Control microscope

2017, Maxime PINSARD

To you, who wants to modify/improve the python code

- Be aware that the main GUI should NOT call an inner function for doing an action that is not instantaneous, it should rather emit a pyQtSignal that is connected to another worker that will perform the action

→ For instance, to home the XY stage (takes 3 secs) : you should tell the APT worker to do it in parallel (so the APT libs and motors should be initialised in this worker)

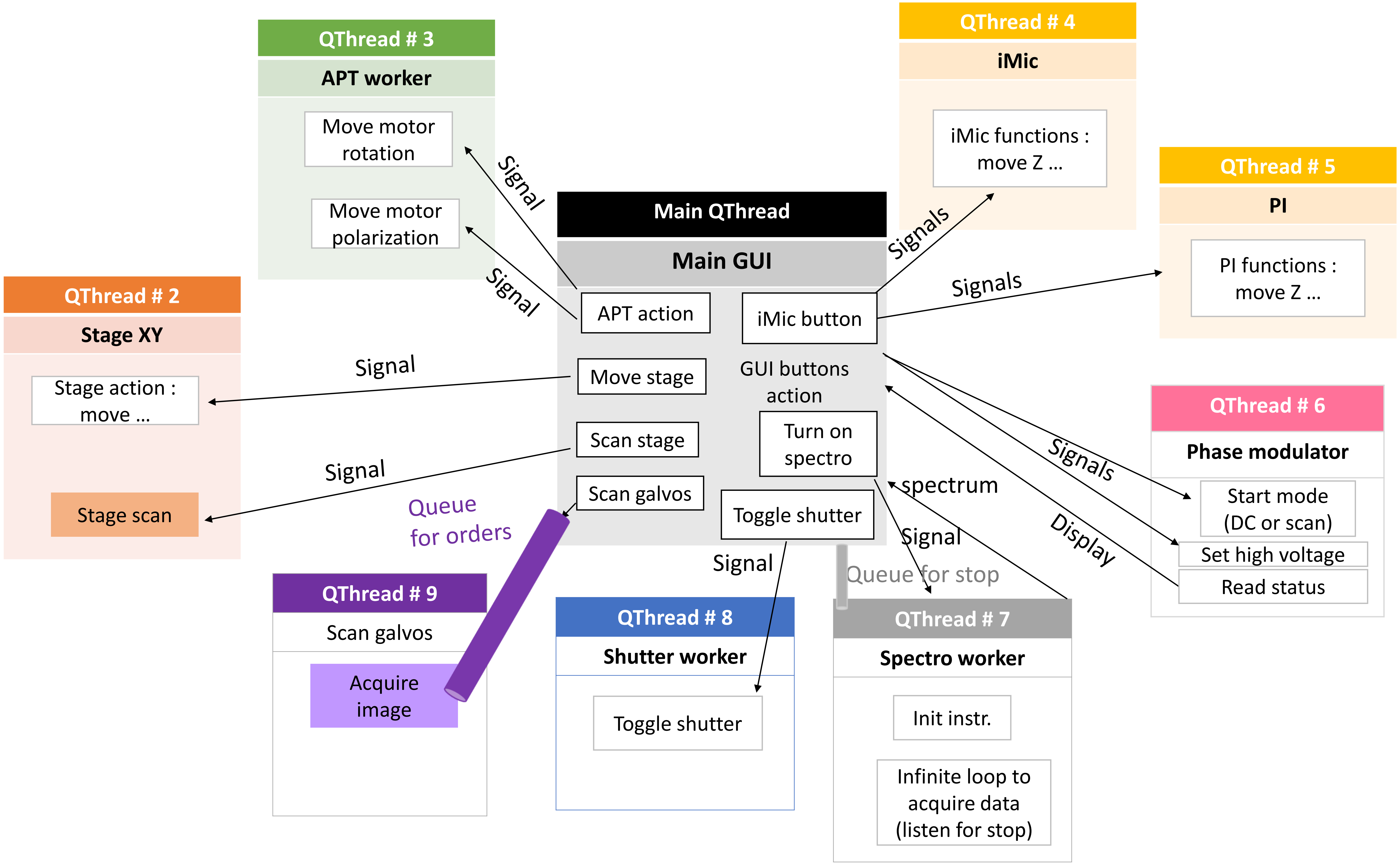
→ By clicking "home button", you should be connected directly to one of the APT worker's method: if you go through a GUI method before, it will wait inside this method until it's completed and will freeze the GUI

- For multiprocess, non 'normal' variable are said 'non-pickable' : you should only pass classic python and numpy objects to the init of processes. This means you can't pass VISA resource, DAQ object nor APT motor nor any library to the init, you rather have to init them inside the process itself by importing the corresponding lib in the 'run'

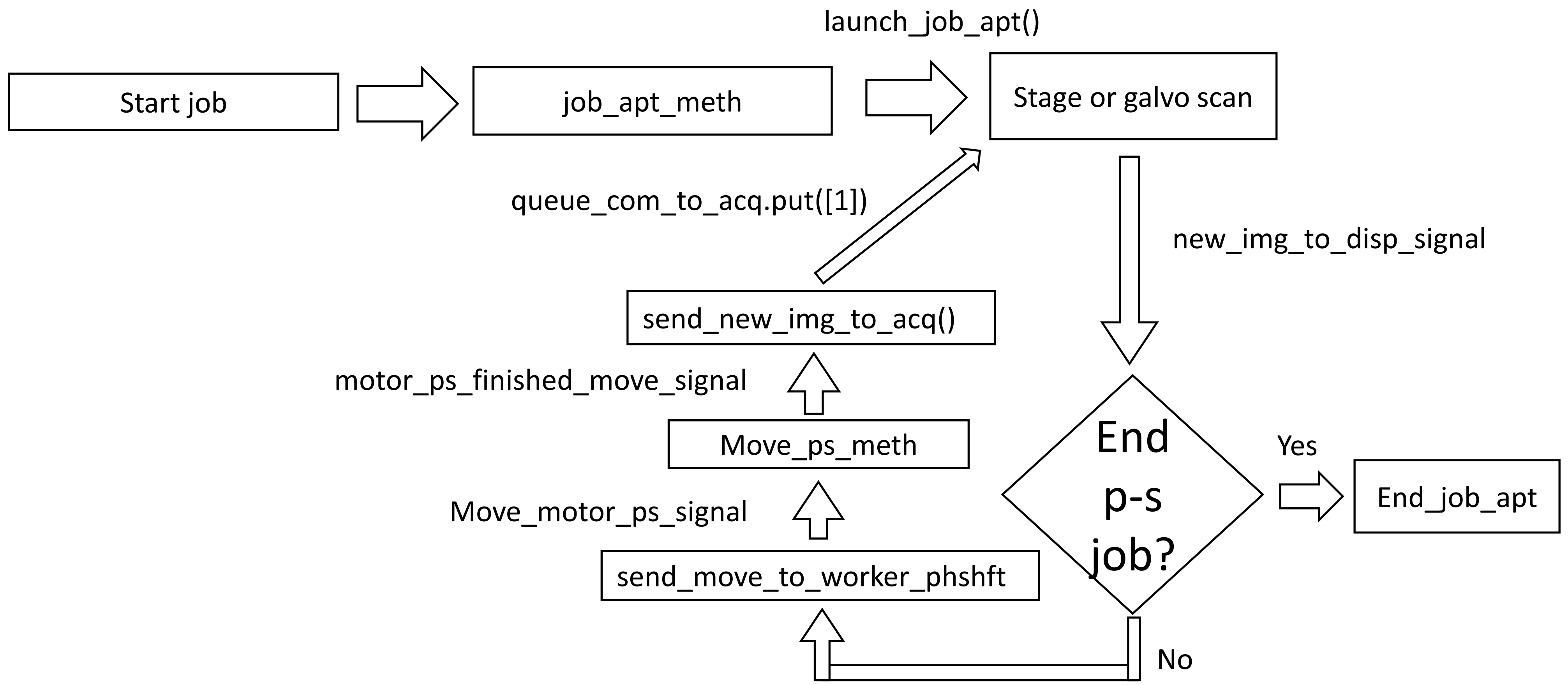
- For efficient use of QThread, you should define a worker in a Class that input only (QObject). Then you define your worker in the GUI by passing to init the var you want through the __init__ method, and you simply call worker_whatever.moveToThread(mythread).

mythread is simply a QThread() that assures that it's indeed in parallel. You can simply call mythread.start() after to start it.

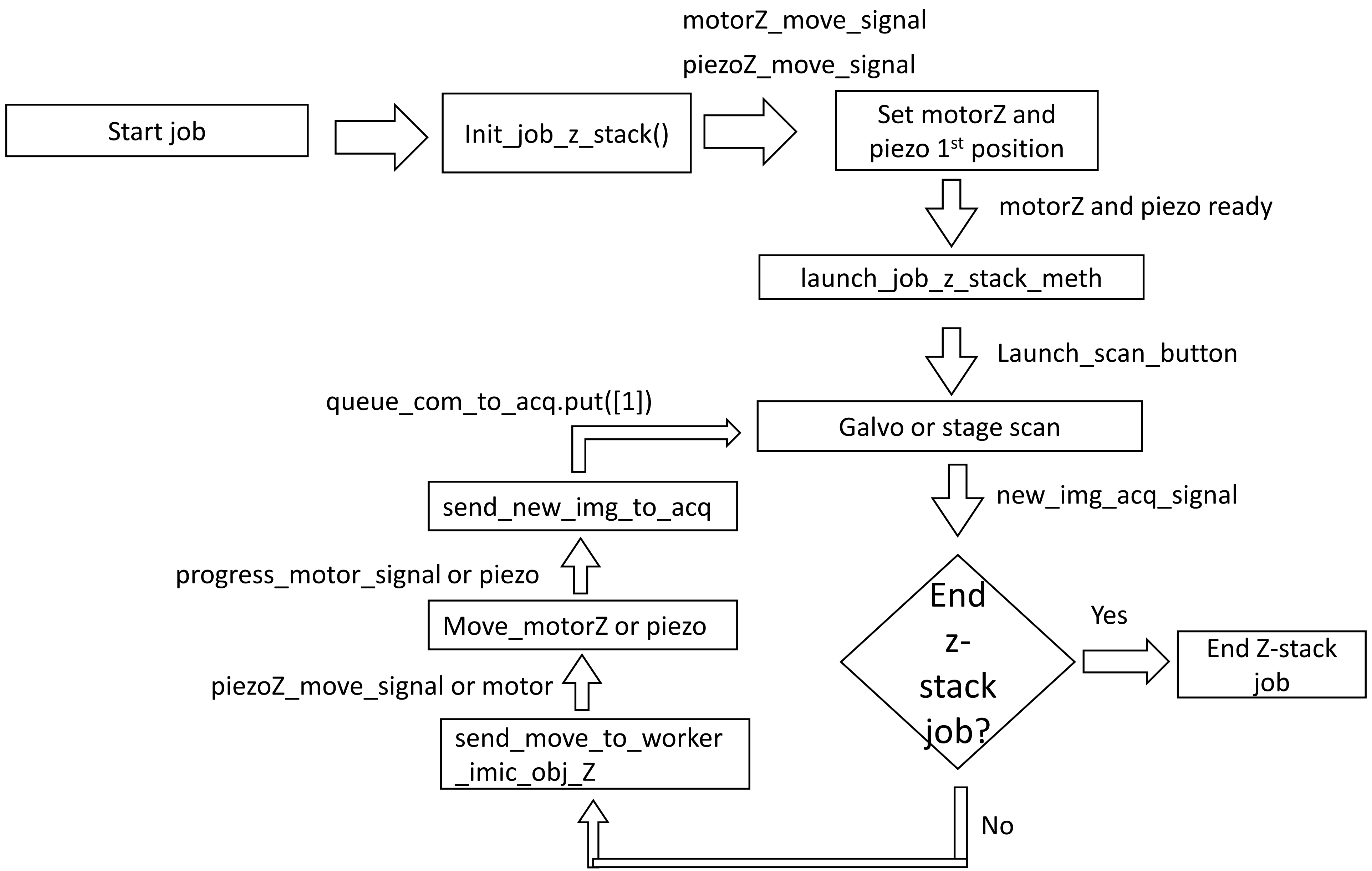
However, a QThread is not a pure parallel process due to GIL. Use Qprocess or multiprocess instead. But it's fine to keep the GUI responsive


QThread # 3
APT worker
Move motor rotation
Move motor polarization
QThread # 4
iMic
iMic functions : move Z …
QThread # 5
PI
PI functions : move Z …
Main QThread
Main GUI
APT action
iMic button
Move stage
GUI buttons action
Scan stage
Turn on spectro
Scan galvos
Toggle shutter
Signal
Signal
Signals
Signals
QThread # 2
Stage XY
Stage action : move …
Stage scan
Signal
Signal
Queue for orders
QThread # 6
Phase modulator
Start mode (DC or scan)
Set high voltage
Read status
Signals
Display
spectrum
Signal
Queue for stop
Signal
QThread # 9
Scan galvos
Acquire image
QThread # 8
Shutter worker
Toggle shutter
QThread # 7
Spectro worker
Init instr.
Infinite loop to acquire data (listen for stop)

The jobs, **Case 01 : phase-shifts**



The jobs, **Case 02 : z-stack**



Start job

Init_job_z_stack()

motorZ_move_signal

piezoZ_move_signal

Set motorZ and piezo 1st position

motorZ and piezo ready

launch_job_z_stack_meth

Launch_scan_button

Galvo or stage scan

new_img_acq_signal

End z-stack job?

Yes

End Z-stack job

No

send_move_to_worker _imic_obj_Z

piezoZ_move_signal or motor

Move_motorZ or piezo

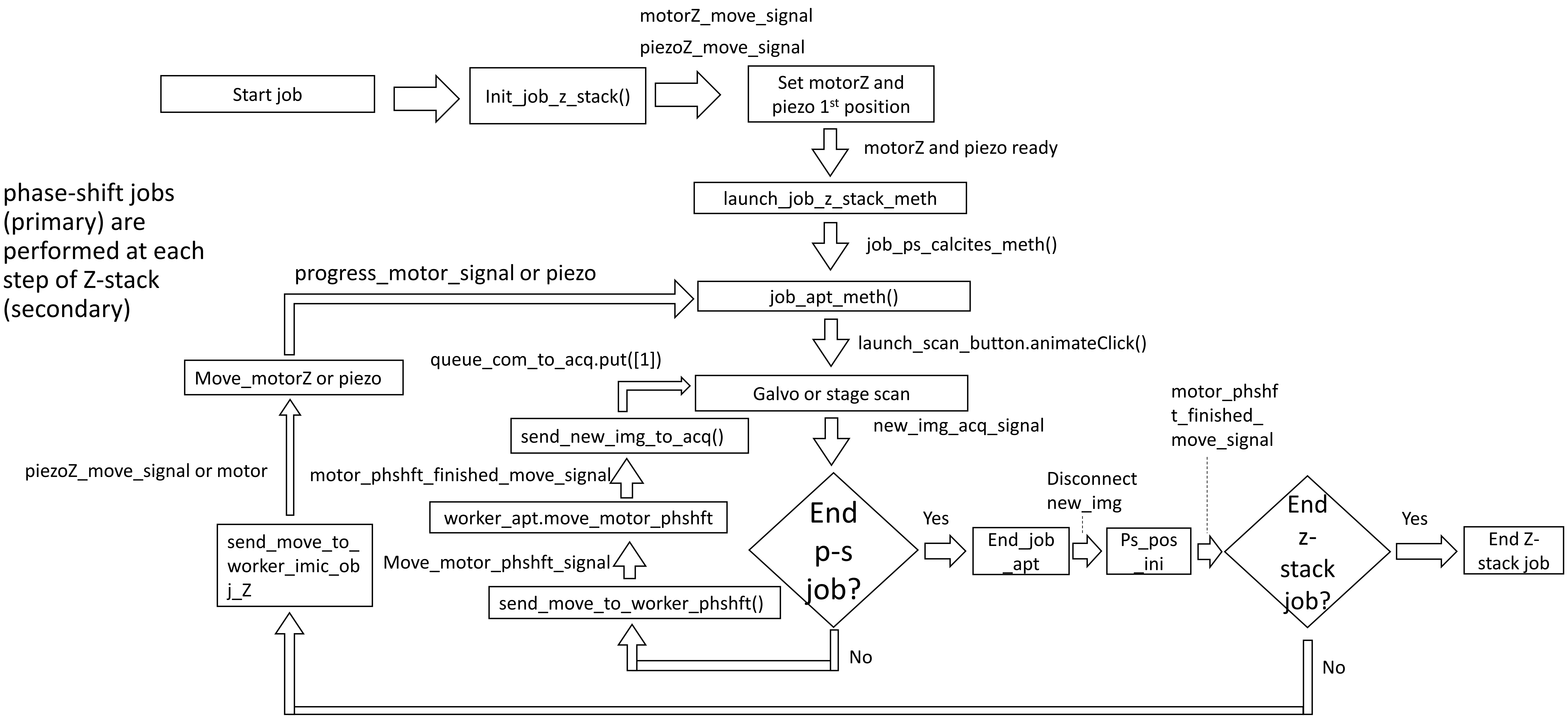
progress_motor_signal or piezo

send_new_img_to_acq

queue_com_to_acq.put([1])

The jobs, **Case 1 : Z-stack > phase-shift (updated 2018/07/09)**

→ phase-shift jobs (primary) are performed at each step of Z-stack (secondary)



Start job

Init_job_z_stack()

motorZ_move_signal

piezoZ_move_signal

Set motorZ and piezo 1st position

motorZ and piezo ready

launch_job_z_stack_meth

job_ps_calcites_meth()

job_apt_meth()

launch_scan_button.animateClick()

Galvo or stage scan

new_img_acq_signal

End p-s job?

Yes

End_job _apt

Disconnect new_img

Ps_pos _ini

motor_phshft_finished_move_signal

End z-stack job?

