BADPIX_spirou:+[...]|Flat Image 2048x2048 loaded
15:49:54.7 -   |cal_BADPIX_spirou:+[...]|Reading Dark Image /scratch/Projects/SPIRou_Pipeline/data/raw/20170710/
     flat_flat02f10.fits
15:49:54.8 -   |cal_BADPIX_spirou:+[...]|Dark Image 2048x2048 loaded
```

```
15:49:54.8 -   |cal_BADPIX_spirou:+[...]|Normalising the flat image
15:49:55.9 -   |cal_BADPIX_spirou:+[...]|Looking for bad pixels
15:49:56.9 -   |cal_BADPIX_spirou:+[...]|Fraction of hot pixels from dark: 3.01 %
15:49:56.9 -   |cal_BADPIX_spirou:+[...]|Fraction of bad pixels from flat: 1.32 %
15:49:56.9 -   |cal_BADPIX_spirou:+[...]|Fraction of NaN pixels in dark: 20.76 %
15:49:56.9 -   |cal_BADPIX_spirou:+[...]|Fraction of NaN pixels in flat: 14.66 %
15:49:56.9 -   |cal_BADPIX_spirou:+[...]|Fraction of bad pixels with all criteria: 24.66 %
15:49:56.9 - * |cal_BADPIX_spirou:+[...]|QUALITY CONTROL SUCCESSFUL - Well Done -
15:49:56.9 -   |cal_BADPIX_spirou:+[...]|Saving Bad Pixel Mask in flat_flat02f10_badpixel.fits
15:49:57.5 - * |cal_BADPIX_spirou:+[...]|Updating Calib Data Base with BADPIX
15:49:57.5 - * |cal_BADPIX_spirou:+[...]|Recipe cal_BADPIX_spirou has been succesfully completed
```

6.12 The cal_CCF_E2DS_spirou recipe

Cross correlation function computation on a E2DS spectra.

File suffixes allowed:

- *e2ds_AB.fits

6.12.1 Summary of procedure

1.

6.12.2 Running cal_DRIFT_RAW_spirou

To run cal_DRIFT_RAW_spirou type:

```
cal_CCF_E2DS_spirou.py  night_repository  *e2ds_AB.fits  mask.mas  RV  RANGE  STEP
```

Note: First argument is a file with suffix *e2ds_AB.fits Note: The mask (Ex : UrNe.mas) is presently loaded locally (in the working directory) Note: RV, RANGE, STEP are the central velocity, half range in velocity and step in km/s of the CCF

6.13 Currently unused

6.13.1 cal_loc_ONE_spirou

White lamp exposures on fiber science only (A and B), on fiber C only, or on all of them (ABC). These exposures are used for order localisation and order profil (the pipeline will also need a Fabry-Perot exposure for this last point).

6.13.1.1 Running cal_loc_ONE_spirou

To run cal_loc_ONE_spirou type:

```
cal_loc_ONE_spirou.py  night_repository  filename
```

6.13.2 cal_FF_spirou

A sequence of N white lamp exposures where the two fibres are simultaneously illuminated. This sequence is used by the data reduction pipeline for producing a spectral "master flat-field", monitor the ageing of detector (IR detector loose pixels during its life), monitor the fiber transmission, produce localisation and order profil (the pipeline will also need of exposure « Fabry-Perot » for this last point).

6.13.2.1 Running cal_FF_spirou

To run cal_FF_spirou type:

```
cal_FF_spirou.py  night_repository  filename1 filename2 ... filenameN
```

6.13.3 cal_WAVE_spirou

Hollow Cathode lamp exposures in which the three fibres are simultaneously fed by light from the Hollow Cathode lamps. DRS use each exposure to build a wavelength solution. The instrumental drift with respect to the previous calibration frames is measured.

6.13.3.1 Running cal_WAVE_spirou

To run cal_WAVE_spiroutype:

```
cal_WAVE_spirou.py  night_repository  filename
```

6.13.4 cal_FP_spirou

Fabry-Perot exposures in which the three fibres are simultaneously fed by light from the Fabry-Perot filter. Each exposure is used to build a wavelength solution and the slit orientation. The instrumental drift with respect to the previous calibration frames is measured. A « super » wavelength solution will be builded in using Hollow Cathode and Fabry-Perot exposures; Hollow Cathode exposures give an absolute reference and Fabry-Perot exposures provide a signal very regularly spaced in wavelength.

6.13.4.1 Running cal_FP_spirou

To run cal_FP_spirou type:

```
cal_FP_spirou.py  night_repository  filename
```

Chapter 7

The Modules

Some text here

7.1 The Module directory

This contains all the python worker files for the DRS. The file structure is as below.

