Using Tkinter of Python to create Graphical User Interfaces (GUI) for scripts in LNLS

Brazilian Synchrotron Light Laboratory

Beniz, D. B., douglas.beniz@lnls.br, Espindola, A. M., alexey.espindola@lnls.br

Brazilian Center for Research in Energy and Materials

Python is being widely used to create scripts which cover different necessities in computational scenario. At LNLS we successfully developed Python scripts to control beamlines operations, including a case of GUI (*Graphical User Interface*) creation using Tkinter, which is the standard GUI programming toolkit of Python, for one of our beamlines, DXAS (*Dispersive X-ray Absorption Spectroscopy*). Tkinter offers the basic components necessary to build a GUI that help users to quickly inform a set of parameters defining which device to use, its configuration to set, among others, and to easily start or stop operations.

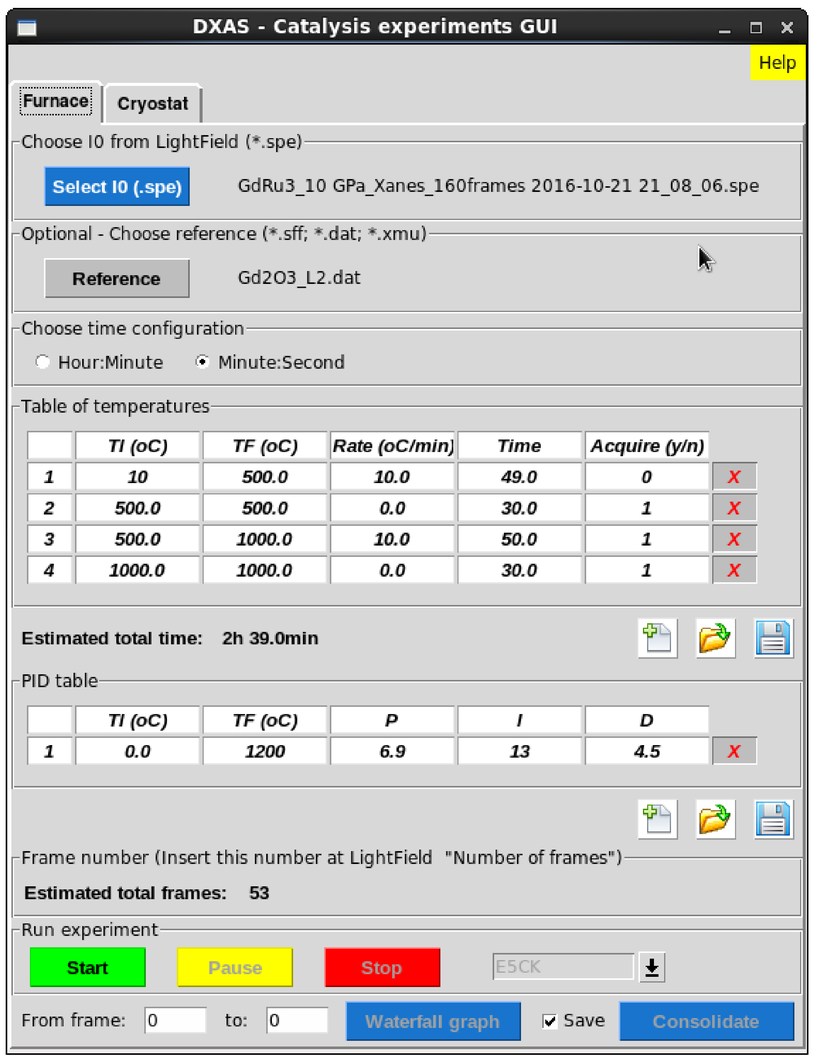
Why Python?