Yes

End Z-stack job

No

send_move_to_worker_phshft()

Move_motor_phshft_signal

worker_apt.move_motor_phshft

motor_phshft_finished_move_signal

send_new_img_to_acq()

queue_com_to_acq.put([1])

No

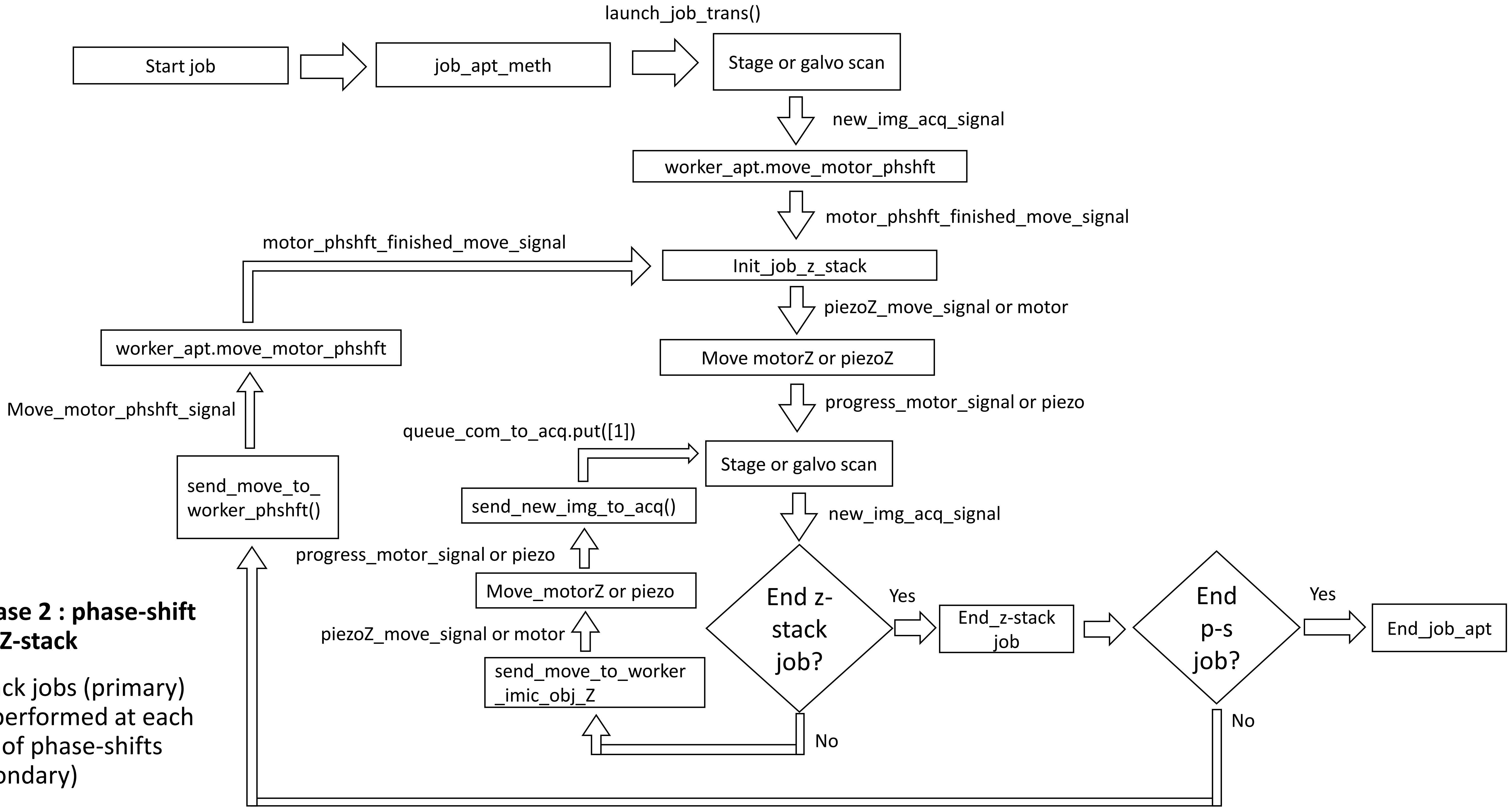
send_move_to_worker_imic_obj_Z

piezoZ_move_signal or motor

Move_motorZ or piezo

progress_motor_signal or piezo

The jobs, **Case 2 : phase-shift > Z-stack (updated 2018/07/09)**


launch_job_trans()
Start job
job_apt_meth
Stage or galvo scan
new_img_acq_signal
worker_apt.move_motor_phshft
motor_phshft_finished_move_signal
motor_phshft_finished_move_signal
Init_job_z_stack
piezoZ_move_signal or motor
worker_apt.move_motor_phshft
Move motorZ or piezoZ
progress_motor_signal or piezo
Move_motor_phshft_signal
queue_com_to_acq.put([1])
Stage or galvo scan
send_move_to_ worker_phshft()
send_new_img_to_acq()
new_img_acq_signal
progress_motor_signal or piezo
Move_motorZ or piezo
End z-stack job?
Yes
End_z-stack job
End p-s job?
Yes
End_job_apt
piezoZ_move_signal or motor
send_move_to_worker _imic_obj_Z
No
No

- **Case 2 : phase-shift > Z-stack**

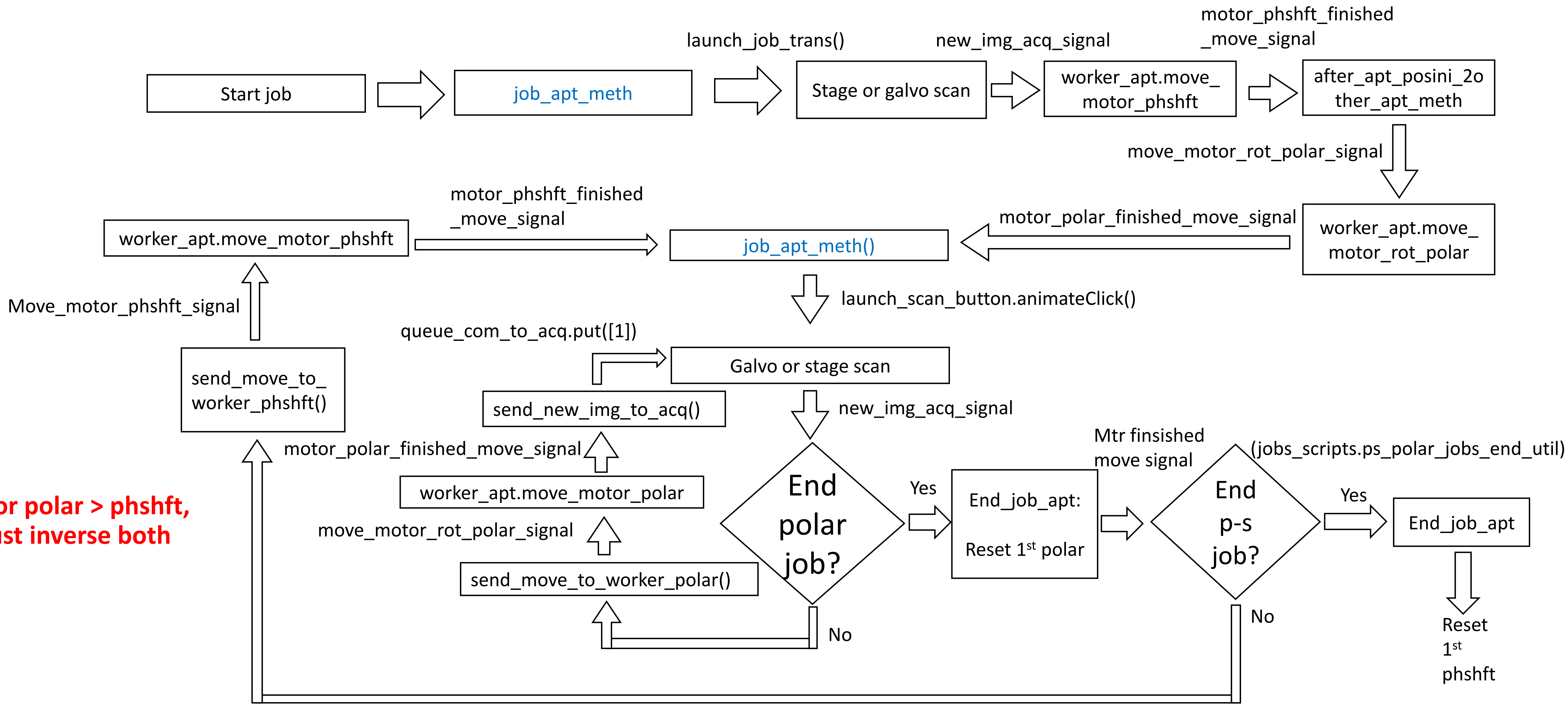
Z-stack jobs (primary) are performed at each step of phase-shifts (secondary)

The jobs, **Case 3 : polar jobs , + Z**

→ Same APT worker

→ Replace phshft by polar

The jobs, **Case 4 : phase-shift > polar (updated 2018/07/09)**



Start job

job_apt_meth

launch_job_trans()

Stage or galvo scan

new_img_acq_signal

worker_apt.move_motor_phshft

motor_phshft_finished_move_signal

after_apt_posini_2other_apt_meth

move_motor_rot_polar_signal

worker_apt.move_motor_rot_polar

motor_polar_finished_move_signal

job_apt_meth()

motor_phshft_finished_move_signal

worker_apt.move_motor_phshft

Move_motor_phshft_signal

send_move_to_worker_phshft()

launch_scan_button.animateClick()

queue_com_to_acq.put([1])

Galvo or stage scan

send_new_img_to_acq()

new_img_acq_signal

motor_polar_finished_move_signal

worker_apt.move_motor_polar

move_motor_rot_polar_signal

send_move_to_worker_polar()

End polar job?

Yes

No

End_job_apt:

Reset 1st polar

Mtr finsished move signal

End p-s job?

(jobs_scripts.ps_polar_jobs_end_util)

Yes

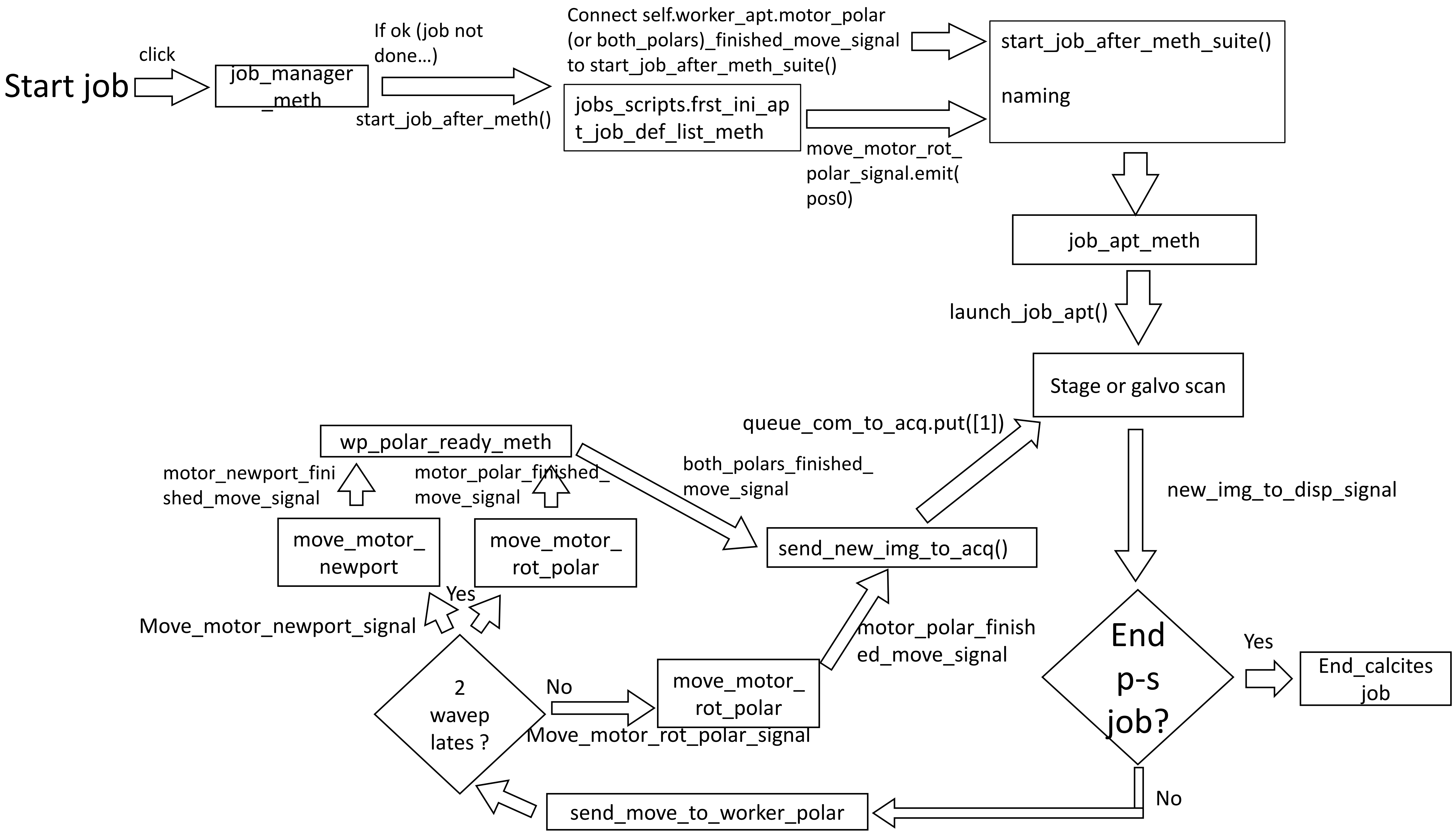
No

End_job_apt

Reset 1st phshft

- **For polar > phshft, just inverse both**

The jobs, **Case 05 : polar with many WP**



Start job

click

job_manager_meth

If ok (job not done…)

start_job_after_meth()

jobs_scripts.frst_ini_apt_job_def_list_meth

Connect self.worker_apt.motor_polar (or both_polars)_finished_move_signal to start_job_after_meth_suite()

move_motor_rot_polar_signal.emit(pos0)

start_job_after_meth_suite()

naming

job_apt_meth

launch_job_apt()

Stage or galvo scan

new_img_to_disp_signal

End p-s job?

Yes

End_calcites job

No

send_move_to_worker_polar

2 waveplates ?

