

UTILIZAÇÃO DE PYTHON PARA INSTRUMENTAÇÃO



if

CONTROLE DE EQUIPAMENTOS E AQUISIÇÃO DE DADOS

Fabício Frizera Borghi
Germano Maioli Penello
Jaime F. de Oliveira

SUMÁRIO

- Introdução
 - Motivação
 - Equipamentos
 - Tipos de comunicação
- Projeto: Caracterização de um LED
 - Comunicação com uma fonte de alimentação
 - Obtenção da curva de corrente por potência
- Referências

INTRODUÇÃO

Motivação

Equipamentos

Tipos de comunicação

INTRODUÇÃO

- Por que automatizar medidas?
 - Alto número de equipamentos para controle
 - Alto número de detectores para a coleta de dados
 - Automação das medidas
- Tipos de equipamentos que podem ser controlados
 - Praticamente todos com alguma porta de comunicação :)

PORTAS DE COMUNICAÇÃO (E SUAS BIBLIOTECAS)

- Ethernet – URLLib (request comum)
- RS232 → PySerial
- GPIB → PyVISA*





FONTE DE ALIMENTAÇÃO

Conexão física e
virtual

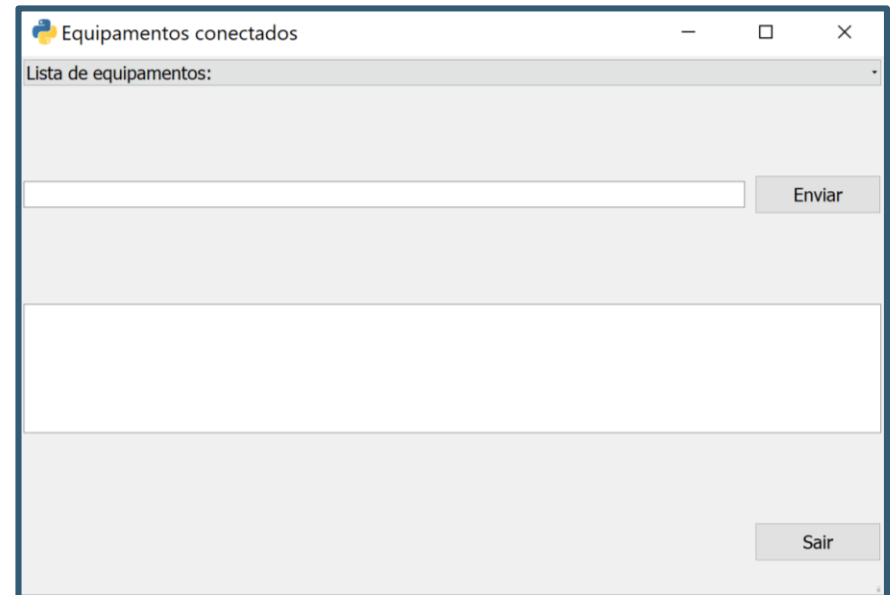
FONTE DE ALIMENTAÇÃO

- GPIB to USB (da National Instruments, driver)
- Biblioteca PyVisa
 - `import visa`
- Identificação
 - `rm = visa.ResourceManager()`
 - `equipamentos = rm.list_resources()`



FONTE DE ALIMENTAÇÃO

- Identificação e primeiros comandos (com interface)
 - `ListarEquipamentos.py`



FONTE DE ALIMENTAÇÃO

- `fonte = rm.open_resource(equipamentos[-1])`
- `fonte.timeout = 10000` # 10s para responder
- `fonte.read_termination = '\n'`
- `id = fonte.query("*IDN?")`

FONTE DE ALIMENTAÇÃO

MANUAL – Capítulo 4

Remote Interface Reference

*IDN?

Read the power supply's identification string. The power supply returns four fields separated by commas. The first field is the manufacturer's name, the second field is the model number, the third field is not used (always "0"), and the *fourth field* is a revision code which contains three numbers. The first number is the firmware revision number for the main power supply processor; the second is for the *input/output* processor; and the third is for the front-panel processor.