```
{DIR}
└── {INSTALL_ROOT}
    └── DRS_SPIROU
        └── python
            └── Modules
                ├── hadgtVISU
                │   └── ..........................................hadmrVISU module files
                ├── hadmrBACK
                │   └── ..........................................hadmrBACK module files
                ├── hadmrBIAS
                │   └── ..........................................hadmrBIAS module files
                ├── hadmrCDB
                │   └── ..........................................hadmrCDB module files
                ├── hadmrDARK
                │   └── ..........................................hadmrDARK module files
                ├── hadmrEXTOR
                │   └── ..........................................hadmrEXTOR module files
                ├── hadmrFITS
                │   └── ..........................................hadmrFITS module files
                ├── hadmrFLAT
                │   └── ..........................................hadmrFLAT module files
                ├── hadmrLED
                │   └── ..........................................hadmrLED module files
                ├── hadmrLOCOR
                │   └── ..........................................hadmrLOCOR module files
                ├── hadmrMATH
                │   └── ..........................................hadmrMATH module files
                ├── hadmrRV
                │   └── ..........................................hadmrRV module files
                ├── hadmrSIMU
                │   └── ..........................................hadmrSIMU module files
                ├── hadmrTHORCA
                │   └── ..........................................hadmrTHORCA module files
                └── hadrgdCONFIG
                    └── ..........................................hadrgdCONFIG module files
```

```
├── spirouBACK.py
├── spirouBIAS.py
├── spirouCDB.py
├── spirouEXTOR.py
├── spirouFITS.py
├── spirouFLAT.py
├── spirouLOCOR.py
├── spirouRV.py
├── spirouTHORCA.py
└── spirouVISU.py
```

7.2 The spirouBACK module

Contains functions to calculate the detector background

7.2.1 measure_bkgr

```
measure_bkgr(data, order_profile, size, ccdgain, ccdsigdet)
    """
    Measures the background via interpolation over many small boxes of
         width/height="size" uses the order_profile to mask out (?) the orders

    :param data: 2D numpy array, the image to measure the background of
    :param order_profile: 2D numpy array, the smoothed image using the order
                          fits
    :param size: int, size of the sub-frame to measure the background in
    :param ccdgain: float, the gain of the image
    :param ccdsigdet: float, the sigdet of the image

    :return bkgr: 2D numpy array, the interpolated background image
    :return xc: numpy array (size x data.shape[0]) in steps of 2*size
    :return yc: numpy array (size x data.shape[1]) in steps of 2*size
    :return mode: numpy array (size x size) the mode (?) of each box
    """
```

Note: Not 100% sure how this function works.

7.2.2 measure_bkgr2

```
measure_bkgr2(data, order_profile, size, ccdgain, ccdsigdet)
    """
    Measures the background via interpolation over many small boxes of
         width/height="size" uses the order_profile to mask out (?) the orders

    :param data: 2D numpy array, the image to measure the background of
    :param order_profile: 2D numpy array, the smoothed image using the order
                          fits
    :param size: int, size of the sub-frame to measure the background in
    :param ccdgain: float, the gain of the image
    :param ccdsigdet: float, the sigdet of the image

    :return bkgr: 2D numpy array, the interpolated background image
    :return xc: numpy array (size x data.shape[0]) in steps of 2*size
    :return yc: numpy array (size x data.shape[1]) in steps of 2*size
    :return mode: numpy array (size x size) the mode (?) of each box
    :return binc: numpy array, the bins the histogram is binned along
    :return histo: numpy array, the histogram of each subframe
    """
```

Note: Not 100% sure how this function works.

7.2.3 measure_bkgr_FF

```
measure_bkgr_FF(data, size, ccdgain, ccdsigdet)
    """
    Measures the background via interpolation over many small boxes of
         width/height="size"

    :param data: 2D numpy array, the image to measure the background of
    :param size: int, size of the sub-frame to measure the background in
    :param ccdgain: float, the gain of the image
    :param ccdsigdet: float, the sigdet of the image

    :return bkgr: 2D numpy array, the interpolated background image
    :return xc: numpy array (size x data.shape[0]) in steps of 2*size
    :return yc: numpy array (size x data.shape[1]) in steps of 2*size
    :return minlevel: numpy array (size x size) - the central value of each
                      subframed box
    """
```

7.2.4 measure_bkgr_FF2

```
measure_bkgr_FF2(data, size, ccdgain, ccdsigdet, seuil)
    """
    Measures the background via interpolation over many small boxes of
         width/height="size" uses a threshold="seuil" above which the value is
         set to the max value in a box

    :param data: 2D numpy array, the image to measure the background of
    :param size: int, size of the sub-frame to measure the background in
    :param ccdgain: float, the gain of the image
    :param ccdsigdet: float, the sigdet of the image
    "param seuil: float, a threshold for

    :return bkgr: 2D numpy array, the interpolated background image
    :return xc: numpy array (size x data.shape[0]) in steps of 2*size
    :return yc: numpy array (size x data.shape[1]) in steps of 2*size
    :return minlevel: numpy array (size x size) - the central value of each
                      subframed box
    """
```

7.3 The spirouCDB module

Contains functions for the calibrations database (infos for master_calib.txt, and to coy files in the caliDB directory)

7.3.1 filename2tunix