No

Move_motor_rot_polar_signal

move_motor_rot_polar

motor_polar_finished_move_signal

Yes

Move_motor_newport_signal

move_motor_newport

move_motor_rot_polar

motor_newport_finished_move_signal

motor_polar_finished_move_signal

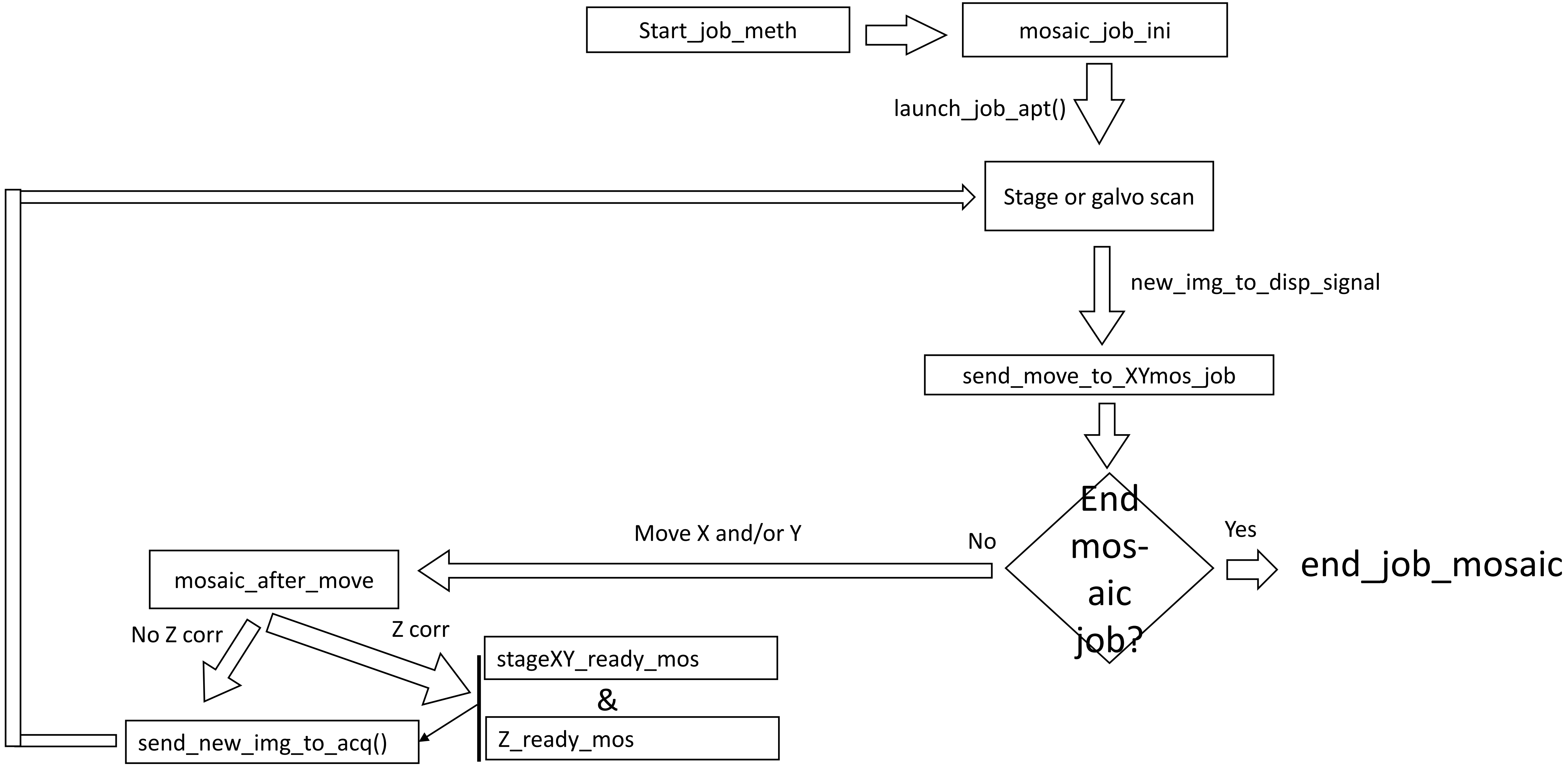
wp_polar_ready_meth

both_polars_finished_move_signal

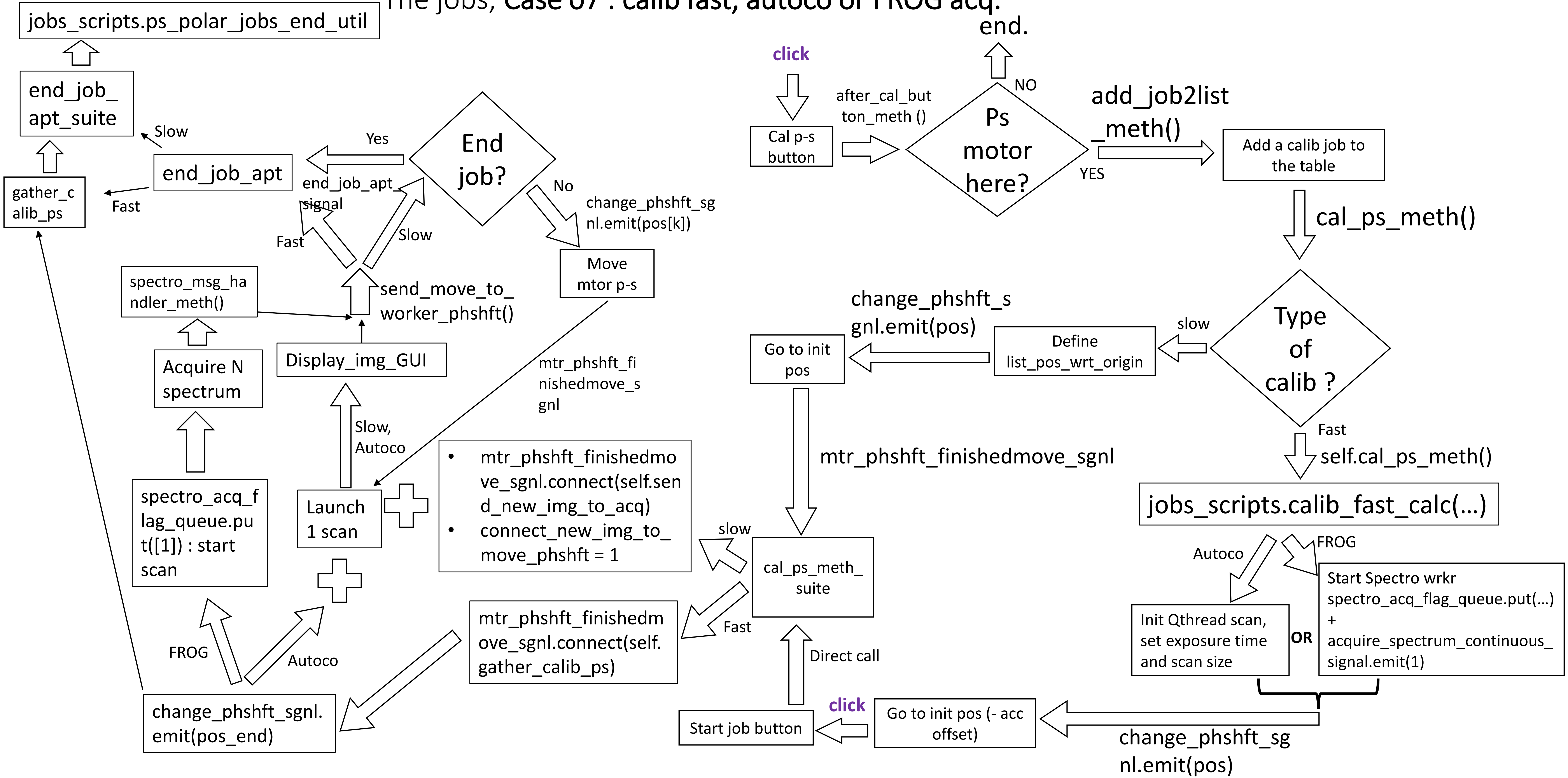
send_new_img_to_acq()

queue_com_to_acq.put([1])

The jobs, **Case 06 : mosaic, with or without Z corr.**



The jobs, **Case 07 : calib fast, autoco or FROG acq.**



click

Cal p-s button

after_cal_button_meth ()

Ps motor here?

NO

end.

YES

add_job2list_meth()

Add a calib job to the table

cal_ps_meth()

Type of calib ?

slow

Define list_pos_wrt_origin

change_phshft_sgnl.emit(pos)

Go to init pos

mtr_phshft_finishedmove_sgnl

Fast

self.cal_ps_meth()

jobs_scripts.calib_fast_calc(...)

Autoco

FROG

Init Qthread scan, set exposure time and scan size

OR

Start Spectro wrkr spectro_acq_flag_queue.put(...) + acquire_spectrum_continuous_signal.emit(1)

change_phshft_sgnl.emit(pos)

Go to init pos (- acc offset)

click

Start job button

Direct call

cal_ps_meth_suite

slow

- mtr_phshft_finishedmove_sgnl.connect(self.send_new_img_to_acq)
- connect_new_img_to_move_phshft = 1

Fast

mtr_phshft_finishedmove_sgnl.connect(self.gather_calib_ps)

change_phshft_sgnl.emit(pos_end)

FROG

spectro_acq_flag_queue.put([1]) : start scan

Acquire N spectrum

spectro_msg_handler_meth()

Autoco

Launch 1 scan

Slow, Autoco

Display_img_GUI

send_move_to_worker_phshft()

Fast

Slow

End job?

Yes

end_job_apt_signal

end_job_apt

Slow

end_job_apt_suite

jobs_scripts.ps_polar_jobs_end_util

Fast

gather_calib_ps

No

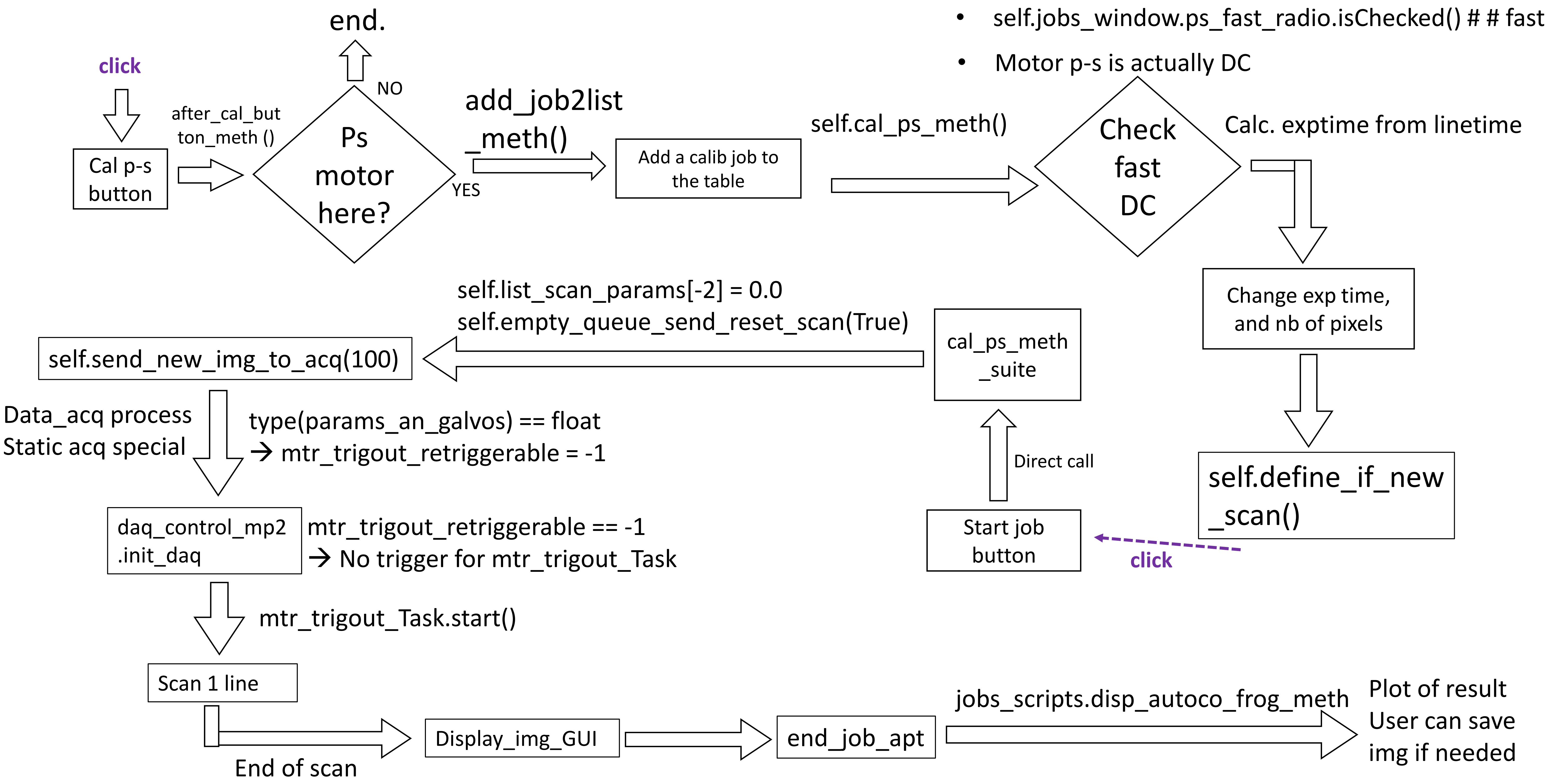
change_phshft_sgnl.emit(pos[k])

Move mtor p-s

mtr_phshft_finishedmove_sgnl

The jobs, **Case 08 : calib fast of EOM AC**

Cond: fast calib mode, with DC as p-s motor (in 2nd job window)
→ User has to put himself the right ramp mode



- self.jobs_window.ps_fast_radio.isChecked() # # fast
- Motor p-s is actually DC

click

Cal p-s button

after_cal_button_meth ()

Ps motor here?

NO

end.

YES

add_job2list_meth()

Add a calib job to the table

self.cal_ps_meth()

Check fast DC

Calc. exptime from linetime

Change exp time, and nb of pixels

self.define_if_new_scan()

click

Start job button

Direct call

cal_ps_meth_suite

self.list_scan_params[-2] = 0.0
self.empty_queue_send_reset_scan(True)

self.send_new_img_to_acq(100)

Data_acq process
Static acq special

type(params_an_galvos) == float
→ mtr_trigout_retriggerable = -1

daq_control_mp2.init_daq

mtr_trigout_retriggerable == -1
→ No trigger for mtr_trigout_Task

mtr_trigout_Task.start()

Scan 1 line

End of scan

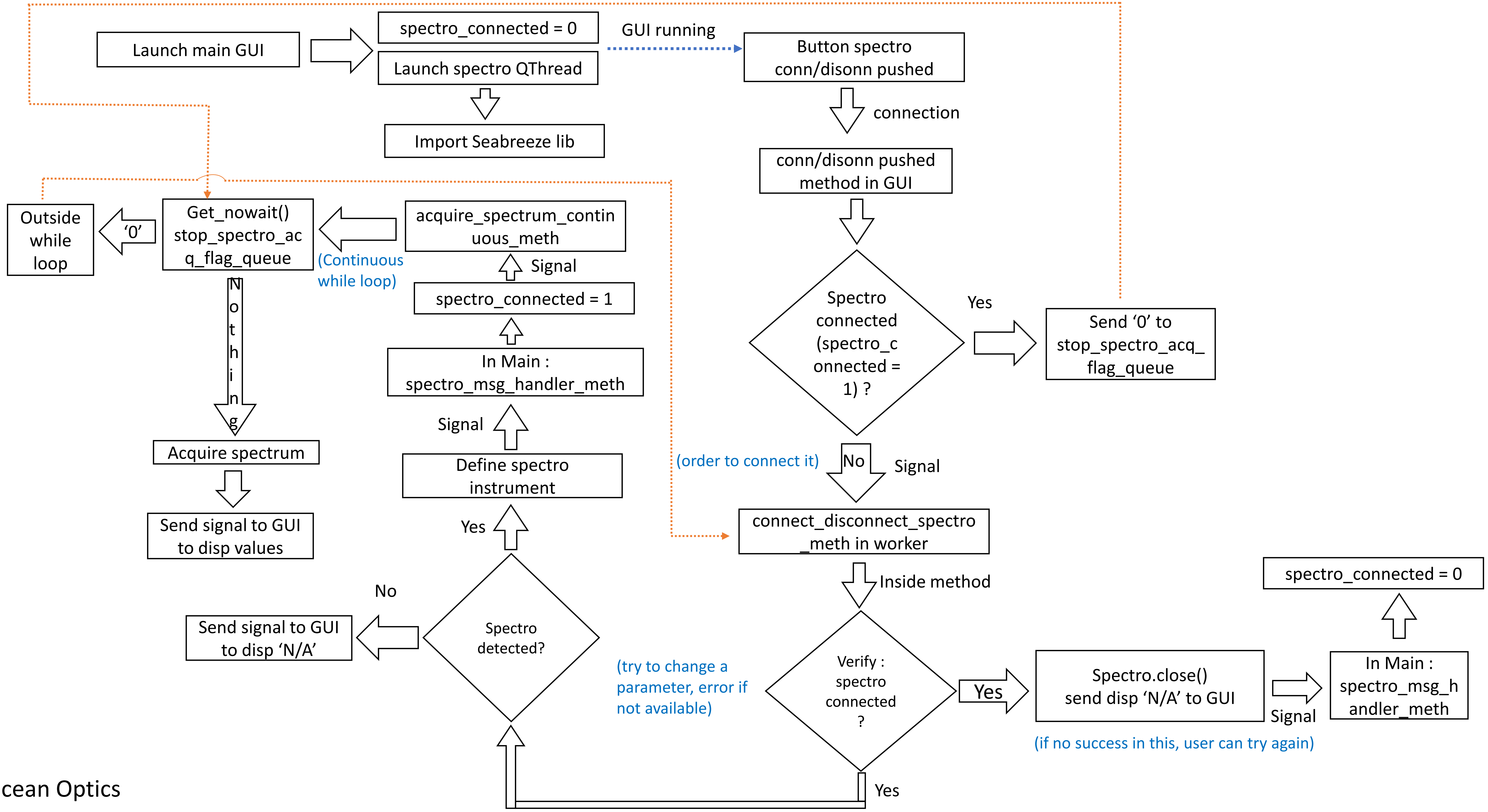
Display_img_GUI

end_job_apt

jobs_scripts.disp_autoco_frog_meth

Plot of result
User can save img if needed

Spectro QThread



Launch main GUI

spectro_connected = 0

Launch spectro QThread

Import Seabreeze lib

GUI running

Button spectro conn/disonn pushed

connection

conn/disonn pushed method in GUI

Spectro connected (spectro_connected = 1) ?

Yes

Send ‘0’ to stop_spectro_acq_flag_queue

No

Signal

(order to connect it)

connect_disconnect_spectro_meth in worker

Inside method

Verify : spectro connected ?

Yes

Spectro.close()
send disp ‘N/A’ to GUI

(if no success in this, user can try again)

Signal

In Main : spectro_msg_handler_meth

spectro_connected = 0

Yes

Spectro detected?

(try to change a parameter, error if not available)

No

Send signal to GUI to disp ‘N/A’

Yes

Define spectro instrument

Signal

In Main : spectro_msg_handler_meth

spectro_connected = 1

Signal

acquire_spectrum_continuous_meth

(Continuous while loop)

Get_nowait() stop_spectro_acq_flag_queue

‘0’

Outside while loop

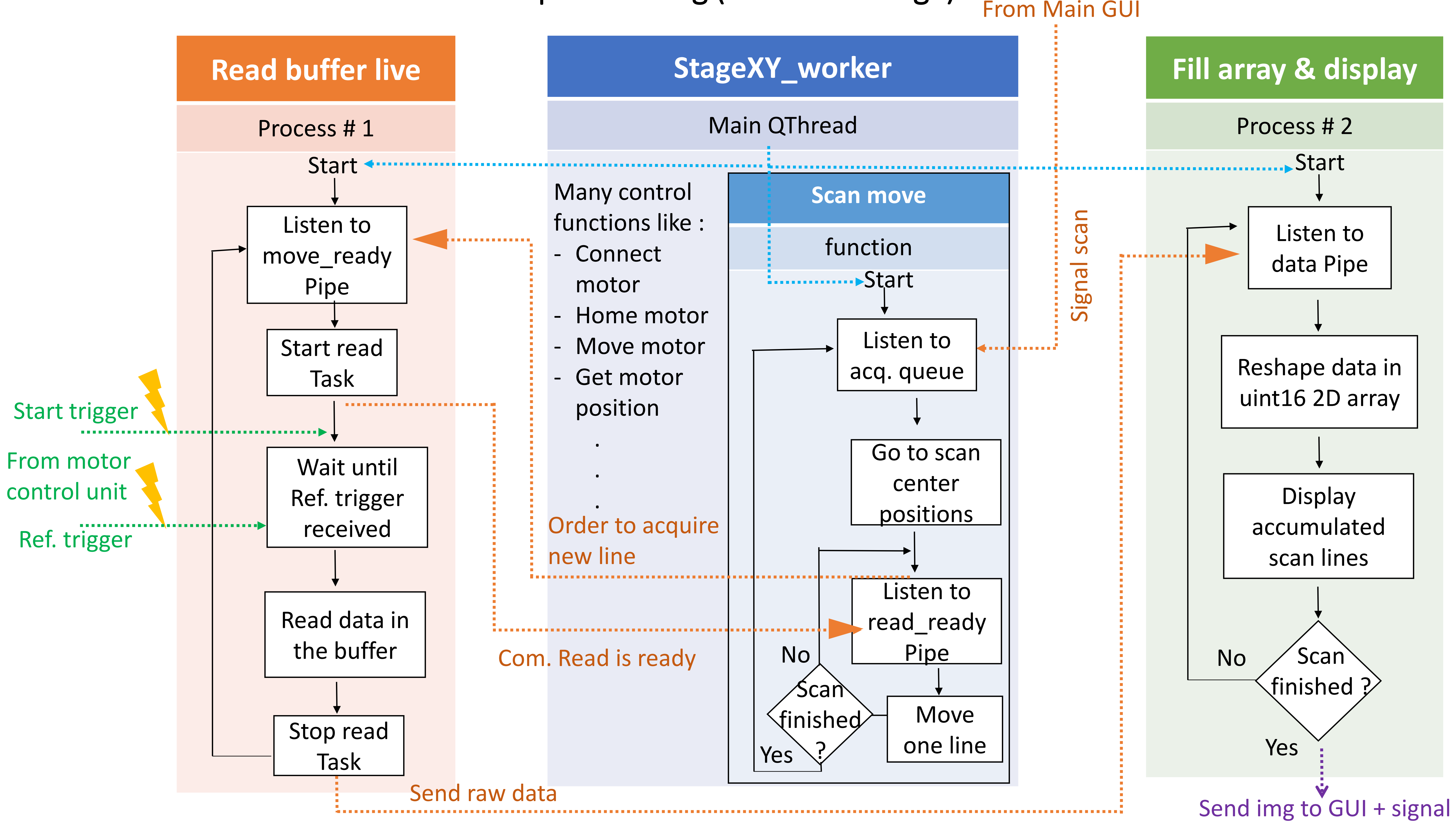
Nothing

Acquire spectrum

Send signal to GUI to disp values

Ocean Optics

Sample scanning (motorized stage)



Read buffer live

Process # 1

Start

Listen to move_ready Pipe

Start read Task

Start trigger

From motor control unit

Ref. trigger

Wait until Ref. trigger received

Read data in the buffer

Stop read Task

Send raw data

StageXY_worker

Main QThread

Many control functions like :

- Connect motor
- Home motor
- Move motor
- Get motor position

.

