creating a gauge part 3

We already have a program that draws the graphics. Right now it just displays random data, which is not really usefull.This part will fill the dummy functions with life.

CPU usage information

The source for the cpu usage is the file /proc/stat

It shows the ticks the cpu has spend in idle, nice, system or user mode.

We save the last values we got from there and calculate the difference. This gives our cpu usage value.

2 example lines from /proc/stat

```
cpu  92208 2282668 19432 79824 739 1441 3304
cpu  92310 2284457 19454 79910 739 1443 3306
```

Those 2 values are taken with a 10 sec delay.

So what are those values?

The /proc/stat gives the following information

user nice system idle ...

All values are 1/100 of a second

In the example above our cpu has spent 102/1789/22/86 ticks in user/nice/system/idle mode.

Something's wrong, it doesn't sum up to 1000 (1 second). Right, the mashine I used was a multi cpu mashine :) It had 2 cpus, so it should go to 2000. But this doesnt matter, because when you do the following calculation

```
(user+system)/(user+system+nice+idle)
```

you'll get the cpu usage in values from 0 to 1. You can add nice to your left side if you want. Because I got a few things running in nice mode that consume a lot of cpu, I don't add it to my calculations.

Thats basically all you need to get the cpu information. Parsing the cpu line from /proc/stat is left to the reader.

bandwidth usage

Getting the bandwidth is basically the same work. We need to parse /proc/net/dev an example output fo that file: (The line was broken into 2 parts to fit this page)

```
Inter-|    Receive                                                ...
 face | bytes    packets errs drop fifo frame compressed multicast
   lo  63686427   34789   0    0    0     0          0         0
 eth0  502589778  2816788 0    0    0     0          0         0
 eth1  3587148619 3339472 0    0    0     0          0         0
 ppp0  33739494   58634   0    0    0     0          0         0
...   |   Transmit
      |bytes     packets errs drop fifo colls carrier compressed
   lo  63686427   34789   0    0    0     0       0          0
 eth0  3886483446 3781565 0    0   10 18450       0          0
 eth1  382757499  2360540 1    0    1     0       0          0
 ppp0  16003460   60075   0    0    0     0       0          0
```

This output is quite self-explaining.

All you have to do is to parse the device you want and compare that value to the values X seconds ago. Depending on your network hardware you have to divide that value with your maximum upload/download speed of that device and you have your bandwidth usage.