PM Lab Tools

Johannes Weiß, Jonathan Dimond

23. Januar 2012

Outline

- Basics
- Programmatic C API (a.k.a. `libpmlab`)
- Stream Based Plain Text Data (a.k.a. `pmclient`)
- Real–time Data Viewer (a.k.a. `realtime.py`)

git Repository

```
$ git clone http://github.com/weissi/pm-lab-tools.git
$ cd pm-lab-tools/client
$ ls
libpmlab.h # Programmatic C-API
libpmlab.c # Programmatic C-API

pmlabclient.c # Stream Based Plain Text Data

realtime.py # Real--time Data Viewer

sample.R # R sample
sample.gp # gnuplot sample
```

Building

```
cd pm-lab-tools
git pull origin master # Update from repository
./build.sh client

build/pmlabclient i30pm1 12345 pmX # Execute
```

Programmatic C API

```
void *pm_connect(char *server,
                 char *port,
                 unsigned int *channels,
                 unsigned int num_channels);


int pm_read(void *handle,
            size_t buffer_sizes,
            double *analog_data,
            digival_t *unused,
            unsigned int *samples_read,
            uint64_t *timestamp_nanos);

void pm_close(void *handle);
```

Stream Based Plain Text Data

```
$ build/pmlabclient localhost 12345 | head
40221.000000 0.023850 0.023165 0.023238 0.000482 0.003031 0
40221.000056 0.024285 0.023034 0.022915 0.002563 0.002781 0
40221.000111 0.024726 0.023468 0.023225 0.002860 0.002827 -
40221.000167 0.023791 0.023053 0.024166 0.003097 0.002853 -
40221.000222 0.023732 0.023238 0.023132 0.002056 0.003242 0
40221.000278 0.024298 0.022895 0.023145 0.002570 0.001332 0
40221.000333 0.024496 0.023086 0.023099 0.002590 0.002972 0
40221.000389 0.024094 0.023475 0.023883 0.002418 0.002385 -
40221.000444 0.023811 0.023238 0.022941 0.003452 0.003413 -
40221.000500 0.024199 0.023139 0.023185 0.002003 0.003044 0
```

Real–time Viewer

```
build/pmlabclient localhost 12345 |
 client/realtime.py 40 1
```

or

```
./plot-watts pmX
```