.

.

Order to acquire new line

Com. Read is ready

Scan move

function

Start

Listen to acq. queue

Go to scan center positions

Listen to read_ready Pipe

Move one line

Scan finished ?

No

Yes

From Main GUI

Signal scan

Fill array & display

Process # 2

Start

Listen to data Pipe

Reshape data in uint16 2D array

Display accumulated scan lines

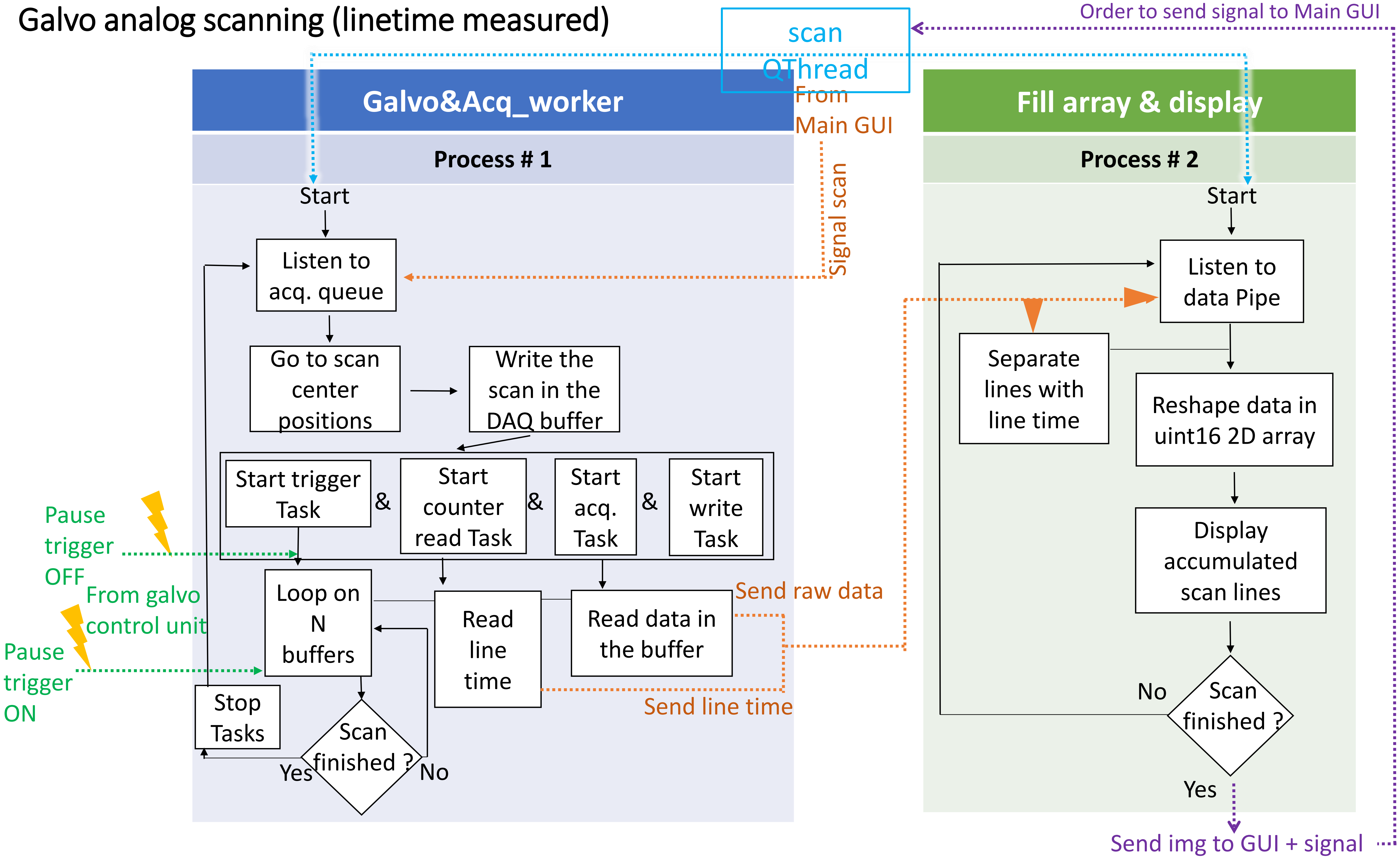
Scan finished ?

No

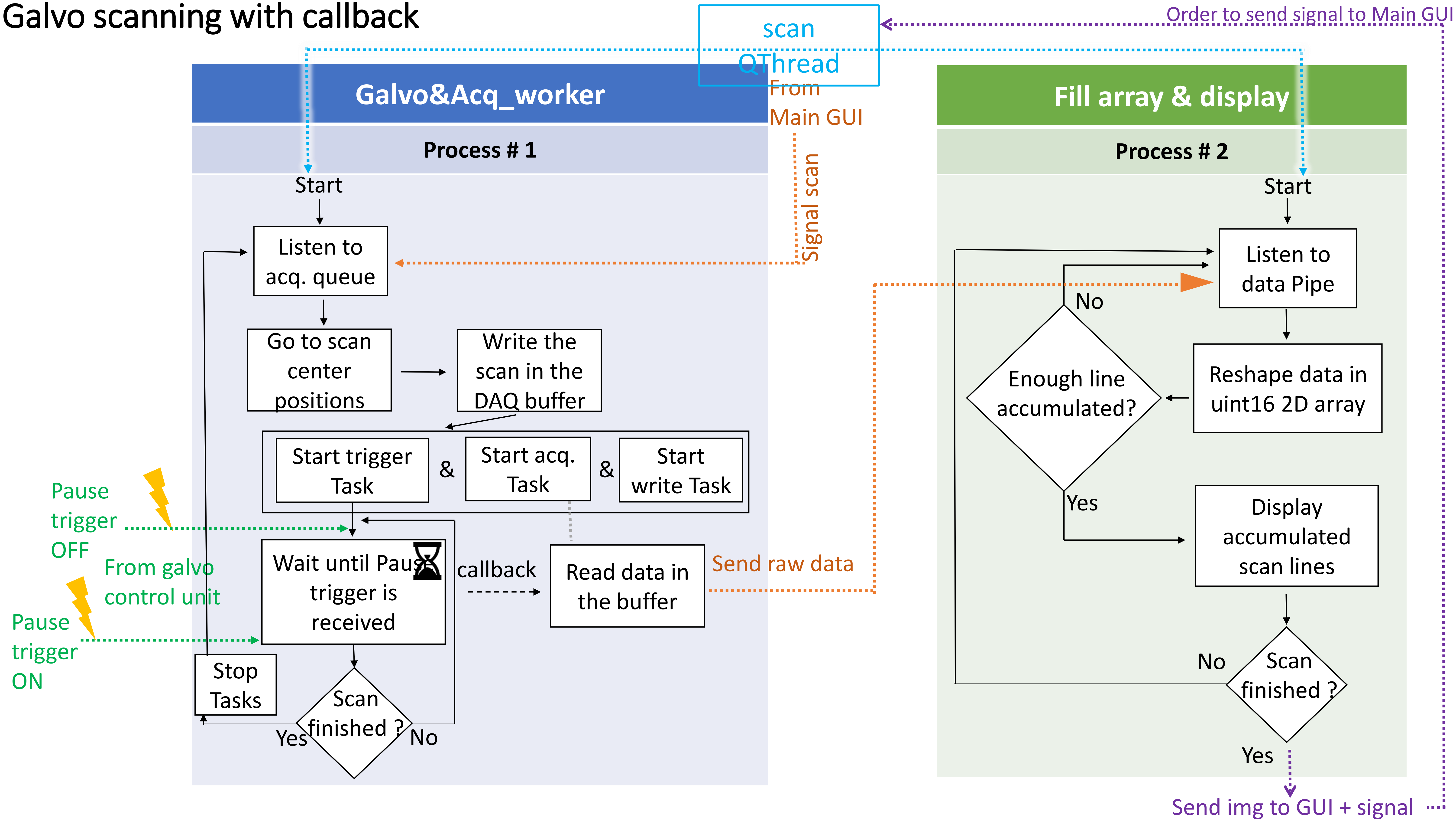
Yes

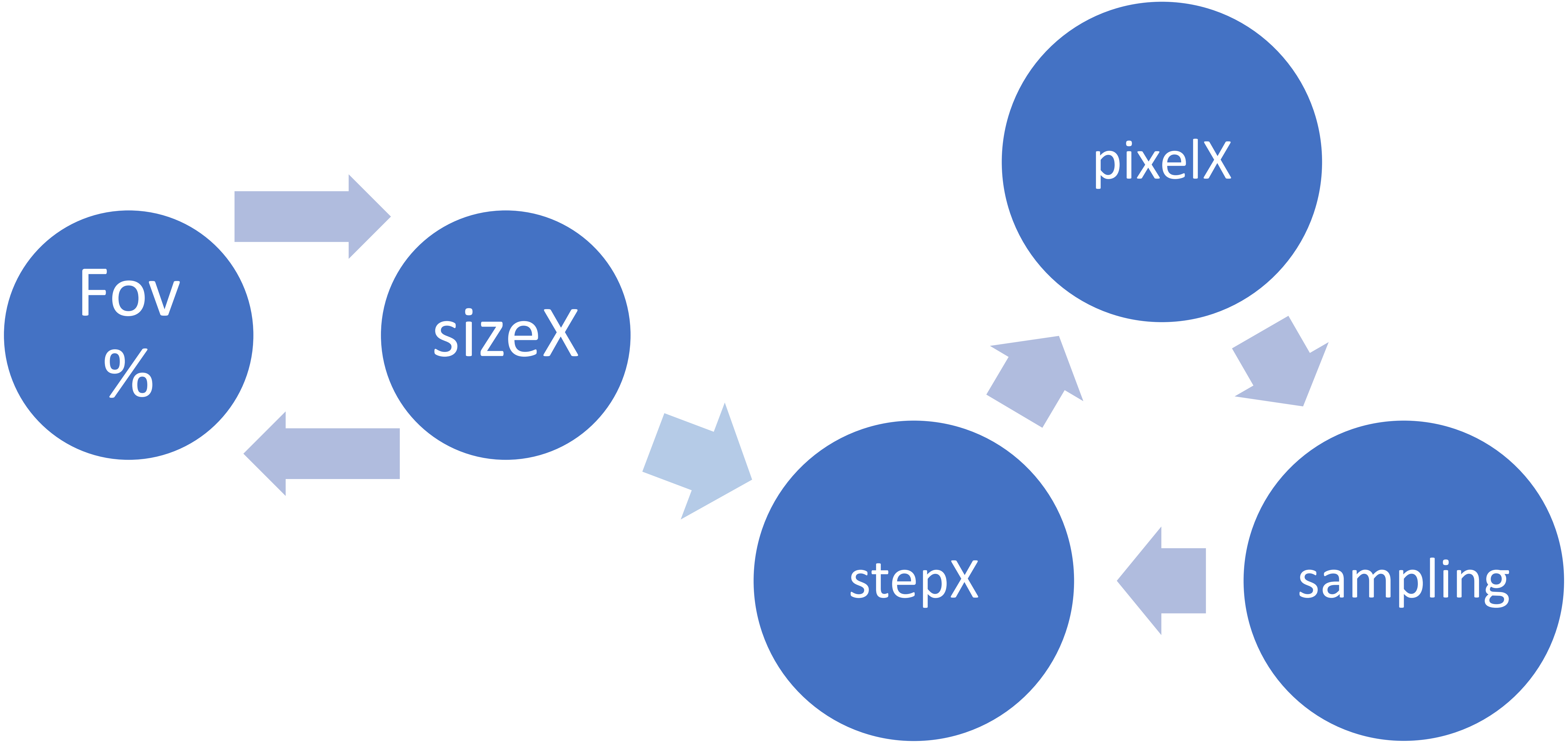
Send img to GUI + signal

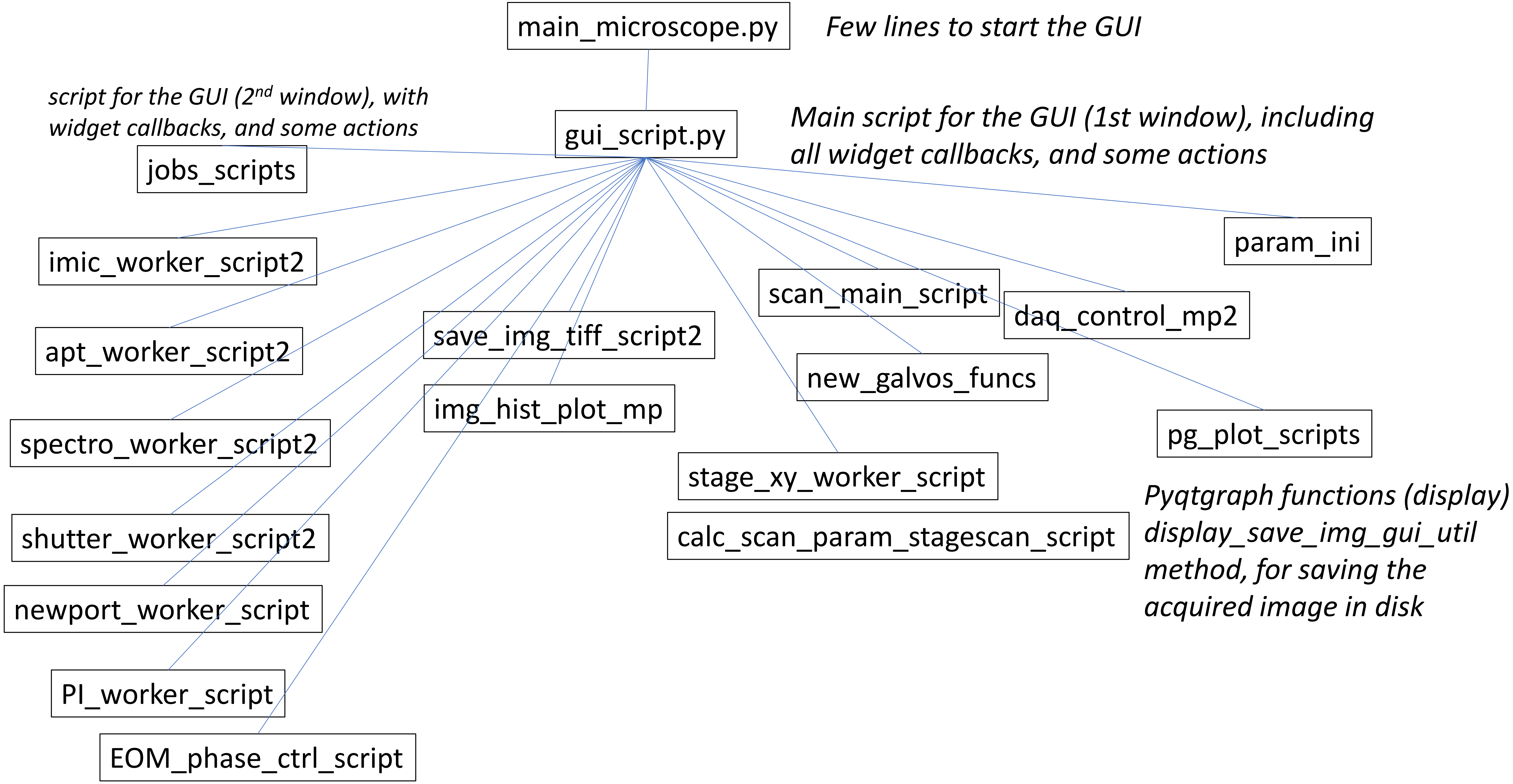
Galvo analog scanning (linetime measured)



