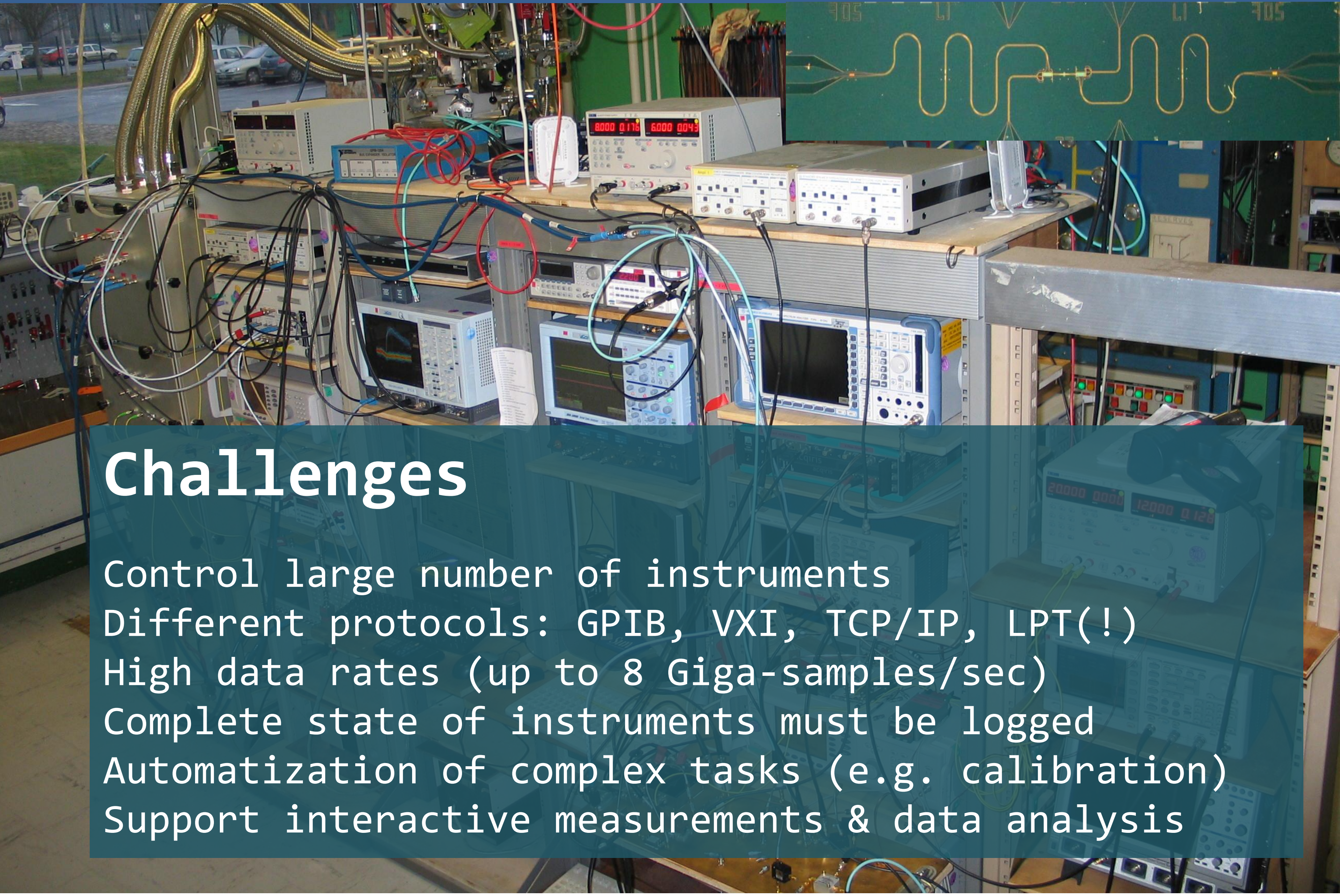
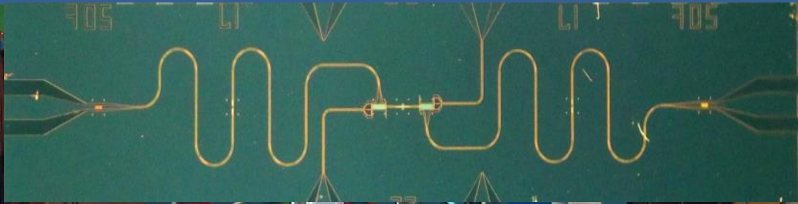
Interactive Data Acquisition & Data Analysis with Python

