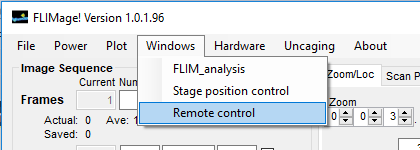
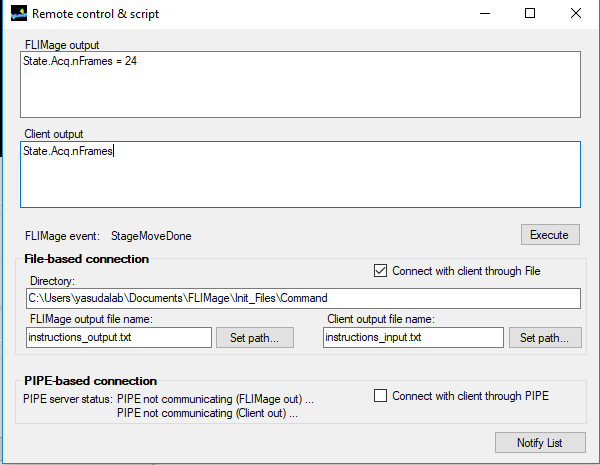
Remote control can be done either through text file or PIPE.

For controlling this feature, click “Windows menu” and then “Remote control” (see below).



You will see a window named “Remote control & script” pops up (see below).



Text file based communication is much easier than PIPE-based communication. You simply use files for output and input to communicate. For now, both output and input instructions need to be in the same folder. Press “set path” to select a folder and files for communication. The line must be appended to the last line of the file created. In C#, it will be something like:

File.AppendAllText(FilePath, Command + "\r\n");

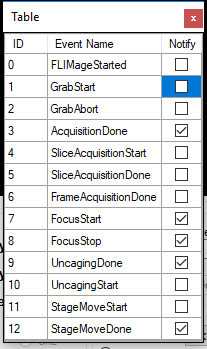
For opening of command file, make sure to use read-only mode.

If you want to use PIPE, turn on “Connect with client through PIPE”, and then turn on client PIPE. Handshaking and so on will be done through PIPE server. A example script for will be found in the folder (FLIM\_pipeClient.py). For now, PIPE name is set to FLIMageW (for client to write) and FLIMageR (for client to read). Immediately after establishing connection, FLIMage will send a code “FLIMage”. Client needs to return “FLIMage” to make sure the connection. PIPE will be 1 command – to 1 response communication.

One can simulate sending commands from client by typing commands in “Client output” window, and press “Execute”.

# Event notification from FLIMage to Client

To view the event notifications from FLIMage, press "Notify List" button at the bottom in the "Remote control & script" window:



-- sent after FLIMage start ---

command: FLIMageStarted

arguments: none

-- set after Grab start --

command: GrabStart

arguments: none

--sent after movement to new coordinates is complete--

command = StageMoveDone

arguments: x, y, z

example: StageMoveDone,1,34.2,-2

--sent after slice acquisition is done

Command = SliceAcquisitionDone

arguments; {current slice}, {total number of slices}

# Execution command from Client to FLIMage

In general, there will be no return. It will execute command.

--Move to new stage position--

command = SetMotorPosition

arguments: x y z

example: SetMotorPosition,12,89.2,0

--Start grab --

command = StartGrab

--Start loop –

(Use parameters in “Loop Sequence” panel)

command = StartLoop

--Stop loop –

command = StopLoop

--Abort grab—

command = AbortGrab

--Start uncaging—

command = StartUncaging

arguments: ROI\_x\_pixels, ROI\_y\_pixels

example: StartUncaging,37,42

return: none

# Parameter request (set, get) from Client to FLIMage

You can access to all parameters by typing the parameter name (see FLIMage\default.txt):

Example: State.Acq.nFrame

Return exmaple: State.Acq.nFrame = 24

All parameters can be set by inserting values as follows:

Example: State.Acq.nFrame = 24

Return: State.Acq.nFrame = 24

## Other parameter requests:

--Load setting file

Command: LoadSetting

Arguments: file number or file name

Example: LoadSetting, 1

Example2: LoadSetting, C:\Users\yasudalab\Documents\FLIMage\Init\_Files\Uncaging\_imaging.txt

Example3: LoadSetting, Uncaging\_imaging.txt

Return: Setting, file number or file name

Note: If there is it is not a full path, it will try default path (…. FLIMage\Init\_Files\)