Galvo scanning with callback




Fov
%
sizeX
pixelX
stepX
sampling



main_microscope.py

Few lines to start the GUI

gui_script.py

Main script for the GUI (1st window), including all widget callbacks, and some actions

script for the GUI (2^{nd} window), with widget callbacks, and some actions

jobs_scripts

imic_worker_script2

apt_worker_script2

spectro_worker_script2

shutter_worker_script2

newport_worker_script

PI_worker_script

EOM_phase_ctrl_script

save_img_tiff_script2

img_hist_plot_mp

scan_main_script

new_galvos_funcs

stage_xy_worker_script

calc_scan_param_stagescan_script

param_ini

daq_control_mp2

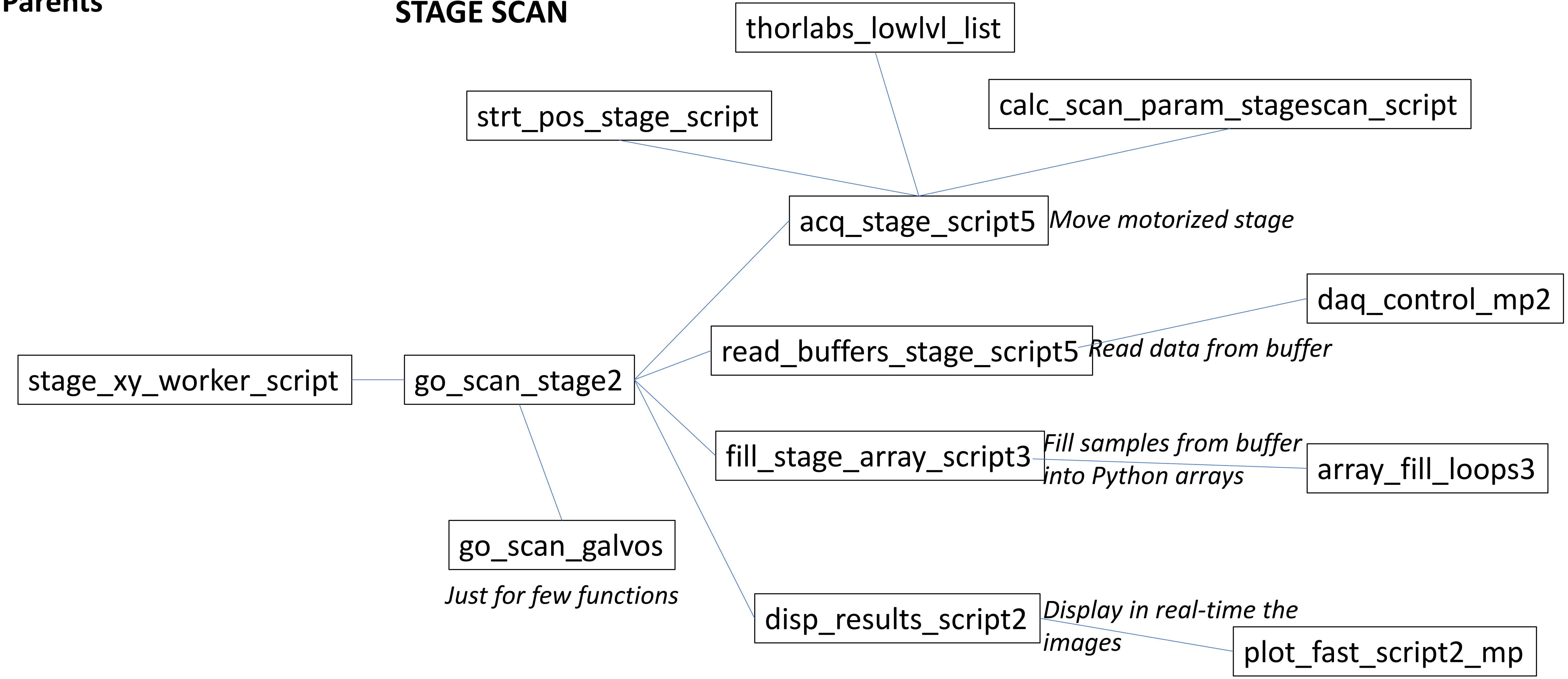
pg_plot_scripts

Pyqtgraph functions (display) display_save_img_gui_util method, for saving the acquired image in disk

STAGE SCAN

Parents

Children



thorlabs_lowlvl_list

strt_pos_stage_script

calc_scan_param_stagescan_script

acq_stage_script5

Move motorized stage

daq_control_mp2

stage_xy_worker_script

go_scan_stage2

read_buffers_stage_script5

Read data from buffer

fill_stage_array_script3

Fill samples from buffer into Python arrays

array_fill_loops3

go_scan_galvos

Just for few functions

disp_results_script2

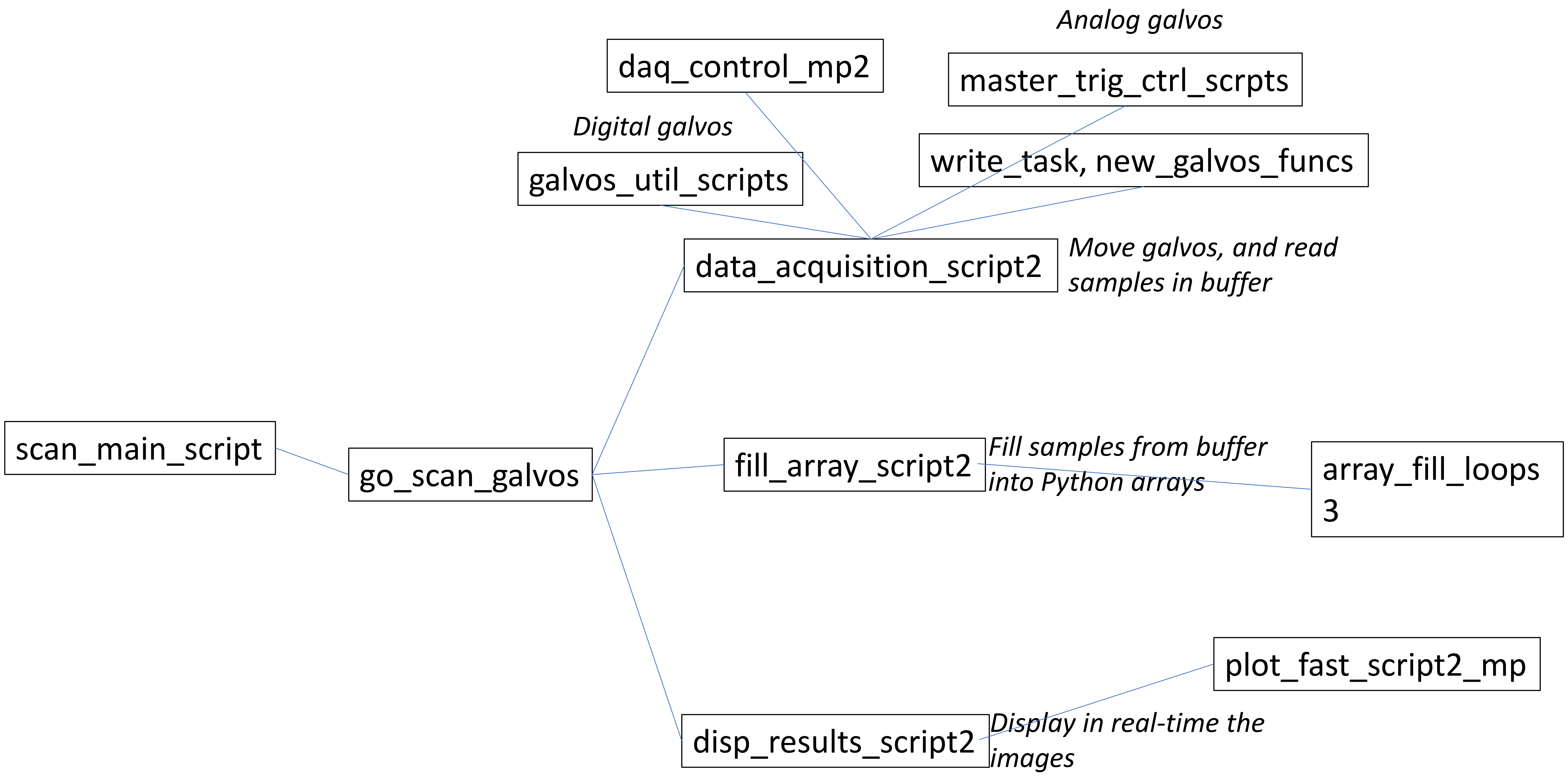
Display in real-time the images

plot_fast_script2_mp

Parents

GALVOS SCAN

Children



Analog galvos

daq_control_mp2

master_trig_ctrl_scrpts

Digital galvos

galvos_util_scripts

write_task, new_galvos_funcs

data_acquisition_script2

Move galvos, and read samples in buffer

scan_main_script

go_scan_galvos

fill_array_script2

Fill samples from buffer into Python arrays

array_fill_loops 3

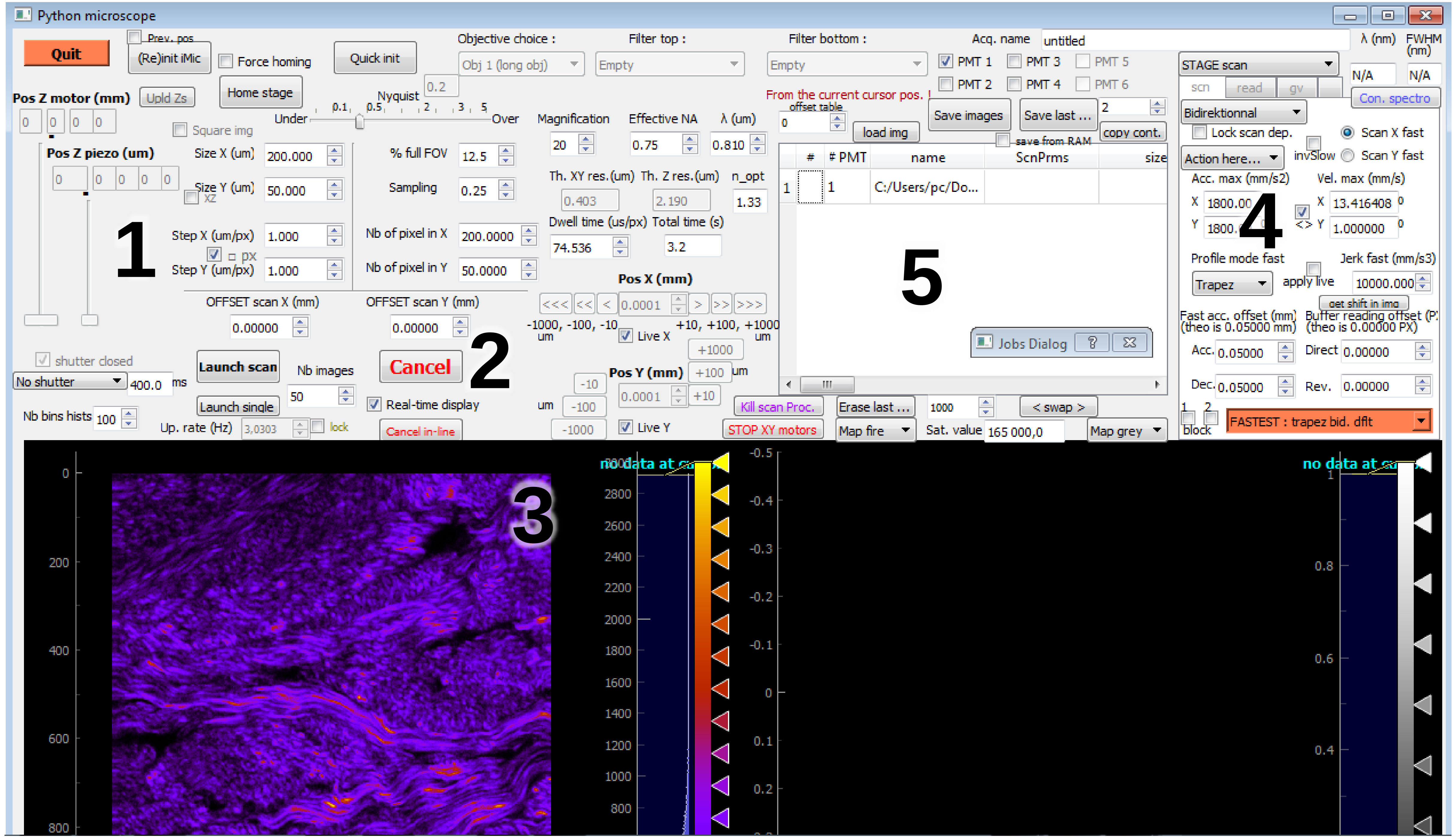
plot_fast_script2_mp

disp_results_script2

Display in real-time the images

Full GUI description

Full GUI description in Python



Python microscope

Acq. name: untitled

Objective choice : Obj 1 (long obj)

Filter top : Empty

Filter bottom : Empty

☑ PMT 1 ☐ PMT 3 ☐ PMT 5
☐ PMT 2 ☐ PMT 4 ☐ PMT 6

λ (nm): N/A FWHM (nm): N/A

STAGE scan

Bidirektionnal

☐ Lock scan dep. ● Scan X fast ○ Scan Y fast

Pos Z motor (mm): 0 0 0 0

Pos Z piezo (um): 0 0 0 0 0

☐ Square img

Size X (um): 200.000
Size Y (um): 50.000
Step X (um/px): 1.000 ☑ □ px
Step Y (um/px): 1.000

OFFSET scan X (mm): 0.00000
OFFSET scan Y (mm): 0.00000

Nyquist: 0.2

Magnification: 20
Effective NA: 0.75
λ (um): 0.810

% full FOV: 12.5
Sampling: 0.25

Th. XY res.(um): 0.403
Th. Z res.(um): 2.190
n_opt: 1.33

Nb of pixel in X: 200.0000
Nb of pixel in Y: 50.0000

Dwell time (us/px): 74.536
Total time (s): 3.2

Pos X (mm): 0.0001 ☑ Live X
Pos Y (mm): 0.0001 ☑ Live Y

☑ shutter closed
No shutter 400.0 ms

Nb bins hists: 100
Up. rate (Hz): 3,0303 ☐ lock

Nb images: 50
☑ Real-time display

From the current cursor pos. offset table: 0

#	# PMT	name	ScnPrms	size
1	1	C:/Users/pc/Do...		

Acc. max (mm/s2): X 1800.00 Y 1800.
Vel. max (mm/s): X 13.416408 Y 1.000000

Profile mode fast: Trapez ☐ apply live
Jerk fast (mm/s3): 10000.000

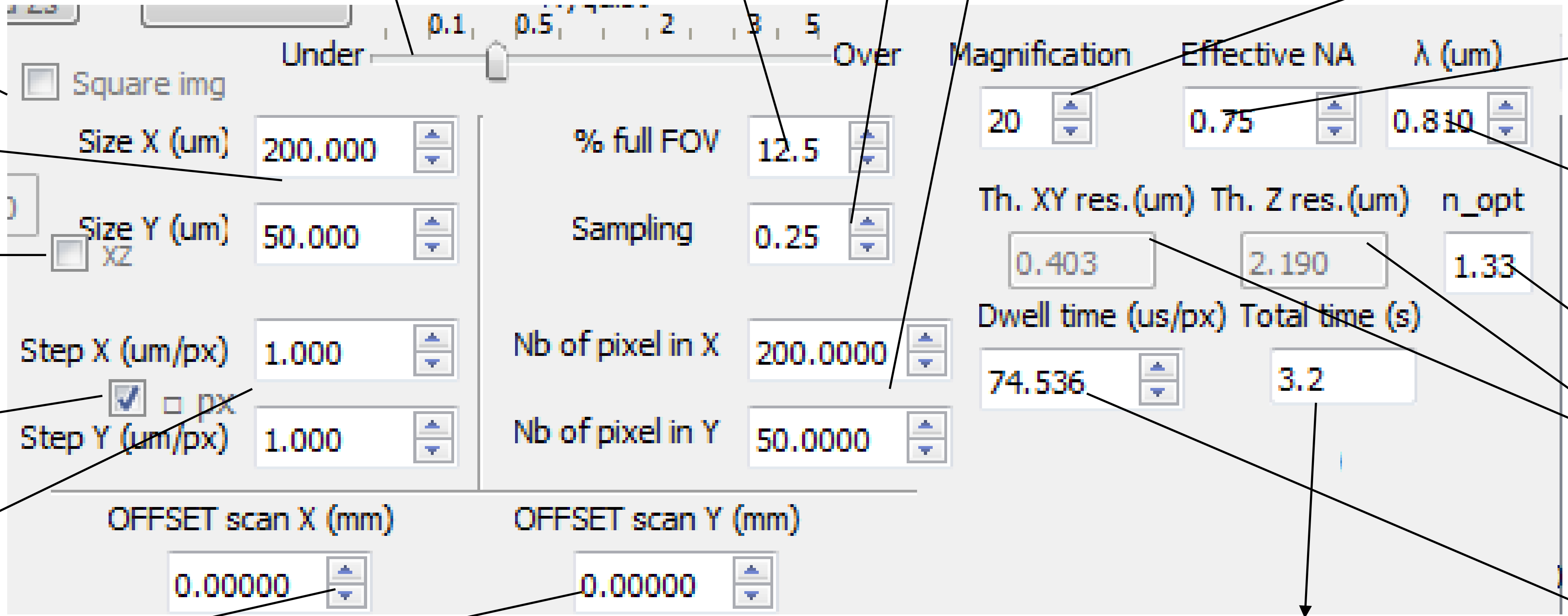
Fast acc. offset (mm) (theo is 0.05000 mm): Acc. 0.05000 Dec. 0.05000
Buffer reading offset (P (theo is 0.00000 PX): Direct 0.00000 Rev. 0.00000

FASTEST : trapez bid. dflt

Erase last ... 1000 < swap >

Map fire Sat. value 165 000,0 Map grey

1) GUI : (main) scan parameters



□ Square img

Under 0.1 0.5 2 3 5 Over

Size X (um) 200.000

Size Y (um) 50.000

□ XZ

Step X (um/px) 1.000

☑ □ px

Step Y (um/px) 1.000

% full FOV 12.5

Sampling 0.25

Nb of pixel in X 200.0000

Nb of pixel in Y 50.0000

OFFSET scan X (mm) 0.00000

OFFSET scan Y (mm) 0.00000

Magnification 20

Effective NA 0.75

λ (um) 0.810

Th. XY res.(um) 0.403

Th. Z res.(um) 2.190

n_opt 1.33

Dwell time (us/px) 74.536

Total time (s) 3.2

Impose square image

SizeX/Y in µm

Do a XZ scan instead of XY (need modif to be implemented)

Impose square pixels

Step X/Y in µm/px

Offset in X/Y of the scan, used with galvos to change the 0 pos of the alignment.