Andreas Dewes

DataRave – 2014/04/02

The Setup

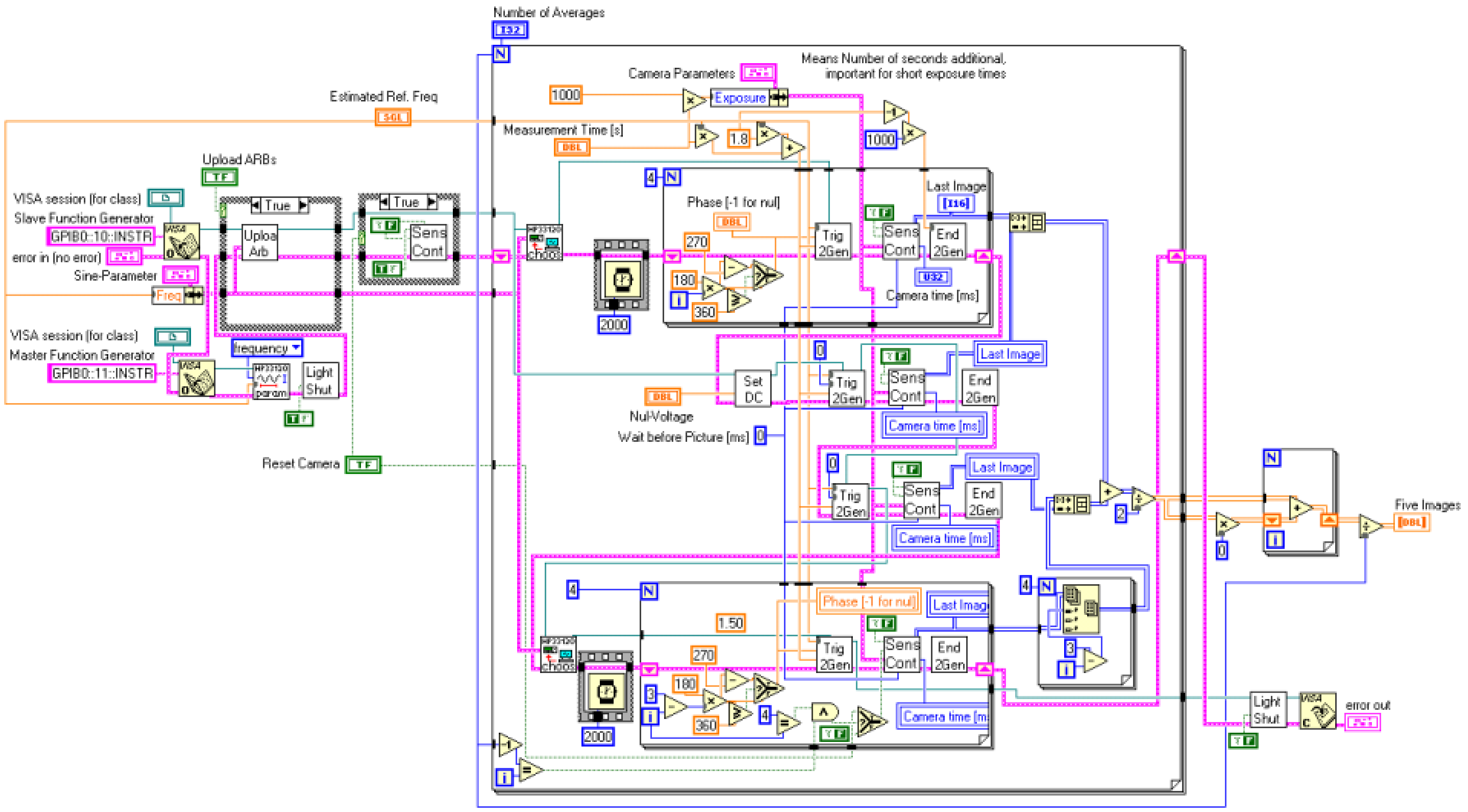




Challenges

Control large number of instruments
Different protocols: GPIB, VXI, TCP/IP, LPT(!)
High data rates (up to 8 Giga-samples/sec)
Complete state of instruments must be logged
Automatization of complex tasks (e.g. calibration)
Support interactive measurements & data analysis

The Old Way: Labview



The New Way: Python ~~MVC~~-Framework

Instrument

Can represent a physical instrument (e.g. a microwave source) or a virtual one (e.g. a qubit)

Frontpanel

Displays instrument state and gets notified about state changes of the instrument

Controller

Interacts with instruments to perform measurements and analyze data, controls instrument state