--Set Zoom to specific value

command = SetZoom

arguments: zoom

example: setZoom,20

return: Zoom, {zoom value}

--Request for current position to be sent--

command = GetCurrentPosition

arguments: None

return: CurrentPosition, {x}, {y}, {z}

--Request for FOV size--

command = GetFOVXY

arguments: None

return: FovXYum, {x value}, {y value}

--Request Galvo Scanning Position--

command = GetScanVoltageXY

arguments: None

example: GetScanVoltageXY

return: ScanVoltageXY, {x value}, {y value}

--Set Galvo Scanning Position

command = SetScanVoltageXY

arguments: x, y (in voltage)

example: SetScanVoltageXY,0.2,-4

return: ScanVoltageXY, {x value}, {y value}

--Request Galvo Scan Voltage Multiplier

command = GetScanVoltageMultiplier

arguments: None

return: ScanVoltageMultiplier, {x value}, {y value}

--Request Galvo Scan Voltage Range Reference

command = GetScanVoltageRangeReference

arguments: None

return: ScanVoltageRangeReference, {x value}, {y value}

--Set Number of Z Slices

command = SetZSliceNum

arguments: zSliceNum

example: SetZSliceNum,10

return: ZSliceNum, {z slice number}

--Set X,Y Imaging Resolution

command = SetResolutionXY

arguments: x,y

example: SetResolutionXY,128,128

return: ResolutionXY, {x pixel number}, {y pixel number}

--Get X,Y Imaging Resolution

command = GetResolutionXY

arguments: None

example: GetResolutionXY

return: ResolutionXY, {x pixel number}, {y pixel number}

--Convert pixel into voltage

command = PixelToVoltage

arguments: xpixel, ypixel

example: PixelToVoltage, 64, 64

return: PixelToVoltage, 0, 0

-- Turn Intensity Image saving On/Off

Command = SetIntensitySaving

Arguments: 0 or 1 (for on or off)

Return: IntensitySaving, {0 or 1}

--Request Image File Path

Command = GetIntensityFilePath

Arguments: None

Return: IntensityFilePath, {file path}

--Set channels to be saved--

Command=SetChannelsToBeSaved

Arguments: {channel to be saved}

Return: ChannelsToBeSaved, {channel to be saved}

Note: if channel == -1 (default), all channels will be saved.

--Ask if images are acquired:

Command = IsGrabbing

Return 0 or 1

--Ask if uncaging is going:

Command = IsUncaging

Return 0 or 1

--Ask if digital output is going:

Command = IsDORunning

Return 0 or 1

## Commands for analysis (Useful for batch processing):

--Open FLIM file—

Command = “OpenFIle

Arguments: {Full path to Filename}

Return: None

Note: Open .flim file and display

-- Read ImageJ format ROIs—

Command = ReadImageJROI

Arguments: {Filename}

Return: None

Note: Open ImageJ format ROI file (zip file)

--Bin Pages/Frames—

Command = BinFrames

Arguments: {Number of frames to be binned}

Return: None

Note: Bin frames or pages to increase signal-to-noise ratio.

Same as “Bin time course…” menu.

--Calculate time course—

Command = CalcTimeCourse

Arguments: None

Note: Same as pressing button of “Calc

--Set FLIM intensity ranges—

Command = SetFLIMIntensityOffset

Arguments: Lower threshold, Higher threshold

Example: SetFLIMIntensityOffset, 20, 60

--Fix Tau for fitting—

Command = FixTau

Arguments: Tau1, Tau2

Example: FixTau, 2.6, 1.1

--Set fit range--

Command = SetFitRange

Arguments: StartPoint, EndPoint (in time point)

Example: SetFitRange, 5, 60

--Fix tau all—

Command = FixTauAll

Arguments: 0 or 1

Note: Fix or unfix all tau to the current values.

--Fit data--

Command = FitData

Arguments: none

Note: it will fit the fluorescence lifetime curve with current setting.

--Set Channel—

Command = SetChannel

Arguments: 1, 2, or 12

Note: Set display channel.

--Apply fit offset—

Command = ApplyFitOffset

Arguments: none

Note: It will fit the data and then adjust the offset of FLIM image display.

--Fit each frame—

Command = FitEachFrame

Arguments: 0 or 1 (for off or on)

Example: FitEachFrame, 1

Note: If turned on, it will fit fluorescence lifetime curve for each frame.

--Align frames—

Command = AlignFrames

Arguments: none

--Get number of pages of the opened file.—

Command = GetNPages

Return: number of pages of the opened file.

# Using python example code for PIPE

You can run attached FLIM\_pipeClient.py class.

Once you run it you can:

exec(open("FLIM\_pipeClient.py").read())

value = flim.sendCommand(‘State.Acq.msPerLine?’) #Return ‘2’

flim.sendCommand(‘State.Acq.msPerLine = 2’) #Change msPerline to 2 ms.

flim.sendCommand(‘StartGrab’) #Will grab.

flim.sendCommand(‘SetMotorPosition,12,89.2,0’) #set motor position.