Ratio to the Nyquist factor

A scan size corresponds to a % of the FOV

A scan step corresponds to a sampling

Corresponding number of pixels, knowing the scan size and step

Magnification of the objective

N.A. (measured) of objective

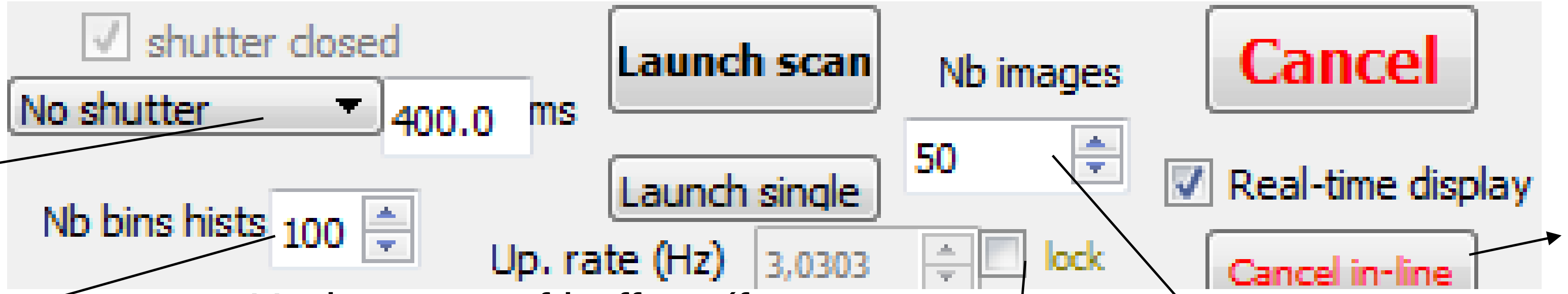
Fundamental wavelength

Index of medium (sample)

THEORETICAL resolution in SHG (Zipfel et al.)

Exposure time/px

Total time of img (s)



☑ shutter closed

No shutter ▾ 400.0 ms

Launch scan

Nb images 50

Cancel

☑ Real-time display

Launch single

Nb bins hists 100

Up. rate (Hz) 3,0303 □ lock

Cancel in-line

Use of shutter, or not

Nb of bins for the 2D histogram

Update rate of buffers (for static acq., or galvos)
Higher rate→ smaller packets

Lock update rate

Nb of images to acquire

Cancel image during acquisition (not ok for dig galvos, as scan has to finish anyways)

2) GUI : motor(s) control

Use previous positions in Z (last time soft was closed), and for offsets

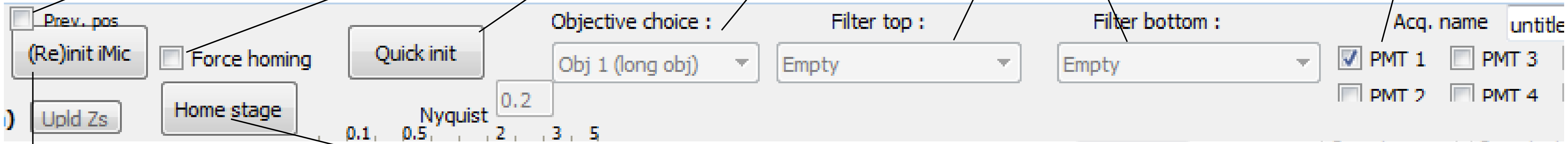
Force physical movement to home stage XY

StageXY+posZ init

iMic objective

Filters iMic position

Use of PMT # N



□ Prev. pos

Objective choice : Obj 1 (long obj)

Filter top : Empty

Filter bottom : Empty

Acq. name untitle

☑ PMT 1 □ PMT 3
□ PMT 2 □ PMT 4

Nyquist 0.2

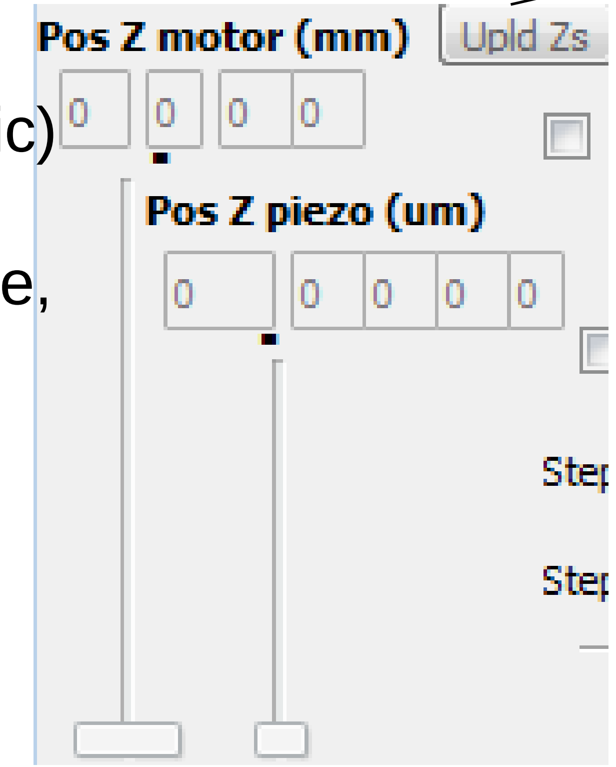
Init iMic position

Home or check pos of stage XY

Upload positions in Z

Z position (coarse, iMic)

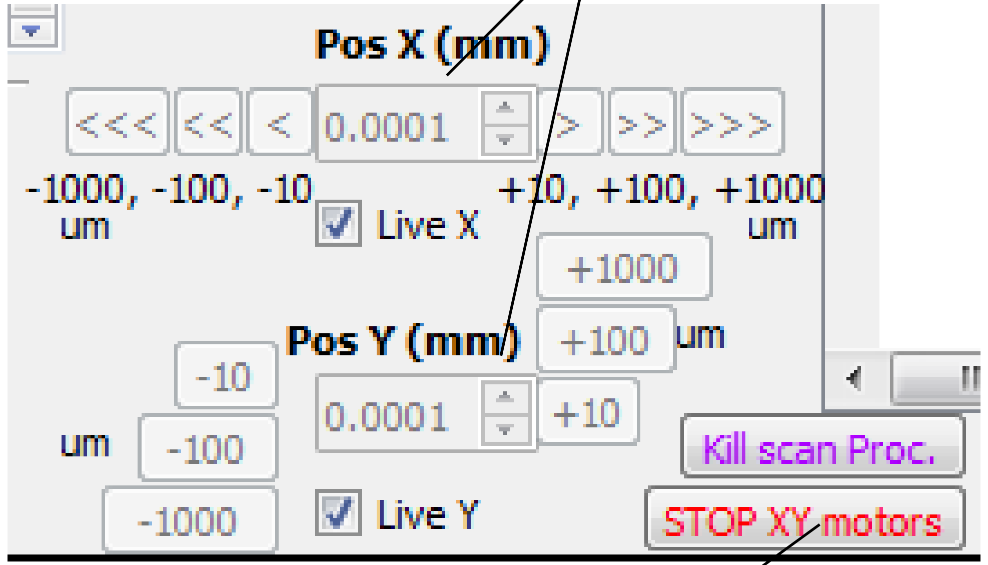
Z position (fine, piezo)



Pos Z motor (mm) 0 0 0 0

Pos Z piezo (um) 0 0 0 0 0

Position of the sample



Pos X (mm) 0.0001

Pos Y (mm) 0.0001

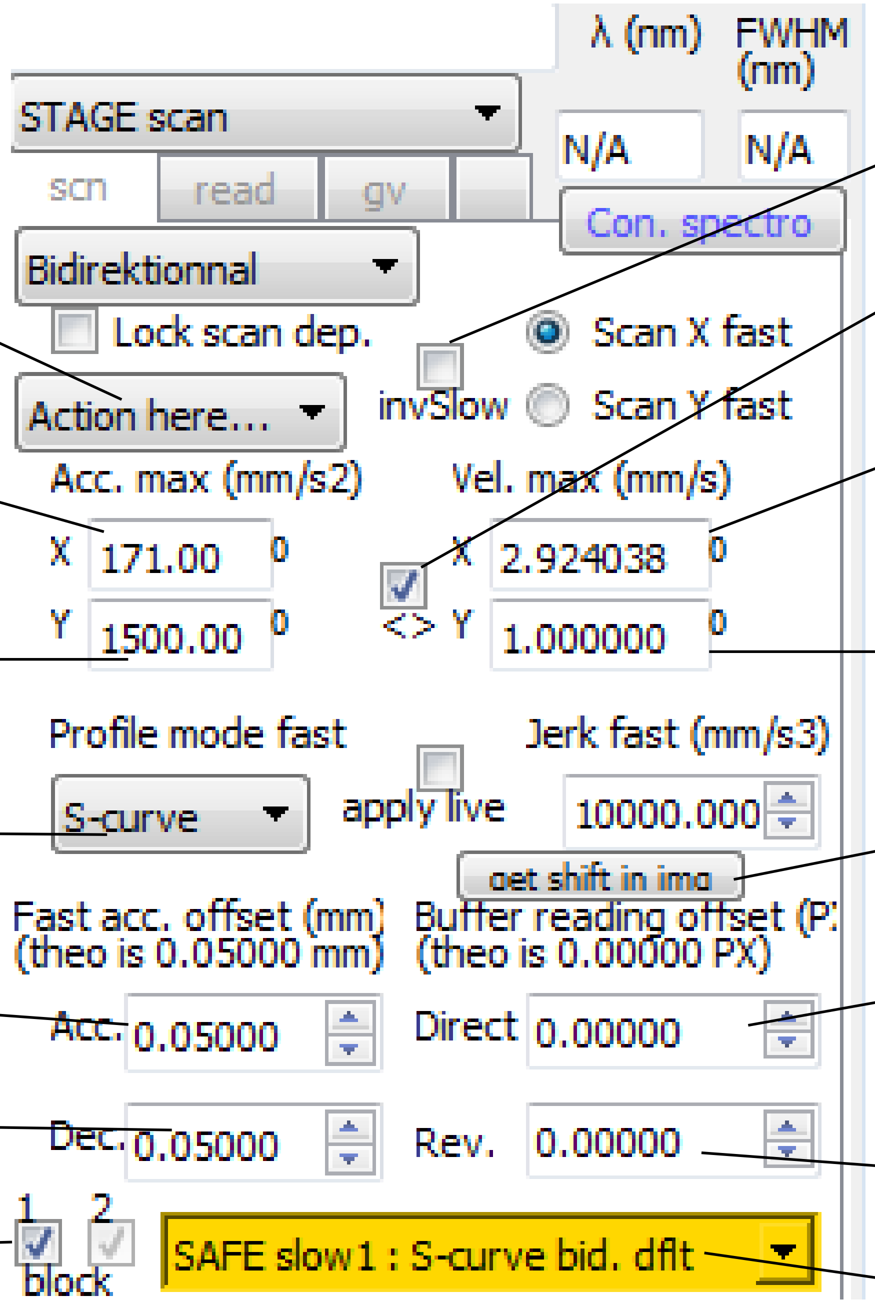
☑ Live X

☑ Live Y

Stop the motor XY (emergency)

4) GUI: Detailed parameters of the mode

FOR STAGE SCAN



Force update, or set of optimal params, or lock params

Acceleration of the motor (X, usually fast)

Acceleration of the motor (Y, usually slow)

Scan mode profile

Acceleration offset (spatial)

Deceleration offset (spatial)

Block until end of move (can uncheck for fast) See stage_scan summary

Invert slow direction

Enable/disable link between acceleration and speed of motor

Velocity of the motor (X, usually fast)

Velocity of the motor (Y, usually slow)

Get shifts (fast unprecise, use DIPimage for more precise)

reading offset (in buffer)

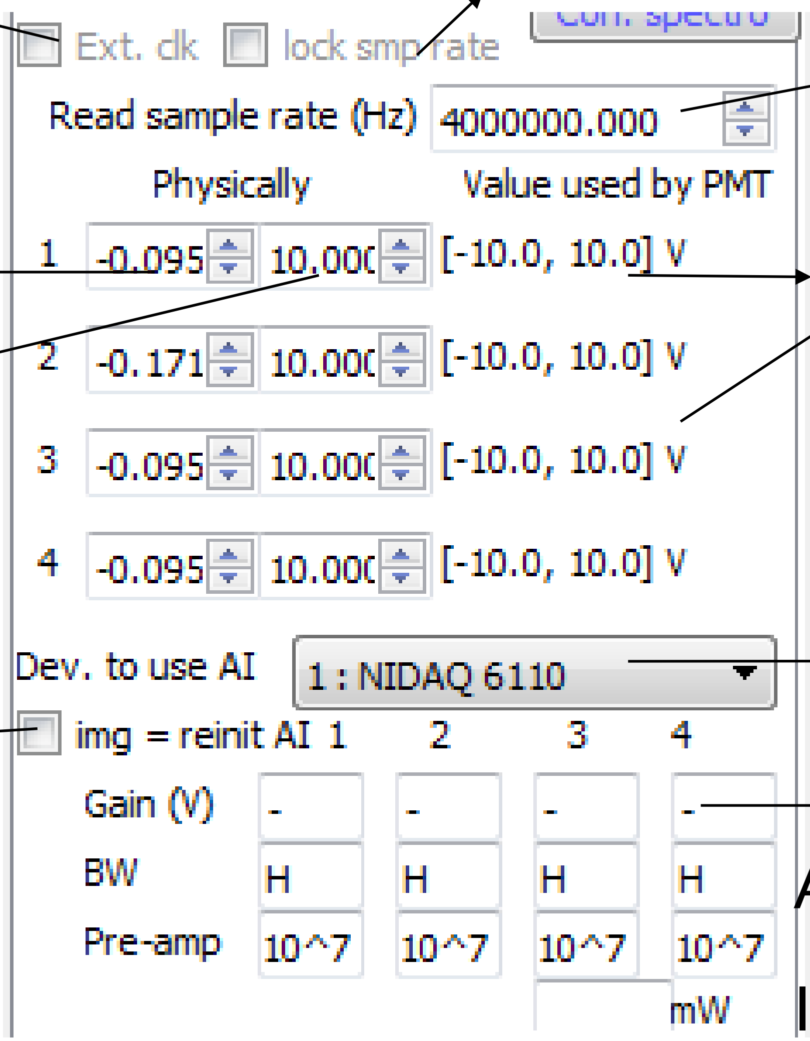
reading offset (in buffer, reverse)

Easy change of mode

4b) GUI: Detailed parameters of the mode

Use external clock for sample rate

Lock sample rate, to avoid it to vary with other parameters



□ Ext. clk □ lock smp rate

Read sample rate (Hz) 4000000.000

	Physically		Value used by PMT
1	-0.095	10.00(	[-10.0, 10.0] V
2	-0.171	10.00(	[-10.0, 10.0] V
3	-0.095	10.00(	[-10.0, 10.0] V
4	-0.095	10.00(	[-10.0, 10.0] V

Dev. to use AI 1 : NIDAQ 6110

□ img = reinit AI	1	2	3	4
Gain (V)	-	-	-	-
BW	H	H	H	H
Pre-amp	10^7	10^7	10^7	10^7
				mW

Sample rate for acquisition

Min and max values physically given by the PMT

Value used by daq card for analog input bounds, that determines the precision (for PMT 1, 2, …)

NI card to use for acquisition (2 choices)

Check to re-init Tasks of card at each images

Gain on amplifier of PMT

Anode amplifier carac

Input optical power

LAST TAB

Draw a rectangle on current image to define a ROI

Define the drawn ROI in scan parameters

4c) Galvos, anlg & dig

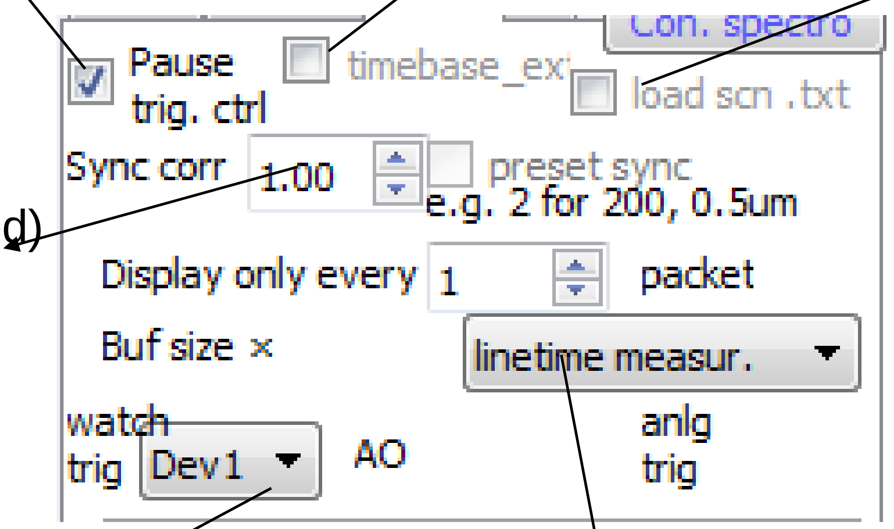
Use a pause trigger as the tool to extract the image lines from buffer

Use external timebase, e.g. the 8MHz from digital galvos,(or the lock-in?)