```
filename2tunix(cfht_name)
    """
    Open the fits file "cfht_name" read the header key 'DATEEND' and return
    it as a unix timestamp

    :param cfht_name: string, the fits filename and location of file
    :return tt: float, the unix time of "cfht_name" store in 'DATEEND'
    """
```

Note: Currently not used in code

7.3.2 filename2tunix2

```
filename2tunix2(cfht_name)
    """
    Open the fits file "cfht_name" read the header key 'ACQTIME1' and return
    it as a unix timestamp

    :param cfht_name: string, the fits filename and location of file
    :return tt: float, the unix time of "cfht_name" store in 'ACQTIME1'
    """
```

7.3.3 filename2tuni_old

```
filename2tunix_old(file_name)
    """
    Extracts the timestamp information from a filename and converts it to a
        unix timestamp

    :param file_name: string, filename toe look for timestamp in, must contain
                      *.YY-MM-DDThh:mm:ss_*  or *.YY-MM-DDThh:mm:ss.fits
                      i.e. filename.17-06-12T15:32:42

    :return tt: float, the unix time from file_name
    """
```

Note: Currently not used in code

7.3.4 data2tunix

```
date2tunix(argdate)
    """
    Converts a string date in format YY-MM-DD into a unix timestamp

    :param argdate: string in form YY-MM-DD

    :return tt: float, the unix time from argdate
    """
```

Note: Is this used?

7.3.5 update_master

```
update_master(master_file, night, key, file_name, opt_cmt)
    """
    Update calibration database with new row in form:
```

```
    {key} {night respoitory} {filename} {human readable time} {unix time}

    Note calibration database is locked while this function is active
    Note time comes from file_name so must have header 'ACQTIME1'

    :param master_file: string, the master file to open
    :param night: string, the night repository in form YYMMDD
    :param key: string, the key for this database entry (e.g. 'DARK')
    :param file_name: string, the filename for this database entry
    :param opt_cmt: string, the program name (for logging)
    :return None:
    """
```

7.3.6 update_master_onlast

```
update_master_onlast(master_file, night, key, file_name, opt_cmt)
    """
    Update calibration database with new row in form:

    {key} {night respoitory} {filename} {human readable time} {unix time}

    Note calibration database is locked while this function is active
    Note time comes from file_name so must have header 'ACQTIME1'

    :param master_file: string, the master file to open
    :param night: string, the night repository in form YYMMDD
    :param key: string, the key for this database entry (e.g. 'DARK')
    :param file_name: string, the filename for this database entry
    :param opt_cmt: string, the program name (for logging)
    :return None:
    """
```

Note: exactly the same as update_master?

7.3.7 get_master

```
get_master(master_file, max_time, opt_cmt)
    """
    Gets the latest entries (less than max_time) for all unique keys
        (if multiple keys exist latest less than max_time is used)

    calibration database must be in the form

    {key} {night respoitory} {filename} {human readable time} {unix time}

    :param master_file: string, the calibration database file
    :param max_time: float, the unix time to use as the latest date than can be
                     used for calibration files
    :param opt_cmt: string, the program name (for logging)
    :return C_db: dictionary, dictionary database containing key value pairs

                where the C_db[key] = [night repository, calibration filename]
    """
```

7.3.8 get_early_last_master

```
get_early_last_master(master_file, opt_cmt)
    """
    Gets the earliest and latest entries for all unique keys
        (if multiple keys exist earliest and latest are used)

    calibration database must be in the form

    {key} {night respoitory} {filename} {human readable time} {unix time}

    :param master_file: string, the calibration database file
    :param opt_cmt: string, the program name (for logging)
```

```
    :return C_db_early: dictionary, dictionary database containing key value
                        pairs for the earliest entry of each key
    :return C_db_last: dictionary, dictionary database containing key value
                        pairs for the latest entry of each key

                where the C_db[key] = [night repository, calibration filename]
    """
```

7.3.9 cp_db_files

```
cp_db_files(calib_db_dir, reduc_dir, db_dict, opt_cmt)
    """
    Copies the calibration file if it doesn't already exist in the
        reduced directory

    :param calib_db_dir: string, the calibration database directory
    :param reduc_dir: string, the reduced directory
    :param db_dict: dictionary, the calibration database dictionary
    :param opt_cmt: string, the program name (for logging)

    :return None:
    """
```

7.3.10 put_files

```
put_file(calib_db_dir, file, opt_cmt)
    """
    Put a calibration file in the calibration database directory

    :param calib_db_dir: string, the calibration directory
    :param file: string, the calibration file to put in calibration directory
    :param opt_cmt: string, the program name (for logging)

    :return None:
    """
```

7.4 The spirouEXTOR module

Contains function for the ordres extraction