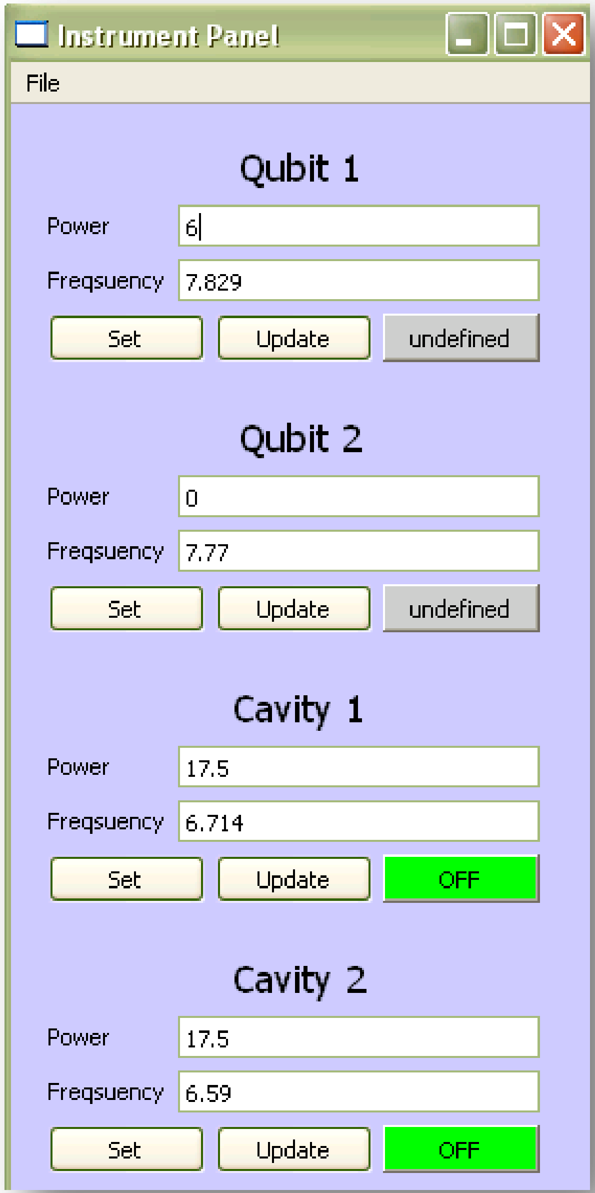
The Instrument

```python
class Yokogawa(VisaInstrument):

	"""Yokogawa (a voltage source) instrument class """

	def voltage(self):
		""" Returns the voltage """
		string = self.ask("od;")
		voltage = float(re.sub(r'^(NDCV|EDCV)',r'',string))
		self.notify("voltage",voltage)
		return voltage
```

Classes are instrument types, **instances** are actual instruments
Class inheritance: VisaInstrument -> MicrowaveSource -> …

Subject/Observer pattern for state changes
Dispatcher pattern for asynchronous commands & callbacks
Proxy pattern for controlling remote instruments

