plc --- PlasmaLabControl

Author: Daniel Mohr
Date: 2012-10-14

plc is a software package to control the experiment setup of Plasmalb.

What's there and what we need:

What's there:

- many controlling and measuring boxes (digital controller, camera, ...)

What we need:

- software component for every box (IO communication, logging)
- one software component for controlling (human interface)

What was there:

- stubs of software for some boxes

Software Concept:

- many processes with threads communicating over TCP (socket communication)
  - flexible
  - every process can run on own cpu core (automatically or manually)
  - threading prevents unnecessary blockings of the whole system while IO communication
  - every process can run on different boxes/computers
    - small processes can go to own boxes in future
    - heavy processes can get their own computers in future
    - controlling interface (human interface) can run on separated location (e. g. near the scientist and not near the experiment setup)
    - living in the cloud, e. g.:
      - experiment setup on the A300-ZERO-G (or ISS)
      - logging somewhere in Bordeaux (or Garching)
      - controlling from the Dune of Pyla (or Ringberg)
  - iff running on one Linux system (computer):
    - loopback interface is fast
    - kernel can do the load sharing

Implementation and Development:

- Python (at the moment also C; as far as possible this will be changed in future)
- popular Version: 2.7.x
- popular and suitable operating system: Ubuntu 12.04

Human Interface:

- GUI
  - Tkinter (Tcl/Tk)
  - mostly platform indepent (at the moment some controlling parts are integrated; and therefore only complete funtionable on POSIX systems; this will be changend in near future)
- config files
  - style of RFC 822
    - mostly human readable
    - platform indepent
- Batch Controlling:
  - list of setpoints in a config file, which can be loaded and set in the GUI

Server Components:

For every controlling and measuring box we write some components: (at the moment heavy development)

- server program
  - IO to the device
  - logging
  - socket communication for controlling
- client program (for debugging)
  - human interface (at the moment only GUI)
  - socket communication with the server program
- integration of a useful API in the human interface GUI

Other Parts:

plc contains some extra tools for viewing/analysing the measurements. For example:

- rawmovieviewer.py (player for the img-files (PAM format))
  - quick viewing of the measured/recorded movies
- rawmovies2recordings.py (converts img-files (PAM format) to zip-archives with png graphics)
  - img-files in the PAM format are large and structured to 1 GB files
  - resulting zip-archives are smaller and structured in recordings (parabolas)
- plc_viewer.py (viewer of the movies and other measurements)
  - assignment of the frames to other measurements

Python

Author: Daniel Mohr

Date: 2012-10-14

Famous Quotes related to Python:

from: http://www.python.org/about/quotes/

Quates related to Python:

- "Python is fast enough for our site and allows us to produce maintainable features in record times, with a minimum of developers," said Cuong Do, Software Architect, YouTube.com.
- "Python plays a key role in our production pipeline. Without it a project the size of Star Wars: Episode II would have been very difficult to pull off. From crowd rendering to batch processing to compositing, Python binds all things together," said Tommy Burnette, Senior Technical Director, Industrial Light & Magic.

Advantages of Python:

- high-level programming language
- multi-paradigm programming language
- widely used language (in particular for scientific programming)
- interpreter available -> rapid development (high effectivity)
- fast modules (written in C or C++) available
- fast modules (written in C or C++) programmable -> all C libraries are available (also directly via ctypes)
- simple semantics -> code readability, errorproof programming
- large and comprehensive standard library
- good documentation
- open source (e. g. you can read the code and look for bugs)
- available for many operating systems: (but with the pitfalls/deficiencies of the operating systems)
    - Linux (Ubuntu, Gentoo, MLDE, Fedora, ...)
    - Uniix (AIX, FreeBSD, NetBSD, OpenBSD, ...)
    - Mac OS X
    - Windows
    - MS-DOS, OS/2, Palm OS, PlayStation, Psion, Symbian, ...

Disadvantage of Python:

- Compared to C pure Python code is slow, but:
    - time-critical functions can be translated to C by Cython
    - time-critical functions can be (re)written in C or C++
    - until now, we have no time-critical functions in pure Python
    - time-critical functions we need, are already implemented as modules (mostly in the standard library)

Choosed Version:

- Python V. 2.7.x is still widely-used. (e. g. MAC)
- Developed in Python V. 2.7.x with regard to V. 3.x
- 2to3 converts the code to Python V. 3.x

A few famous example of Python Software:

- Bazaar (distributed revision control system)
- BitTorrent Client (reference implementation)
- Calibre (e-book management tool)
- Dropbox (web-based file hosting service)
- Portage (advanced package management system)
- Ubuntu Software Center (graphical package manager)
- Wammu (mobile phone management utility)
- Plone (content management system)
- PyCUDA (Python bindings for Nvidia CUDA)
- SciPy (library of scientific and numerical routines; based on BLAS/ATLAS)
- Sphinx (documentation generator)
- many tasks of Google (e. g. Google Groups, Gmail, Google Maps, part of the search engine)
- most parts of Sugar (<- One Laptop per Child)

Famous Software using Python as scripting language for the user:

- Blender (3D animation program)
- GIMP (graphic program)
- Inkscape (vector graphic software)
- Scribus (desktop publishing application)
- Vim (text editor)

Famous institutions using Python:

- Google
- Yahoo!
- CERN
- NASA