```
readkeyloco(fitsfilename, kw, nbo, nbc)

cosmic_filter(spe, window_size=128, sigma_clip=4.5)

extract0(data, pos, sig, plage, ccdgain)

extract_weigth(data, pos, sig, plage, order_profil, ccdgain, sigdet)

extract_tilt_weigth(data, pos, sig, tilt, plage, order_profil, ccdgain,
                    sigdet)

extract_weigth2(data, pos, sig, plage1, plage2, order_profil, ccdgain,
                sigdet)

extract_tilt_weigth2(data, pos, sig, tilt, plage1, plage2, order_profil,
                     ccdgain, sigdet)

extract_horne(data, pos, sig, plage, order_profil, ccdgain, sigdet)

extract_tilt_horne(data, pos, sig, tilt, plage, order_profil, ccdgain,
                   sigdet)

extract_tilt(data, pos, sig, tilt, plage, ccdgain)

extract(data, pos, sig, opt, nbsig, ccdgain, sigdet, seuilcosmic)
```

7.5 The spirouFITS module

Contains functions to read FITS file and copy keywords

7.5.1 read_data

```
read_data(fitsfilename)
    """
    Reads "fitsfilename" using pyfits and returns array containing the data
    only uses data from the first header

    :param fitsfilename: string, file location and file name

    :return data: numpy array (2D), the image
    :return nx: int, the shape in the first dimension, i.e. data.shape[0]
    :return ny: int, the shape in the second dimension, i.e. data.shape[1]
    """
```

Note: currently not used?

7.5.2 read_data_raw

```
read_data_raw(fitsfilename)
    """
    Reads "fitsfilename" using pyfits and returns array containing the data

    If there is one extension it is used
    If there are two extensions the second is used

    :param fitsfilename: string, file location and file name

    :return data: numpy array (2D), the image
    :return nx: int, the shape in the first dimension, i.e. data.shape[0]
    :return ny: int, the shape in the second dimension, i.e. data.shape[1]
    """
```

Note: Essentially the same as 'read_data' function

7.5.3 read_data_all

```
read_data_all(fitsfilename)
    """
    Reads "fitsfilename" using pyfits and returns array containing the data

    If there is one extension it is used
    If there are two extensions the second is used
    If there are multiple extensions (1 or 2 extensions) the are joined in form:

    x = [[3, 3, 3],
         [3, 3, 3],
         [3, 3, 3]])

    y = [[5, 5, 5],
         [5, 5, 5],
         [5, 5, 5]])

    returned array is = [[3, 3, 3, 5, 5, 5],
                         [3, 3, 3, 5, 5, 5],
                         [3, 3, 3, 5, 5, 5]])

    :param fitsfilename: string, file location and file name

    :return data: numpy array (2D), the image
    :return nx: int, the shape in the first dimension, i.e. data.shape[0]
    :return ny: int, the shape in the second dimension, i.e. data.shape[1]
    """
```

7.5.4 write_data

```
write_data(fitsfilename, data)
    """
    Create the FITS file and write data

    :param filename: string, filename to save the fits file to
    :param data: numpy array (2D), the image

    :return None:
    """
```


7.5.5 read_ext

```
read_ext(fitsfilename)
    """
    Opens the FITS file and reads the number of extensions

    :param fitsfilename: string, filename to read
    :return length: the length of the extention -1
    """
```

7.5.6 read_key

```
read_key(fitsfilename, keyname, hdu=0)
    """
    Reads the "keyname" from the header of "fitsfilename" and returns
    the value of the "keyname"

    :param fitsfilename: string, filename to read
    :param keyname: string, the keyword to look for in the header,
                    returns 'UNKNOWN' if not found
    :param hdu: int, the extention to read the header from (default = 0)

    :return None:
    """
```

7.5.7 read_keys

```
read_keys(fitsfilename, keylist, hdu=0)
    """
    Reads the "keyname" from the header of "fitsfilename" and returns
    the value of the "keyname"

    :param fitsfilename: string, filename to read
    :param keylist: string, the keyword to look for in the header,
                    returns 'UNKNOWN' if not found
    :param hdu: int, the extention to read the header from (default = 0)

    :return None:
```

Note: currently identical to 'read_key'

7.5.8 write_newkey

```
write_newkey(fitsfilename, key)
    """
    Writes a new keyword "key[0]" to the FITS file header (extension = 0)
    :param fitsfilename: string, filename to write to
    :param key: list of string, the key to write, must be in form:

            key = [KEYWORD NAME, VALUE, COMMENT]

    :return None:
    """
```

7.5.9 update_key

```
update_key(fitsfilename, key)
    """
    Modifies a keyword "key[0]" in the FITS file header (extension = 0)

    :param fitsfilename: string, filename to read/write to
    :param key: string, the key to write, must be in form:

            key = [KEYWORD NAME, VALUE, COMMENT]

    :return None :
    """
```

7.5.10 delete_key

```
delete_key(fitsfilename, keyname)
    """
    Deletes a keyword "keyname" in the FITS file header (extension = 0)

    :param fitsfilename: string, filename to read/write to
    :param keyname: string, the key to write, must be in form:

    :return None:
    """
```

7.5.11 writekey2dlist

```
writekey2dlist(fitsfilename, a, kw)
    """
    Writes a 2D list of values (in "a") to FITs file "fitsfilename" with
    keyword = "kw" + number

    where number = (row number * number of columns) + column number

    :param fitsfilename: string, filename to read/write to
    :param a: 2D list of values, the values to write
    :param kw: string, the prefix of the key to write, key will be in form:
                "kw" + number

    :return None:
    """
```

7.5.12 copy_key

```
copy_key(fits1, fits2)
    """
    Writes the header keys from one FITS fits (fits1) to another (fits2)

    Does not include FITS keywords:

            'SIMPLE', 'BITPIX', 'NAXIS', 'NAXIS1', 'NAXIS2',
            'EXTEND', 'COMMENT', 'CRVAL1', 'CRPIX1', 'CDELT1',
            'CRVAL2', 'CRPIX2', 'CDELT2', 'BSCALE', 'BZERO',
            'PHOT_IM', 'FRAC_OBJ', 'FRAC_SKY', 'FRAC_BB'

    :param fits1: string, filename to read from
    :param fits2: string, filename to write to

    :return None:
    """
```

7.5.13 copy_keys_root

```
copy_keys_root(fits1, fits2, root)
    """
    Copies all keywords with prefix "root" from "fits1" to "fits2"

    :param fits1: string, filename to read from
    :param fits2: string, filename to write to
    :param root: string, prefix - start of the keyword to read/write to

    :return None:
    """
```


7.5.14 readkeyloco

```
readkeyloco(fitsfilename, kw, nbo, nbc)
    """
    Read a key from generated from a 2D list

    :param fitsfilename: string, filename to read/write to
    :param kw: string, the prefix of the key to write, key will be in form:
                "kw" + number

                where number = (row number * number of columns) + column number

    :param nbo: int, the row number
    :param nbc: int, the columns number

    :return None:
    """
```

7.6 The spirouLOCOR module

Contains functions for order localization

7.6.1 meas_back

```
meas_back(y, backini, niter=3, coeff=9)
    """
    Measure and subtract the background from a column, iterative background fit
    by a polynomial and cut off high values

    :param y: 1D numpy array, column to use for background measurement
    :param backini: float, cut level
    :param niter: int, number of fits to perform
    :param coeff: int, polynomial order to fit
    :return yc: 1D numpy array, the image column with background subtracted
    :return f: 1D numpy array, the background fit data
    """
```

Note: Currently not used? Note: required FORTRAN 'fitpoly' function

7.6.2 meas_top

```
meas_top(yc, nbslice, nbpix)
    """
    Measure and normalize the top level of a column, seach for maximum on
    "nbslice" slices of "nbpix" pixels. Should consist of 1 or 2 orders. A
    polynomial fit of order 9 is then divided through the original column

    :param yc: 1D numpy array, the image column
    :param nbslice: int, the number slices to use?
    :param nbpix: int, the number of pixels to use?

    :return ycc: 1D numpy array, the image column normalised by the fit
    :return f: 1D numpy array, the background fit data
    """
```

Note: Not sure exactly what this function is doing Note: Currently not used? Note: required FORTRAN 'fitpoly' function

7.6.3 meas_minmax

```
meas_minmax(y, size)
    """
    Measure the minimum and maximum pixel value for each row using a box which
    contains all pixels for rows:  row-size to row+size and all columns.

    Edge pixels (0-->size and (image.shape[0]-size)-->image.shape[0] are
    set to the values for row=size and row=(image.shape[0]-size)

    :param y: numpy array (2D), the image
    :param size: int, the half size of the box to use (half height)
                 so box is defined from  row-size to row+size

    :return min_image: numpy array (1D length = image.shape[0]), the row values
                       for minimum pixel defined by a box of row-size to
                       row+size for all columns
    :retrun max_image: numpy array (1D length = image.shape[0]), the row values
                       for maximum pixel defined by a box of row-size to
                       row+size for all columns
    """
```

7.6.4 meas_min

```
meas_min(yc, nbslice, nbpix)
    """
    Measure and subtract the minimum level of a column, search for minimum on
    "nbslice" slices of "nbpix" pixels. Should consist of 1 or 2 orders. A
    polynomial fit of order 9 is then subtracted from the original column

    :param yc: 1D numpy array, the image column
    :param nbslice: int, the number slices to use?
    :param nbpix: int, the number of pixels to use?

