Subject: Re: usecs
Posted by Zbych on Thu, 06 Dec 2018 19:19:49 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 06 December 2018 19:46A the moment, it seemed like a good solution to use what C++ lib provides.

I strongly disagree. I've made a test and std::chrono::high_resolution_clock::is_steady returned false in Linux/Gcc.
That means that new version of msecs is not reliable. Maybe for usecs it doesn't matter, but msecs is used to measure timeouts in many places in Upp.

My proposition is to use steady_clock for msecs instead:

```
int msecs(int prev)
{
 auto p2 = std::chrono::steady_clock::now();
 return (int)std::chrono::duration_cast<std::chrono::milliseconds>(p2.time_since_epoch()).count() - prev;
}
```

I compared returned value with old implementation (clock_gettime(CLOCK_MONOTONIC...) and they are exactly the same.
Resolution of steady_clock in Linux is about 1ms, so it doesn't make sense to use it in usecs.