Load galvos scan in .txt after Labview

Factor to correct the lines (imposed)

Dig galvos



☑ Pause trig. ctrl

□ timebase_ex

Con. spectro

□ load scn .txt

Sync corr 1.00

□ preset sync

e.g. 2 for 200, 0.5um

Display only every 1 packet

Buf size ×

linetime measur.

watch trig Dev1

AO

anlg trig

Device to use for trigger

Do the images by linetime measurement (1) or by callback (2, useful for 1S-ISHG)

Reduce the number of display to enhance speed

Factor to multiply the allocation of AI buffer

Device to use for (digital) trigger the fill Process

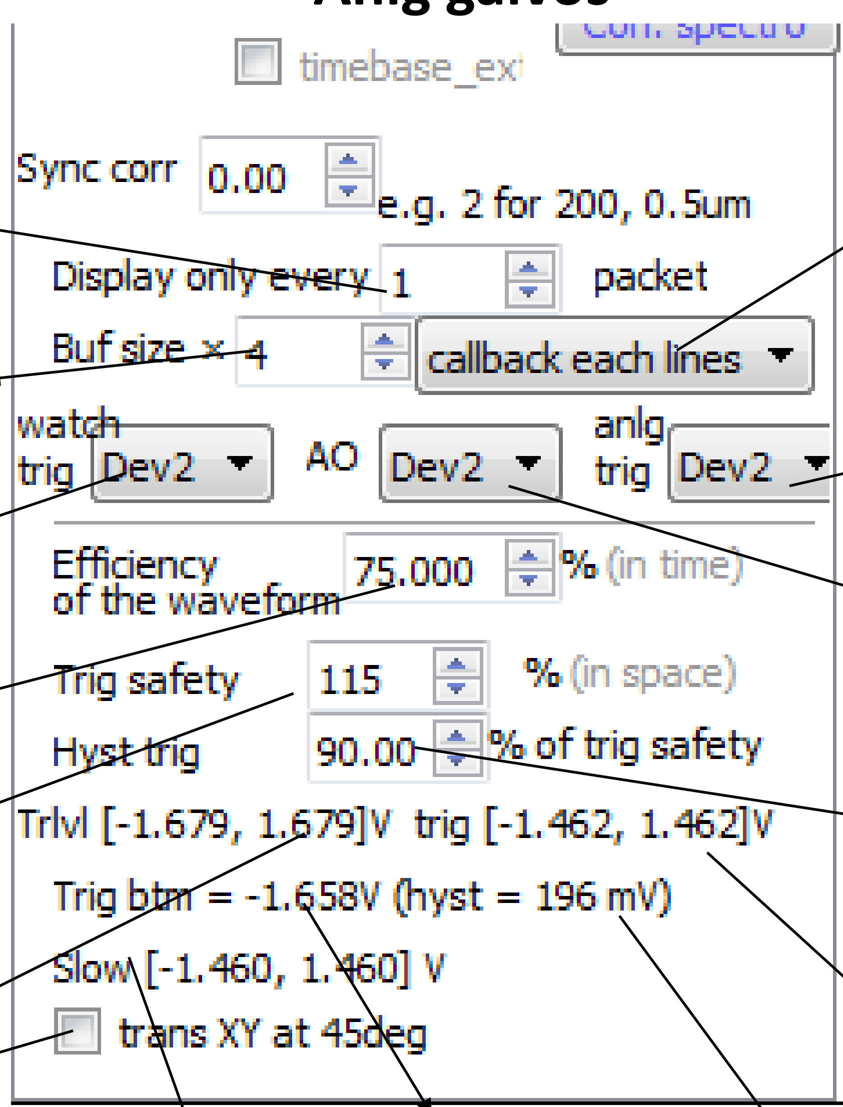
Efficiency (time spent in flyback for unidirectional)

Safety of the trigger (scan goes farther than theoretical)

Voltage reached by the motor

Translate motor at 45° from its axis, following the frame of the galvos

Anlg galvos



□ timebase_ex

Con. spectro

Sync corr 0.00

e.g. 2 for 200, 0.5um

Display only every 1 packet

Buf size × 4

callback each lines

watch trig Dev2

AO Dev2

anlg trig Dev2

Efficiency of the waveform 75.000 % (in time)

Trig safety 115 % (in space)

Hyst trig 90.00 % of trig safety

Trlvl [-1.679, 1.679]V trig [-1.462, 1.462]V

Trig btm = -1.658V (hyst = 196 mV)

Slow [-1.460, 1.460] V

□ trans XY at 45deg

Do the images by linetime measurement (1) or by callback (2, useful for 1S-ISHG)

Anlg trigger for AI Task

Device used to write scan using AO Task

% of trig safety used for hysteresis

Voltage effectively seen by trigger

Voltage of slow direction

Level at which the trigger asserts

Hysteresis level for rejecting oscillations

- Refer to the “New galvos control_summary.pptx” in the galvos docs, or the thesis, for more details

5) GUI: image display

Save selected images

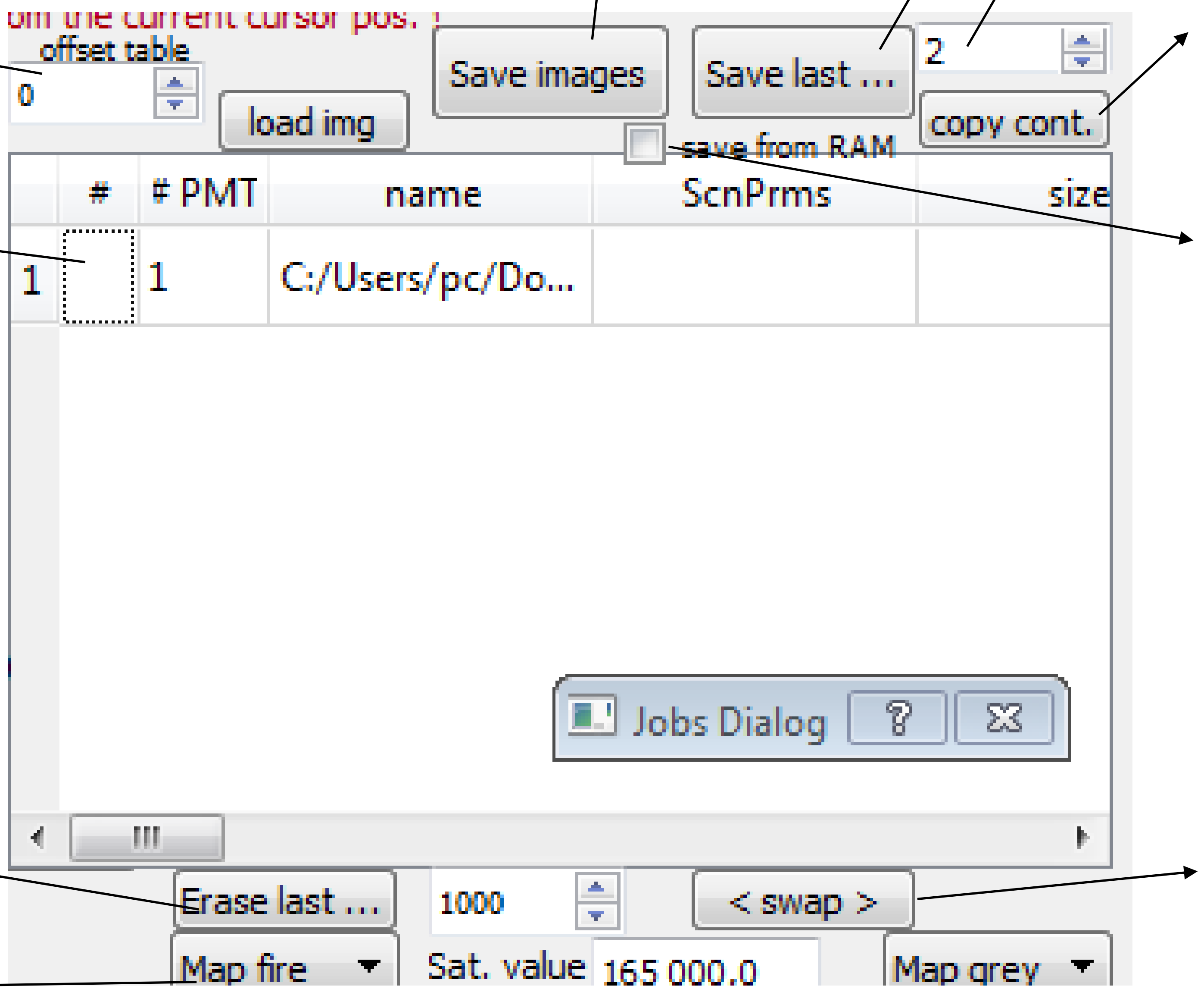
Save N last images

Offset for the # of image in table

Copy content to the PC

Description of the image that has been acquired

Save from RAM in Python, instead of copying from “tmp” folder



offset table: 0

Save last: 2

	#	# PMT	name	ScnPrms	size
1		1	C:/Users/pc/Do...		

1000

Sat. value 165 000.0

Put current image at right (if left) or at left (if right)

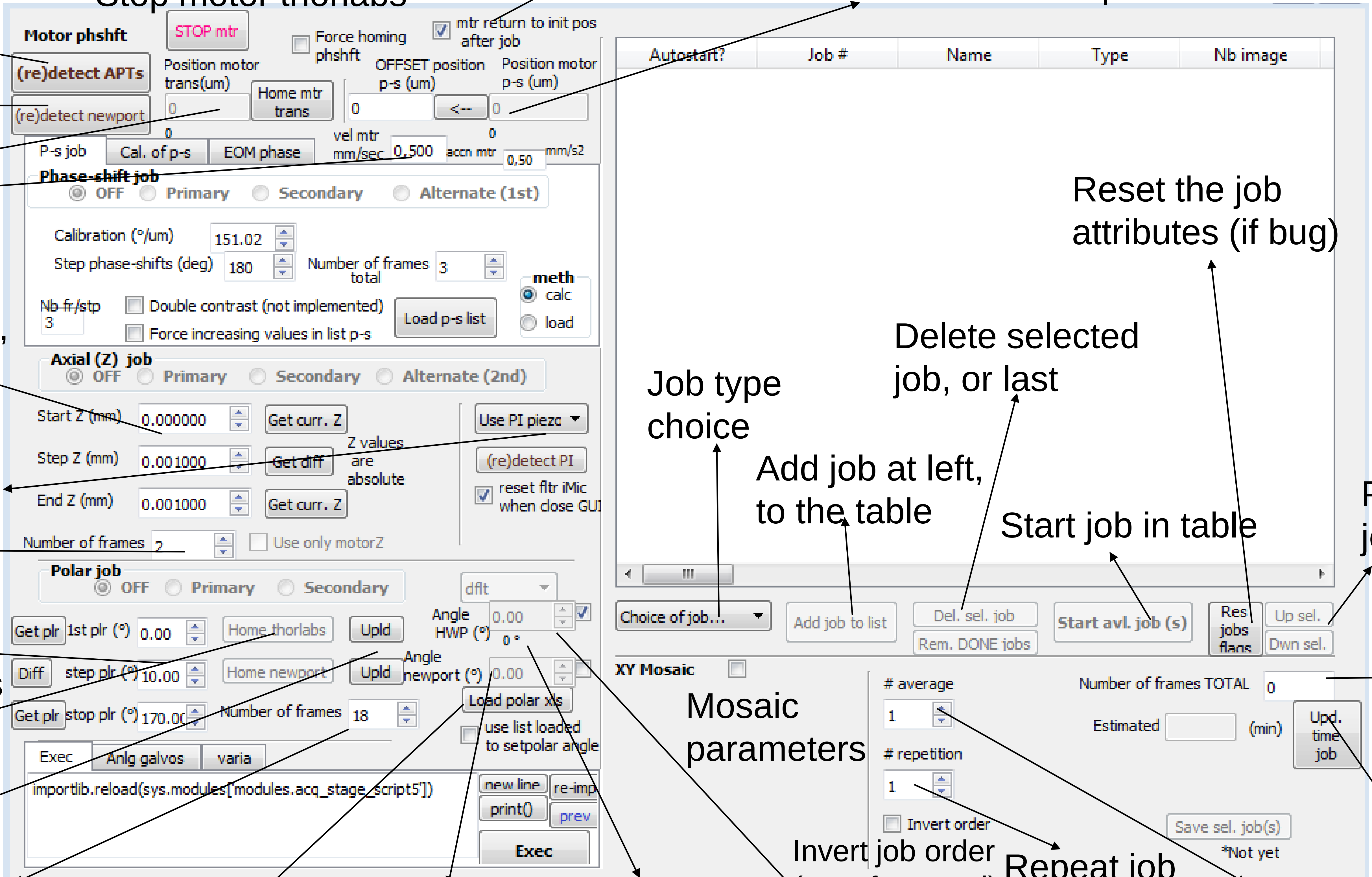
Erase entries in table

Colormap for display

Saturation value (if buffers are summed, and not averaged) of the display

Colormap of left

6) GUI: jobs



Motor phshft

☐ Force homing phshft

☑ mtr return to init pos after job

Position motor trans(um): 0

OFFSET position p-s (um): 0

Position motor p-s (um): 0

0

0

vel mtr mm/sec: 0,500

accn mtr: 0,50 mm/s2

P-s job | Cal. of p-s | EOM phase

Phase-shift job
● OFF ○ Primary ○ Secondary ○ Alternate (1st)

Calibration (°/um): 151.02

Step phase-shifts (deg): 180

Number of frames total: 3

meth: ● calc ○ load

Nb fr/stp: 3

☐ Double contrast (not implemented)

☐ Force increasing values in list p-s

Axial (Z) job
● OFF ○ Primary ○ Secondary ○ Alternate (2nd)

Start Z (mm): 0.000000

Step Z (mm): 0.001000

End Z (mm): 0.001000

Z values are absolute

☑ reset fltr iMic when close GUI

Number of frames: 2

☐ Use only motorZ

Polar job
● OFF ○ Primary ○ Secondary

dflt

Angle HWP (°): 0.00 ☑ 0 °

1st plr (°): 0.00

step plr (°): 10.00

Angle newport (°): 0.00 ☐

stop plr (°): 170.00

Number of frames: 18

☐ use list loaded to setpolar angle

Exec | Anlg galvos | varia

```
importlib.reload(sys.modules['modules.acq_stage_script5'])
```

Autostart?	Job #	Name	Type	Nb image

XY Mosaic ☐

average: 1

repetition: 1

☐ Invert order

Number of frames TOTAL: 0

Estimated (min)

*Not yet

Stop motor thorlabs

return to init position after job of phase-shift

Position motor phase-shift

Detect Thorlabs cube motors

Detect Newport motor

Motor translation

Velociy/accn of motor phase-shift

Z motor job: start, step and stop positions

Detect PI piezo (Z)

Number of frames for Z jobs

Polar start, step, and stop positions

Upload position of waveplates

Nb of frames of polar job

Load polar angles, HWP+QWP

Position of QWP (Newport)

Live position of HWP

Change HWP (Thorlabs) position

Invert job order (start from end)

Repeat job N times

Number of average in jobs

Upload estimation of job duration (automatic normally)

estimation of job duration (automatic normally)

Put selected job up or down

Reset the job attributes (if bug)

Delete selected job, or last

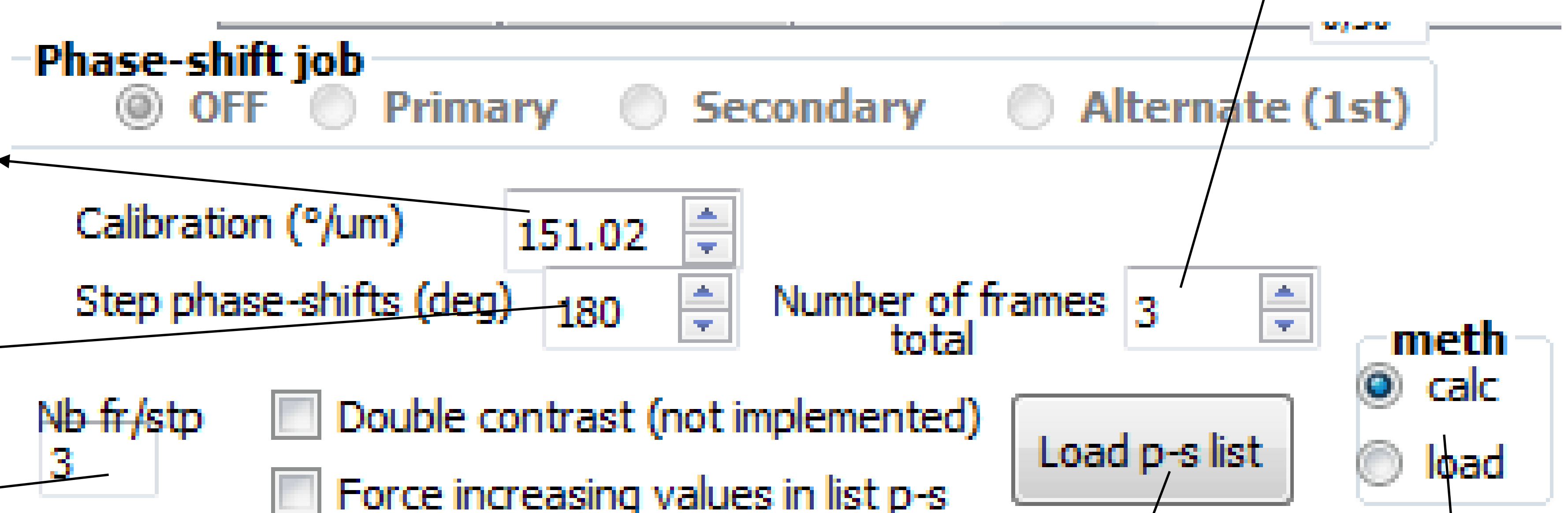
Job type choice

Add job at left, to the table

Start job in table

Mosaic parameters

6a) GUI: jobs



Phase-shift job

● OFF ○ Primary ○ Secondary ○ Alternate (1st)

Calibration (°/um) 151.02

Step phase-shifts (deg) 180

Number of frames total 3

Nb fr/stp 3

□ Double contrast (not implemented)

□ Force increasing values in list p-s

meth

● calc

○ load

Number of frames in this job of phase-shifts

Calibration measured/theoretical

Phase-shift steps

Nb of phase-shift per step (e.g. 3 for 0,180,360…), if phase-shift job per step (slow)

Re-order phase-shifts, to have a purely increasing order (e.g. 0,90,180,270….)