The command returns a string with the following format (be sure to dimension a string variable with at least *40 characters*):

Agilent Technologies,E3640A,0,X.X-Y.Y-Z.Z (*E3640A model*)

System-Related Commands

(see page 85 for more information)

```
DISPlay[:WINDow]
[:STATe] {OFF|ON}
[:STATe]?
:TEXT[:DATA] <quoted string>
:TEXT[:DATA]?
:TEXT:CLear
SYSTem
:BEEPer[:IMMediate]
:ERRor?
:VERSion?
:COMMunicate:GPIB:RDEvice:ADDRess <numeric value>
:COMMunicate:GPIB:RDEvice:ADDRess?
OUTPut
:RELay[:STATe] {OFF|ON}
:RELay[:STATe]?
[:STATe] {OFF|ON}
[:STATe]?
```

*IDN?

*RST

*TST?

FONTE DE ALIMENTAÇÃO

Ainda do manual, podemos controlar a fonte para fornecer 3 V

- `fonte.write("VOLTAge 3.000")`

Mas o LED não acendeu, pois a saída está fechada

- `fonte.write("OUTPut ON")`

Podemos medir a corrente que passa pelos componentes

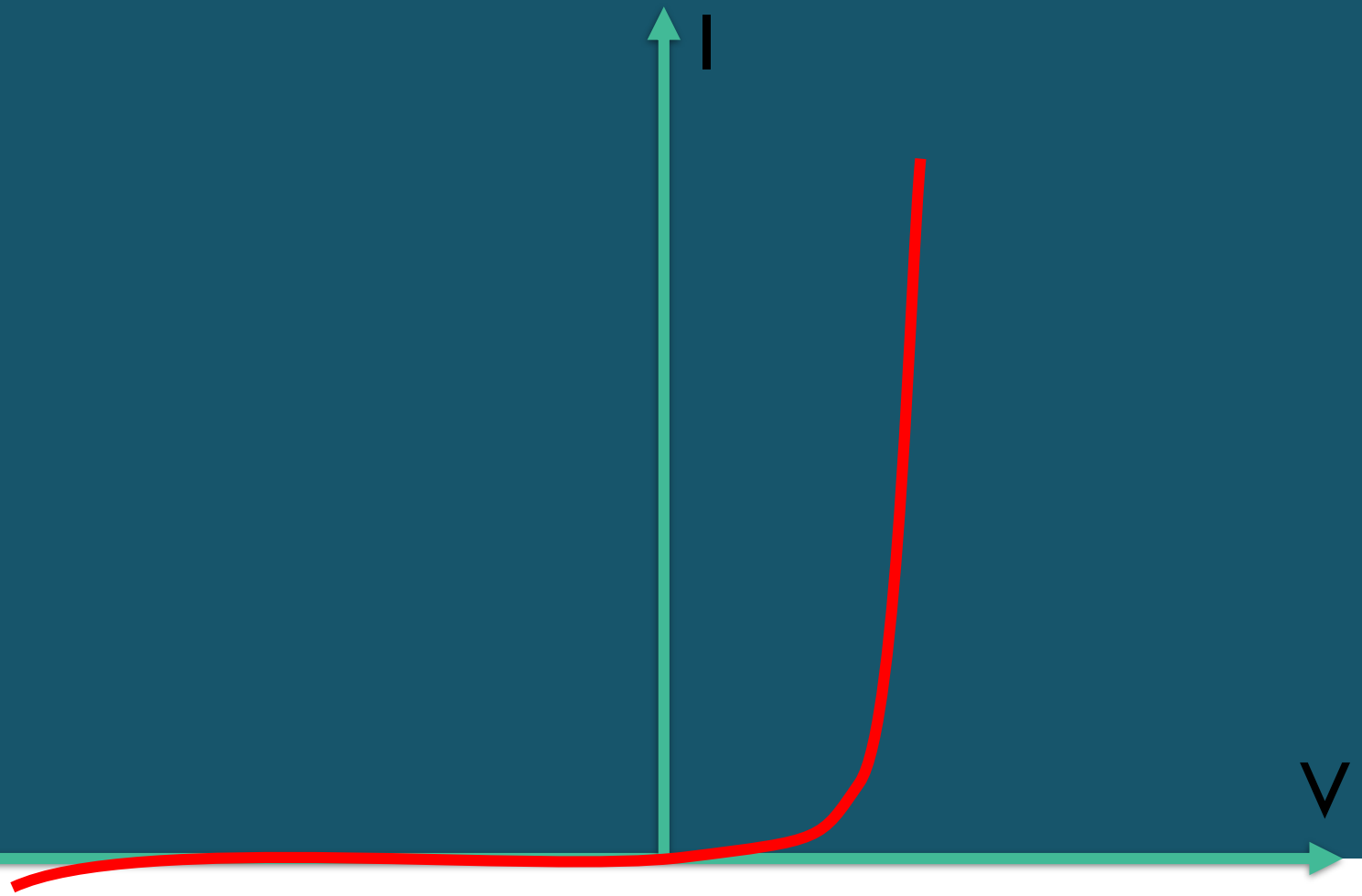
- `fonte.query("MEASure:CURREnt?")`

FONTE DE ALIMENTAÇÃO

Output Setting and Measurement Commands

(see page 78 for more information)

```
APPLY {<voltage>|DEF|MIN|MAX}[, {<current>|DEF|MIN|MAX}]
APPLY?
[SOURCE:]
  CURRENT[:LEVEL][:IMMEDIATE][:AMPLITUDE] {<current>|MIN|MAX|UP|DOWN}
  CURRENT[:LEVEL][:IMMEDIATE][:AMPLITUDE]? [MIN|MAX]
  CURRENT[:LEVEL][:IMMEDIATE]:STEP[:INCREMENT]
    {<numeric value> |DEFAULT}
  CURRENT[:LEVEL][:IMMEDIATE]:STEP[:INCREMENT]? [DEFAULT]
  CURRENT[:LEVEL]:TRIGGERED[:AMPLITUDE] {<current>|MIN|MAX}