    :return ycc: 1D numpy array, the image column subtracted by the fit
    :return f: 1D numpy array, the background fit data
    """
```

Note: Not sure exactly what this function is doing Note: Currently not used? Note: required FORTRAN 'fitpoly' function

7.6.5 poscolc

```
poscolc(ycc, seuil)
    """
    Takes the central pixel values and finds the order centers by looking for
    the start and end above threshold

    :param ycc: numpy array (1D) size = number of rows,
                the central pixel values
    :param seuil: float, the threshold above which to find pixels as being
                  part of an order

    :return positions: numpy array (1D), size= number of rows,
                       the pixel positions in cvalues where the centers of each
                       order should be
    """
```

7.6.6 findord

```
findord(ycc, seuil, widthmin=5)
    """
    Takes a set of pixel values and finds the order center and width by
    looking for the start and end above threshold "seuil"

    :param ycc: numpy array (1D) size = number of rows,
                an array of pixel values (across one order)
    :param seuil: float, the threshold above which to find pixels as being
                  part of an order
    :param widthmin" int, the minimum width of an order to be a valid, accepted
                    order

    :return position: float, the central pixel positions in ycc where the
                      centers of this order should be
    :return width: float, the width in pixels of this order
    """
```

7.6.7 locgaus

```
locgaus(y, ind1, ind2, sigdet, opt=1)
    """
    Find the position and the FWHM of one order using a gaussian using
    weights

    Non-Linear Fit of a Gaussian with 4 Parameters by the Method
    from Levenberg-Marquardt. Uses the subroutines MRQMIN, MRQCOF,
    COVSRT and GAUSSJ.

    :param y: 1D numpy array, the pixel values between ind1 and ind2
    :param ind1: float, the start point of this order
    :param ind2: float, the end point of this order
    :param sigdet: float, the ccdsigdet for the image
```

```
    :param opt: int, whether to use weights
                if opt=1 weights are used in the form:
                    weights = 1/(sqrt(y + sigdet^2))
                if opt=2 weights are used in the form:
                    weights = y

    :return center: float, the center of the order from the fit
                    (set to zero if difference in y is less than 20*sigdet)
    :return fity: 1D numpy array, size=image size, the gaussian fit for y
    :return sigma: float, the FWHM of the order from the fit
                    (set to zero if difference in y is less than 20*sigdet)
    """
```

Note: Currently not used? Note: required FORTRAN 'fitgaus' function

7.6.8 loc2gaus

```
loc2gaus(y, ind1, ind2, delta, sigdet, opt=1)
    """
    Find the position and the FWHM of one order using a two gaussian model with
    fixed widths (delta) using weights

    Non-Linear Fit of a Gaussian with 4 Parameters by the Method
    from Levenberg-Marquardt. Uses the subroutines MRQMIN, MRQCOF,
    COVSRT and GAUSSJ.

    :param y: 1D numpy array, the pixel values between ind1 and ind2
    :param ind1: float, the start point of this order
    :param ind2: float, the end point of this order
    :param delta: int
    :param sigdet: float, the ccdsigdet for the image
    :param opt: int, whether to use weights
                if opt=1 weights are used in the form:
                    weights = 1/(sqrt(y + sigdet^2))
                if opt=2 weights are used in the form:
                    weights = y

    :return center: float, the center of the order from the fit
                    (set to zero if difference in y is less than 20*sigdet)
    :return fity: 1D numpy array, size=image size, the gaussian fit for y
    :return fwhm: float, the FWHM of the order from the fit
                    (set to zero if difference in y is less than 20*sigdet)
    :return profile_error: float, the standard deviation of the residuals
                           between fit and data
    """
```

Note: Currently not used? Note: required FORTRAN 'fit2gaus' function

7.6.9 fitprofil

```
fitprofil(y, sigdet)
    """
    Determine the profile of one order using a Gaussian model
    (weights = 1/sqrt((y + sigdet^2)/max(y)))

    :param y: 1D numpy array, the pixel values of one order
    :param sigdet: float, the ccdsigdet for the image

    :return fw: float, the FWHM?
    :return fity: 1D numpy array, the gaussian fit to the data
    """
```

Note: Currently not used? Note: required FORTRAN 'fitgaus' function

7.6.10 imaloco

```
imaloco(data, ac, fitsfilename)
    """
    Creates and save the image with the superposistion of the center of orders
```

```
    (set to zero)

    :param data: numpy array (2D), the image
    :param ac: numpy array (2D), size=(number of orders x polynomial fit level)
               the coefficient matrix (coefficients for all orders
    :param fitsfilename: string, file location and file name to save the file

    :return None:
    """
```

Note: Currently not used?

7.6.11 imaloco2

```
imaloco2(data, ac, fitsfilename)
    """
    Creates and save the image with the superposistion of the center of orders
    (set to zero)

    :param data: numpy array (2D), the image
    :param ac: numpy array (2D), size=(number of orders x polynomial fit level)
               the coefficient matrix (coefficients for all orders
    :param fitsfilename: string, file location and file name to save the file

    :return None:
    """
```

Note: Exactly the same as imaloco?

7.6.12 tableloco

```
tableloco(tblfilename, a)
```

7.6.13 keyloco

```
keyloco(fitsfilename, ac, ass)
```

7.6.14 keyloco2

```
keyloco2(fitsfilename, ac, ass)
```

7.6.15 writekeyloco3

```
writekeyloco3(fitsfilename, a)
```

7.6.16 writekeyloco

```
writekeyloco(fitsfilename, a, kw)
```

7.6.17 readkeyloco

```
readkeyloco(fitsfilename, ic_locnbmaxo, ic_locdfitc)
```

7.6.18 readkeyloco3