- Widely used in many different applications, including synchrotron laboratories
- Almost all LNLS beamlines have Python scripts being used to control their operations since we developed Py4Syn package to abstract EPICS (*Experimental Physics and Industrial Control System*) IOC (*Input/Output Controller*) to be used with that language facilities
- Easy integration to EPICS via PyEPICS library of CARS
- Perform mathematical calculations and data matrix manipulation, with NumPy
- Flexibility with many libraries that help to build graphical interfaces

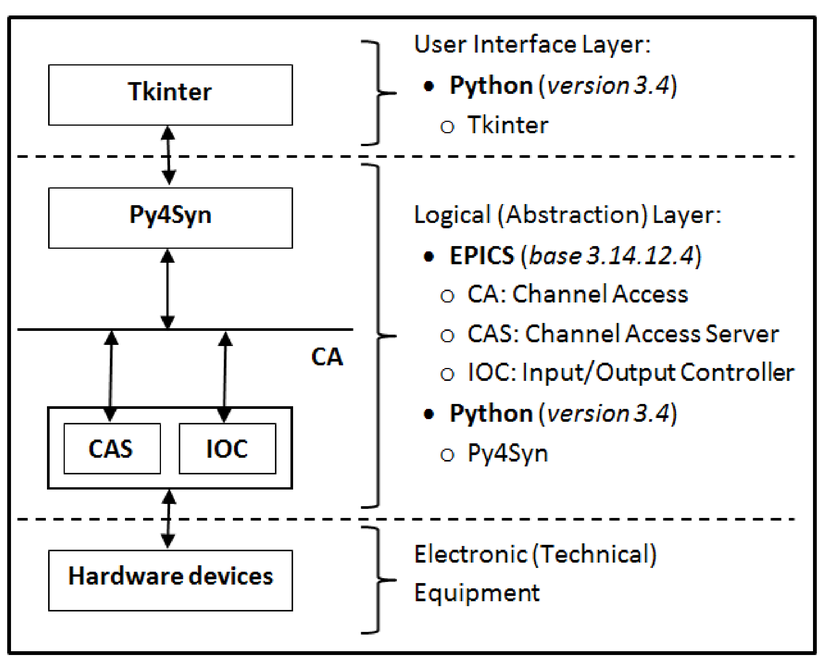
Why Tkinter?

- It is standard GUI (*Graphical User Interface*) package of Python
- Facility to build interfaces and large number of examples and tutorials in the Internet helping to work with it
- Widgets, geometry management and event handling are the three main Tkinter concepts and meet the needs

Tkinter Solution of DXAS



Architecture Overview



Source Code Excerpts

magnetic_dxas.py

```
236
237  # ------------------------------------------------------------
238  self.buttonsFrame = LabelFrame(self.master, text = 'Run experime
239  self.buttonsFrame.pack(side = TOP, fill = X, pady = 2, ipadx = 2
240  self.startButton = Button(self.buttonsFrame,
241                            command = self.StartExperiment,
242                            text = "Scan",
243                            font = textFont4,
244                            background = "green",
245                            fg="black",
246                            state = DISABLED,
247                            height = 1,
248                            width = 8)
249  self.startButton.pack(side = LEFT, anchor = W, padx = 10, pady =
250  self.stopButton = Button(self.buttonsFrame,
251                            command = self.StopExperiment,
252                            text = "Cancel",
253                            font = textFont4,
254                            background = "red",
255                            fg="white",
256                            state = DISABLED,
257                            height = 1,
258                            width = 8)
259  self.stopButton.pack(side = LEFT, anchor = W, padx = 10, pady =
```

*e5ck.py

```
231
232  def callback_curr_temp():
233      try:
234          self._currTemp.set(round(self.furnaceDevice.getValue(),
235          self._currPower.set(round(self.furnaceDevice.getPower()
236          self._currP.set(round(self.furnaceDevice.getP(),2))
237          self._currI.set(round(self.furnaceDevice.getI(),2))
238          self._currD.set(round(self.furnaceDevice.getD(),2))
239      except (ValueError, TypeError):
240          pass
241      global processRunning
242
243      # Updates total time
244      try:
245          totalFurnace = self.__get_total_program_time()
246          if (self._radioTime.get()==0):
247              totalFurnace = totalFurnace * 60
248          self.Furnace_Table._totalTime.set(str(round(totalFurnac
249      except:
250          pass
251
252      # And update it every 1 second later
253      self.master.after(1000, callback_curr_temp)
254  # Trigger of callback (point of return)
255  self.master.after(1000, callback_curr_temp)
256
```

PCaPAC 2016

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES

BRASIL GOVERNO FEDERAL