  CURRENT[:LEVEL]:TRIGGERED[:AMPLITUDE]? [MIN|MAX]
  VOLTAGE[:LEVEL][:IMMEDIATE][:AMPLITUDE]
    {<voltage>|MIN|MAX|UP|DOWN}
  VOLTAGE[:LEVEL][:IMMEDIATE][:AMPLITUDE]? [MIN|MAX]
  VOLTAGE[:LEVEL][:IMMEDIATE]:STEP[:INCREMENT]
    {<numeric value>|DEFAULT}
  VOLTAGE[:LEVEL][:IMMEDIATE]:STEP[:INCREMENT]? [DEFAULT]
  VOLTAGE[:LEVEL]:TRIGGERED[:AMPLITUDE] {<voltage>|MIN|MAX}
  VOLTAGE[:LEVEL]:TRIGGERED[:AMPLITUDE]? [MIN|MAX]
  VOLTAGE:PROTECTION[:LEVEL] {<voltage>|MIN|MAX}
  VOLTAGE:PROTECTION[:LEVEL]? [MIN|MAX]
  VOLTAGE:PROTECTION:STATE {0|1|OFF|ON}
  VOLTAGE:PROTECTION:STATE?
  VOLTAGE:PROTECTION:TRIPPED?
  VOLTAGE:PROTECTION:CLEAR
  VOLTAGE:RANGE {P8V*|P20V*|P35V**|P60V**|LOW|HIGH}
  VOLTAGE:RANGE?
MEASURE
[:SCALAR]
  :CURRENT[:DC]?
  [:VOLTAGE][:DC]?
```



CARACTERIZAÇÃO DE UM LED

Controle da fonte
Caracterização LED

ROTINA DE CONTROLE

- Fornecer várias tensões e obter a corrente correspondente
 - liberar o output
 - intervalo definido da tensão aplicada
 - ler a corrente para cada
 - gráfico
 - [LEDchar.py](#)

MUITO OBRIGADO

BORGHI@IF.UFRJ.BR

Professor IF – UFRJ

GPENELLO@IF.UFRJ.BR

Professor IF – UFRJ

JAIMEFDEOLIVEIRA@GMAIL.COM

Doutorando – CBPF

REFERÊNCIAS

Manual do equipamento

<https://github.com/MarkDing/GPIB-pyvisa>

<https://github.com/hgrecco/pyvisa>

https://pt.wikipedia.org/wiki/Diodo_emissor_de_luz

Acessados 20/03/19 às 14h