```
readkeyloco3(fitsfilename, rootkeyname)
```

7.6.19 write_fitsloco

```
write_fitsloco(fitsfilename, a, dim)
```

7.7 The spirouRV module

Contains functions to compute radial velocity

7.8 The spirouVISU module

Contains functions for visualization

Chapter 8

Quality Control

8.1 cal_DARK_spirou

There are currently two quality control checks for cal_DARK_spirou

- Unexpected dark level:

$$\text{Median Flux} < \text{qc\_max\_darklevel} \tag{8.1}$$

- Unexpected Fraction of dead pixels:

$$\text{Number of dead pixels} < \text{qc\_maxdead} \tag{8.2}$$

where qc_max_darklevel and qc_maxdead are constants (see Section 5.3).

8.2 cal_loc_RAW_spirou

There are currently five quality control checks for cal_loc_RAW_spirou

- Too many rejected orders in center position fit:

$$\text{Number of rejected orders in center fit} > \text{qc\_loc\_maxlocfit\_removed\_ctr} \tag{8.3}$$

- Too many rejected orders in width fit:

$$\text{Number of rejected orders in width fit} > \text{qc\_loc\_maxlocfit\_removed\_width} \tag{8.4}$$

- RMS on center fit too high:

$$\text{Mean rms center fit} > \text{qc\_loc\_rmsmax\_center} \tag{8.5}$$

- RMS on width fit too high:

$$\text{Mean rms width fit} > \text{qc\_loc\_rmsmax\_fwhm} \tag{8.6}$$

- Abnormal number of identified orders:

$$\text{Number of orders found} \neq \text{qc\_loc\_nbo} \tag{8.7}$$

where c_loc_maxlocfit_removed_ctr, qc_loc_maxlocfit_removed_width, qc_loc_rmsmax_center, qc_loc_rmsmax_fwhm and qc_loc_nbo are constants (see Section 5.3).

8.3 cal_SLIT_spirou

No quality control performed yet.

8.4 cal_FF_RAW_spirou

There is currently one quality control check for cal_FF_RAW_spirou

- Too much flux in the image:

$$\text{maximum signal} > \text{qc\_max\_signal} * nbframes \tag{8.8}$$

  Note: This check does not currently lead to a failed run and all files are processed as passing quality checks

where qc_max_signal is a constant defined in Section 5.3.

8.5 cal_extract_RAW_spirou (AB, C and ALL)

There is currently one quality control check for cal_FF_RAW_spirou

- Too much flux in the image:

$$\text{maximum signal} > \text{qc\_max\_signal} * nbframes \tag{8.9}$$

  Note: This check does not currently lead to a failed run and all files are processed as passing quality checks

where qc_max_signal is a constant defined in Section 5.3.

8.6 cal_DRIFT_RAW_spirou

No quality control performed yet.

8.7 cal_HC_E2DS_spirou

Section needs writing

8.8 cal_DRIFT-PEAK_E2DS_spirou

Section needs writing

8.9 cal_WAVE_E2DS_spirou

There are currently three quality control checks for cal_WAVE_E2DS_spirou:

- 'FINAL_xfit_moy' must be a real number

$$\text{FINAL\_xfit\_moy} \neq \infty \text{ and FINAL\_xfit\_moy} \neq \text{NaN} \tag{8.10}$$

- the RMS of each of the three Littrow polynomials fitted across the orders at x=(500,1000,1500) must be lower than qc_rms_littrow_max:

$$\text{rms\_Littrow}_i > \text{qc\_rms\_littrow\_max} \tag{8.11}$$

- the absolute value of their minimum and maximum must be lower than qc_dev_littrow_max for each of the three Littrow polynomials fitted across the orders at x=(500,1000,1500)

$$max(|\text{maxdev\_Littrow}_i|, |\text{mindev\_Littrow}_i|) > \text{qc\_rms\_littrow\_max} \tag{8.12}$$

where qc_rms_littrow_max = 300 m/s and qc_rms_littrow_max = 900 m/s.

8.10 cal_BADPIX_spirou

No quality control performed yet.

8.11 cal_CCF_E2DS_spirou

Section needs writing

Appendix A

Source code for env_setup.sh

Below is the source code for 'env_setup.sh' and should be saved as 'env_setup.sh' and be allowed to execute:

```
chmod +x env_setup.sh
```

```
#!/bin/bash
# alias.sh

# Use --clean to restore environment

# ----------------------------------------------------------------------
# EDIT THESE PARAMETERS
# ----------------------------------------------------------------------
# Set the instrument name
INSTRUMENT_NAME="SPIROU"
# Directory in which all SPIROU files go
DIR="/data/spirou/drs/"
# Define installation folder name
INSTALL_ROOT="$DIR/INTROOT"
# Define data folder name
DATA_ROOT="$DIR/data"
# Define raw path
DATA_ROOT_RAW="$DATA_ROOT/raw/"
# Define reduced path
DATA_ROOT_REDUCED="$DATA_ROOT/reduced/"
# Define calibDB path
DATA_ROOT_CALIB="$DATA_ROOT/calibDB"
# Define msg path
DATA_ROOT_MSG="$DATA_ROOT/msg/"
# Define tmp path
DATA_ROOT_TMP="$DATA_ROOT/tmp/"
# Define whether to run interactive session
INTERACTIVE=1
# Define python version
PYTHON_VERSION="2.7"
# Define python directory (i.e. result of command "which python")
PYTHON_DIR="$DIR/python/miniconda2"
# Define GSL path
GSL_DIR="$DIR/c-libraries/gsl"


# ----------------------------------------------------------------------
# DO NOT EDIT PAST THIS POINT
# ----------------------------------------------------------------------
echo " ======================================== "
echo "  Environmental setup for DRS Pipeline     "
echo " ======================================== "

# get arguments
CLEAN="0"
for i in "$@"; do
  case $i in
    --clean) CLEAN="1";;
    *) echo "Invalid argument"; exit 1;;
  esac
done

if [ "$CLEAN" = "1" ]; then
  if [[ -z "${DRS_ACTIVE}" ]]; then
    if [ "$DRS_ACTIVE" != 1 ]; then
      echo "   No need to clean - environment clean"
      echo ""
```

```
      echo "   Done"
      exit
    fi
  fi
  echo "   Cleaning environment"
  export PATH=$OLDPATH
  export PYTHONPATH=$OLDPYTHONPATH
  unset OLDPATH
  unset DRS_ACTIVE
  unset OLDPYTHONPATH
  unset INTROOT
  unset DRS_LOG
  unset PYTHON_INCLUDE_DIR
  unset GSL_INCLUDE_DIR
  unset GSL_LIBRARY_DIR
  unset INSTRUMENT
  unset DRS_DATA_RAW
  unset DRS_DATA_REDUC
  unset DRS_CALIB_DB
  unset DRS_DATA_MSG
  unset DRS_DATA_WORKING
  unset TDATA
  unset DRS_INTERACTIVE
  unalias conda
  unalias pip
  unalias f2py
else
  echo "   Setting up environment"
  # Backup old path and python path for clean
  OLDPATH=$PATH
  OLDPYTHONPATH=$PYTHONPATH
  export OLDPATH
  export DRS_ACTIVE=1
  export OLDPYTHONPATH

  #User specific environment and startup programs
  export INTROOT="$INSTALL_ROOT"
  export PATH=.:"$INSTALL_ROOT/bin":"$PYTHON_DIR/bin":$PATH
  export PYTHONPATH=.:"$PYTHON_DIR/lib/python$PYTHON_VERSION/site-packages/":"$INSTALL_ROOT/bin/"
  export DRS_LOG=1
  export PYTHON_INCLUDE_DIR="$PYTHON_DIR/lib/python$PYTHON_VERSION/site-packages/numpy/core/include"
  export GSL_INCLUDE_DIR="$GSL_DIR/include"
  export GSL_LIBRARY_DIR="$GSL_DIR/lib"
  # force aliases
  chmod +x "$PYTHON_DIR/bin/conda"
  alias conda="$PYTHON_DIR/bin/conda"
  chmod +x "$PYTHON_DIR/bin/pip"
  alias pip="$PYTHON_DIR/bin/pip"
  chmod +x "$PYTHON_DIR/bin/f2py"
  alias f2py="$PYTHON_DIR/bin/f2py"

  #SPIROU DRS and DATA environment variables
  export INSTRUMENT=$INSTRUMENT_NAME
  export DRS_DATA_RAW=$DATA_ROOT_RAW
  export DRS_DATA_REDUC=$DATA_ROOT_REDUCED
  export DRS_CALIB_DB=$DATA_ROOT_CALIB
  export DRS_DATA_MSG=$DATA_ROOT_MSG
  export DRS_DATA_WORKING=$DATA_ROOT_TMP
  export DRS_INTERACTIVE=$INTERACTIVE
  export TDATA=$DATA_ROOT

  echo "   Set up for $INSTRUMENT"
  echo "    - Python located at: $PYTHON_DIR"
  echo "    - GLS located at: $GSL_DIR"
  echo "    - installation located at: $INSTALL_ROOT"
  echo "    - data located at:"
  echo "           $DATA_ROOT"
  echo "           $DATA_ROOT_RAW"
  echo "           $DATA_ROOT_REDUCED"
  echo "           $DATA_ROOT_CALIB"
  echo "           $DATA_ROOT_MSG"
  echo "           $DATA_ROOT_TMP"
  echo " "

fi
```

```
echo "   Done"
```