Load list of phase-shifts in .txt file

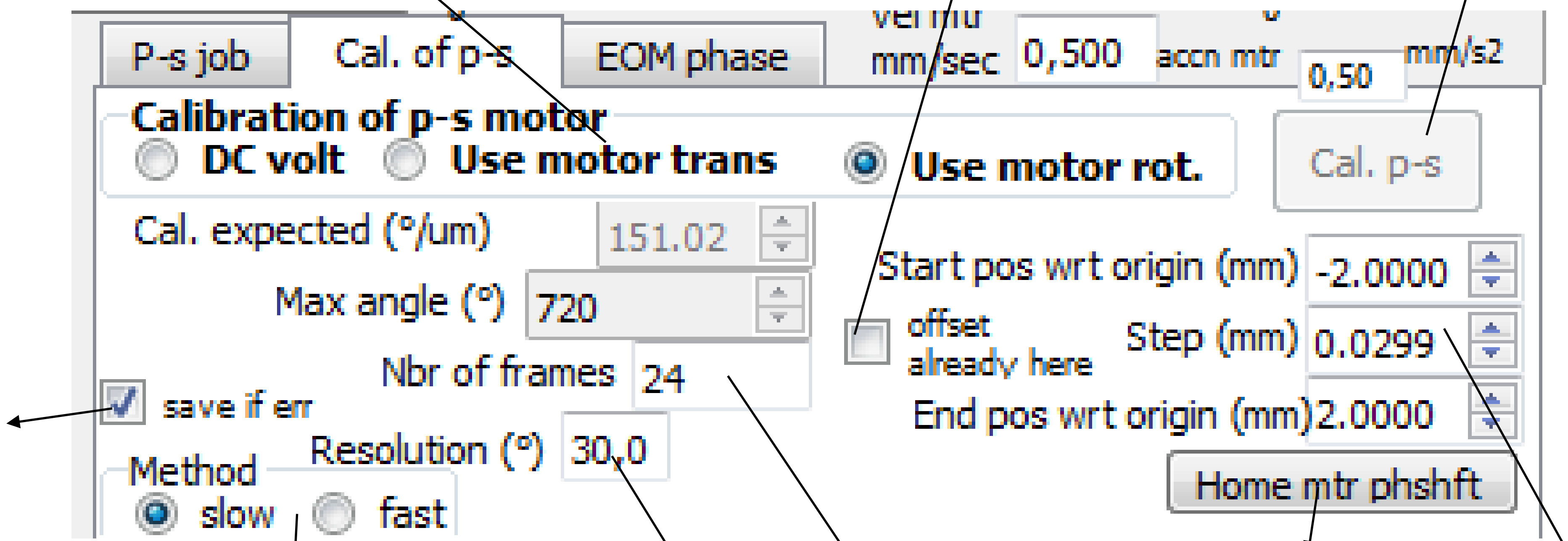
Method: loaded txt file (via button), or calculation with calibration button

6b) GUI: jobs

Motor for phase-shift, or calib

If checked, don't translate motor to compensate acceleration for calib

Start calib



vel mtr mm/sec 0,500 accn mtr 0,50 mm/s2

Calibration of p-s motor

○ DC volt ○ Use motor trans ● Use motor rot.

Cal. expected (°/um) 151.02

Max angle (°) 720

Nbr of frames 24

☑ save if err

Resolution (°) 30,0

Method ● slow ○ fast

Start pos wrt origin (mm) -2.0000

□ offset already here

Step (mm) 0.0299

End pos wrt origin (mm) 2.0000

Save calib if error during acquisition

Save only treated interferograms, or whole arrays, or both

Calib slow (step-by-step), or fast (one translation)

Resolution of calib

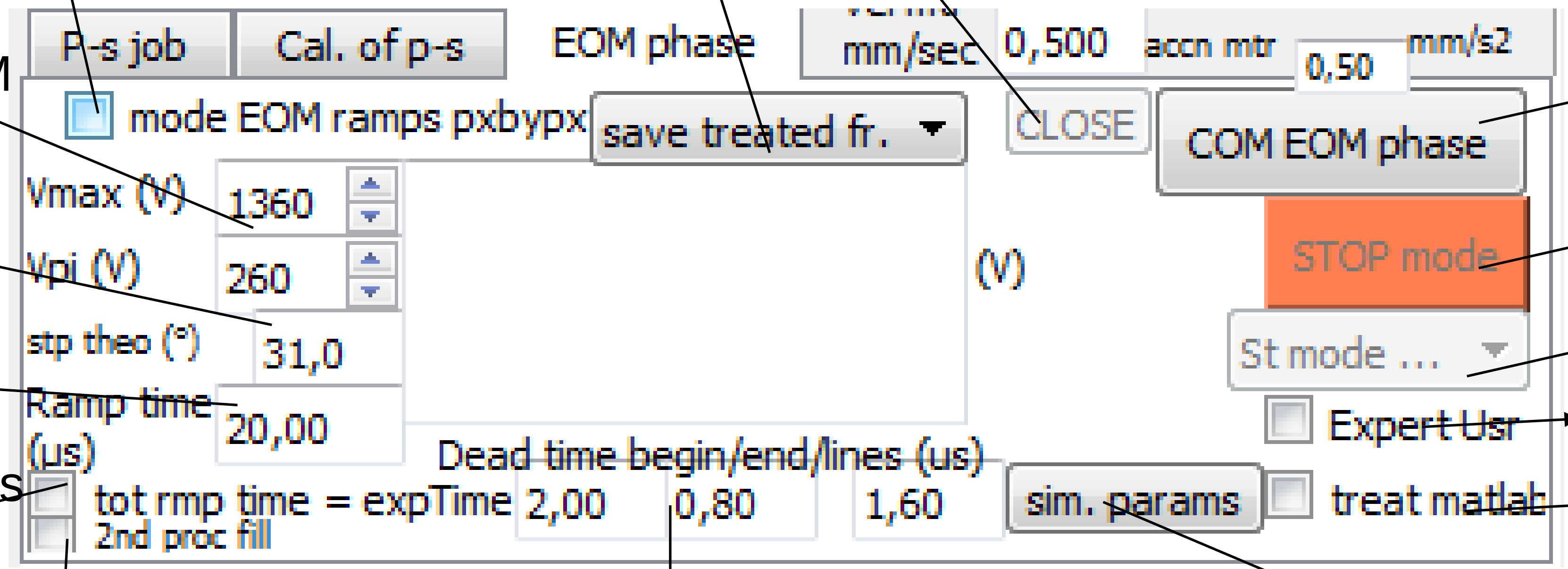
Number of frames in calib

Home motor phase-shift (optional)

Start/step/stop positions for calib

Acquire with EOM phase treatment (even if no mode is started

Close EOM



vel mtr mm/sec 0,500 accn mtr 0,50 mm/s2

□ mode EOM ramps pxbypx save treated fr. ▾

Vmax (V) 1360

Vpi (V) 260

stp theo (°) 31,0

Ramp time (us) 20,00

(V)

St mode ... ▾

□ Expert Usr

Dead time begin/end/lines (us) 2,00 0,80 1,60

□ tot rmp time = expTime

□ 2nd proc fill

□ treat matlab

Voltages of EOM (Vpi and max)

Step of phase-shift

Ramp time of EOM mode

Impose EOM mode as pixel exposure time

Start a 2nd Process to fill buffer for EOM

Dead-times of mode EOM

Display parameters of EOM scan

Connect phase EOM

Stop EOM mode

Start EOM mode

Show advanced buttons of EOM

Treat ps job in MatLab

6c) GUI: jobs

Zone to input text, to be executed in Python's console in live (to modify code, or reload a script)

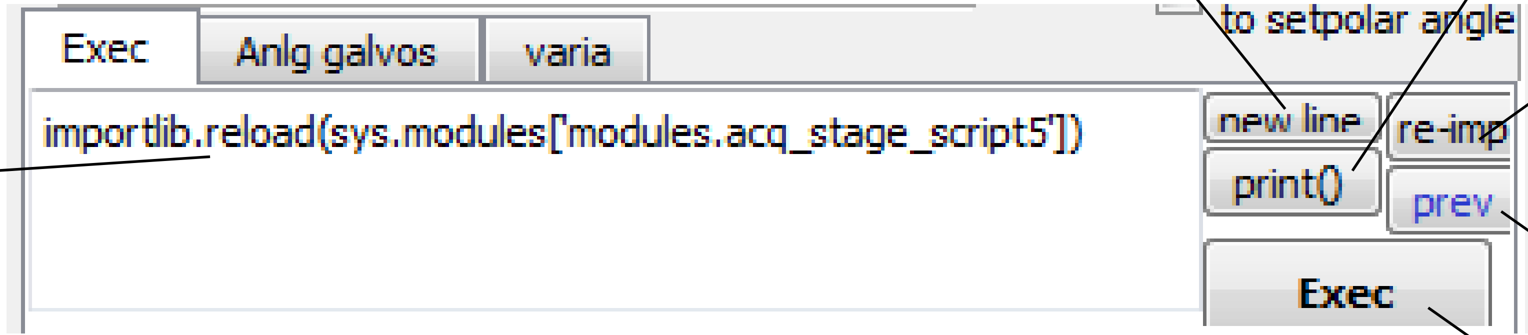
Break a line in the plain Text

Put print balise in widget's text

Put the text to reimport module in PlainText

Display previous commands

execute command in PlainText



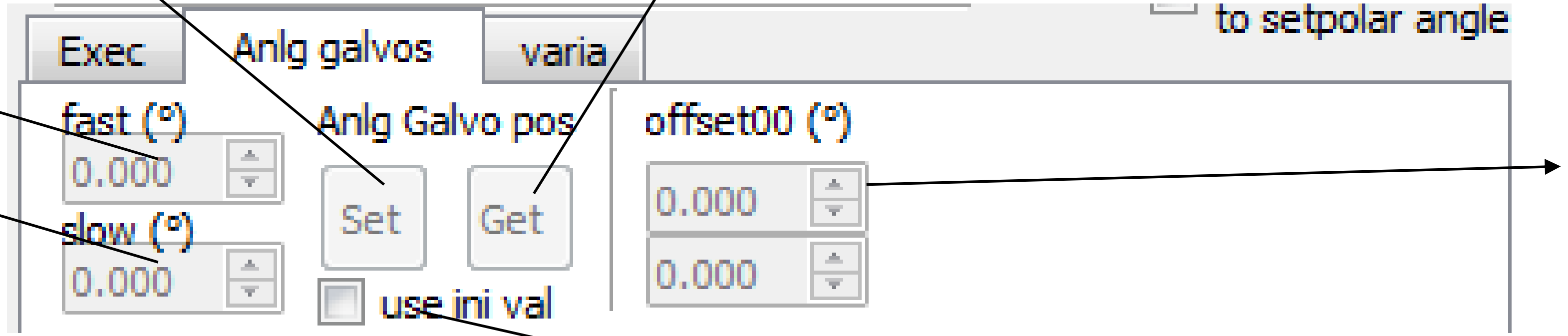
```
importlib.reload(sys.modules['modules.acq_stage_script5'])
```

Fast/slow position of anlg galvos, to set

Set positions of widget

Get positions of galvos, by plugging cables

Offset of anlg glavos



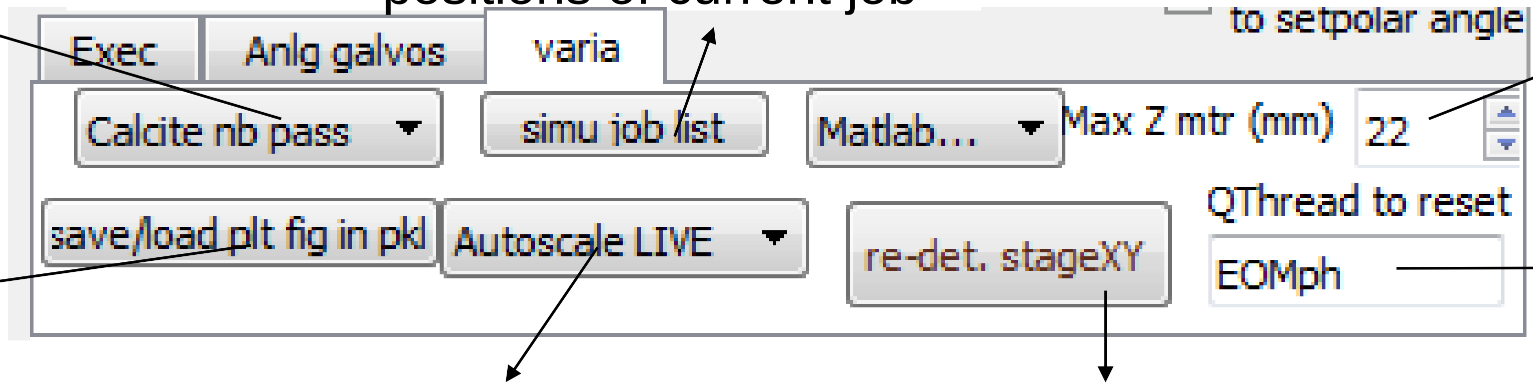
Check to use values saved in txt file (previous)

For calib autoco, number of pass in calcite (for calculations)

Display (in logger) the list of positions of current job

maximum value of Z for objective (can limit to a smaller value to avoid hitting the sample)

Save plotted figure in pkl (left-click), or load a .pkl file (right-click)

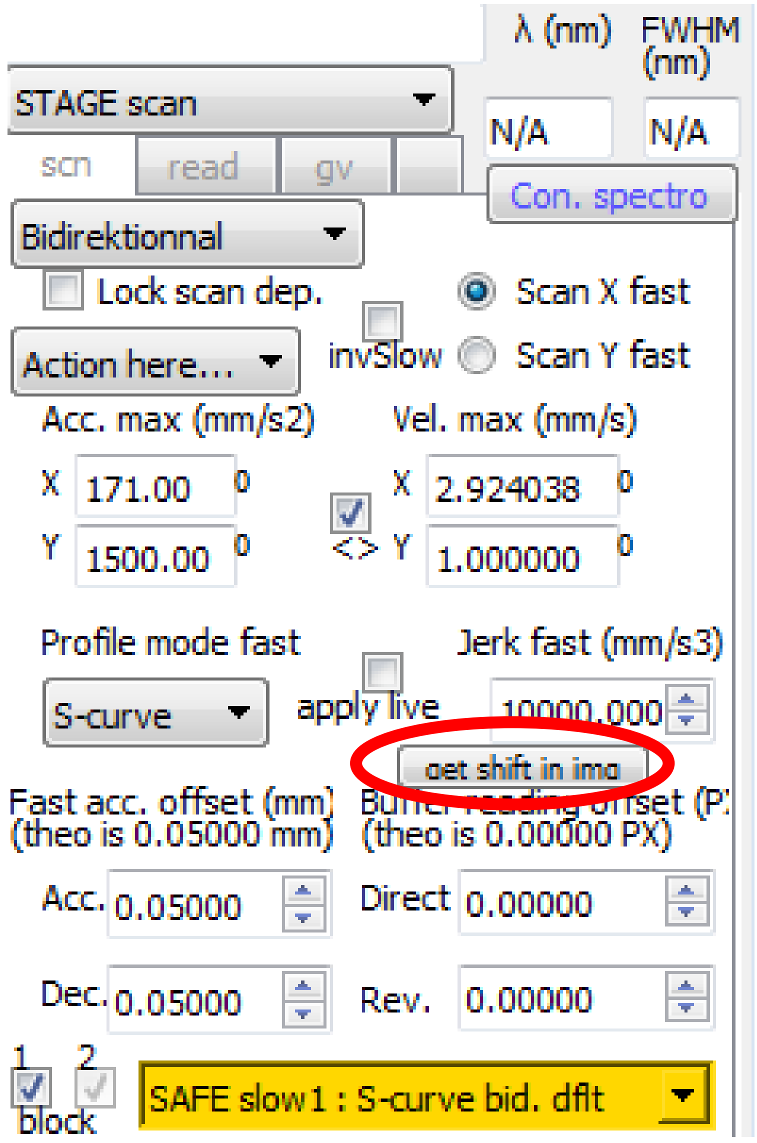


By pressing 'Enter', put widget's text in PlainText for reloading a QThread: shutter, apt, …

Autoscale LIVE, or 90% of dynamic range, or 1st buffer

(re)detect stage Thorlabs XY (if was closed for bug)

Get accurate shift lines in bidirek stage scan



λ (nm): N/A
FWHM (nm): N/A

Bidirektionnal

☐ Lock scan dep.

● Scan X fast
○ Scan Y fast

☐ invSlow

Acc. max (mm/s2)
X 171.00
Y 1500.00

Vel. max (mm/s)
X 2.924038
Y 1.000000

☑ <>

Profile mode fast: S-curve

☐ apply live

Jerk fast (mm/s3): 10000.000

Fast acc. offset (mm) (theo is 0.05000 mm)
Acc. 0.05000
Dec. 0.05000

Buffer reading offset (P (theo is 0.00000 PX)
Direct 0.00000
Rev. 0.00000

1 ☑ 2 ☑ block

SAFE slow1 : S-curve bid. dflt

DIPIMAGE

You could use get shift lines in Python, which uses Skimage. But it was showed to fail for large shifts.

Instead, in Matlab you can use DIPimage (install from website before). Using ‘reg_shift_advanced_func.m’ , the reg is done ~5 times to get accurate result.

```
[shiftv, im] = reg_shift_advanced_func(im0, lim1, off_shift, [])
%
% im0 image to treat (array of numbers), [] if load
% lim1 limit the ROI to X first lines (sizeY dflt)
% off_shift : final imposed offset (0 dflt)
% % !!! transpose the image to find shift on Y (here it's on X) !!!
% shift00 is [] dflt, unless to just shift
```

→ shiftv contains the shifts wanted. Works on X direction, for Y transpose image first !

→ It plots the result.