The Frontpanel



Qubit 1

Power	6
Freqsuency	7.829

Set | Update | undefined

Qubit 2

Power	0
Freqsuency	7.77

Set | Update | undefined

Cavity 1

Power	17.5
Freqsuency	6.714

Set | Update | OFF

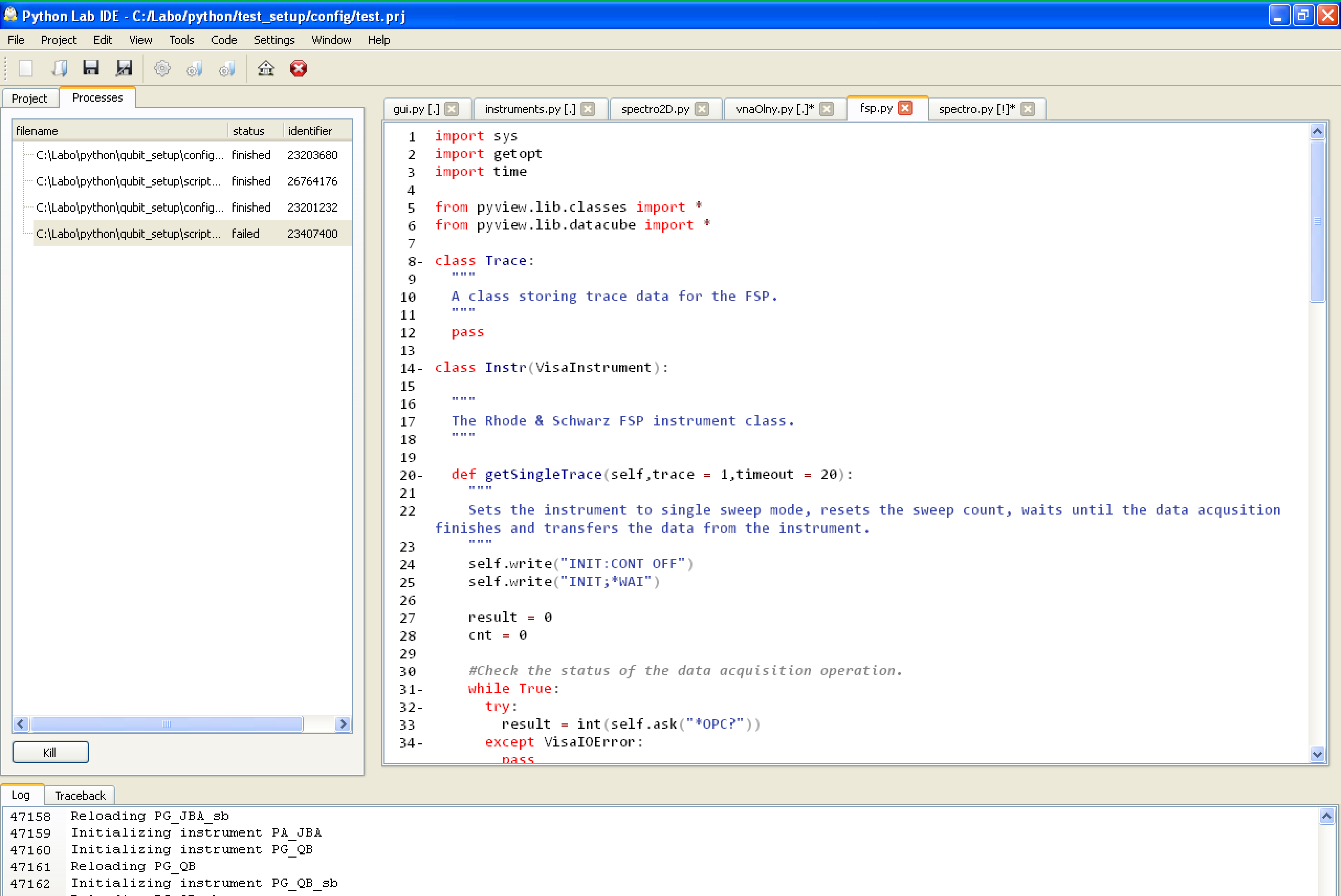
Cavity 2

Power	17.5
Freqsuency	6.59

Set | Update | OFF

```
class Panel(FrontPanel):

 #create GUI elements
 def __init__(self,instrument,parent=None):
    (Panel,self).__init__(instrument,parent)

    self.title = QLabel(instrument.name())
    self.UpdateButton = QPushButton("Update")

    self.connect(self.UpdateButton,
                 SIGNAL("clicked()"),
                 self.updateValues)

  #dispatch commands to the instrument
  def updateValues(self):
    self.instrument.dispatch("frequency")

  #react to status updates from instrument
  def onNotify(self,subject,property,value):
    self.UpdateButton.setText("…")
```

The Controller



```
filename                                status     identifier
C:\Labo\python\qubit_setup\config...   finished   23203680
C:\Labo\python\qubit_setup\script...   finished   26764176
C:\Labo\python\qubit_setup\config...   finished   23201232
C:\Labo\python\qubit_setup\script...   failed     23407400
```

```python
import sys
import getopt
import time

from pyview.lib.classes import *
from pyview.lib.datacube import *

class Trace:
  """
  A class storing trace data for the FSP.
  """
  pass

class Instr(VisaInstrument):

  """
  The Rhode & Schwarz FSP instrument class.
  """

  def getSingleTrace(self,trace = 1,timeout = 20):
    """
    Sets the instrument to single sweep mode, resets the sweep count, waits until the data acqusition finishes and transfers the data from the instrument.
    """
    self.write("INIT:CONT OFF")
    self.write("INIT;*WAI")

    result = 0
    cnt = 0

    #Check the status of the data acquisition operation.
    while True:
      try:
        result = int(self.ask("*OPC?"))
      except VisaIOError:
        pass
```

```
47158  Reloading PG_JBA_sb
47159  Initializing instrument PA_JBA
47160  Initializing instrument PG_QB
47161  Reloading PG_QB
47162  Initializing instrument PG_QB_sb
```

Challenges & Outlook

Challenges

Python can be slow (e.g. for plotting & processing data)
Global interpreter lock makes multi-threading inefficient
Interactivity is very difficult (**no pause button**)
„Hot-swapping“ of code is hard

New Technologies (2014)

ipython notebook: Removes need for custom IDE
ZeroMQ: Powerful library for „socket patterns“
Pandas: Better support for complex data analysis
Cython: Easy-to-use c extension toolkit for Python

Thanks!

Code:

General-purpose classes (IFC framework):

https://github.com/adewes/pyview

Instruments & frontpanels for qubit setup:

https://github.com/adewes/python-qubit-setup

Get in touch:

andreas.dewes@gmail.com