

Re: Horrid Serial Dacs

Source: <http://sci.tech-archive.net/Archive/sci.electronics.design/2004-09/1420.html>

From: Andreas Hadler (*Andreas.Hadler_at_t-online.de*)

Date: 09/07/04

Date: Wed, 08 Sep 2004 00:11:59 +0200

Jim Thompson <thegreatone@example.com> wrote:

>On Tue, 07 Sep 2004 22:50:00 +0200, Andreas Hadler

><Andreas.Hadler@t-online.de> wrote:

>

>>Always code as if the guy that ends up maintaining and/or testing you

>>code is a violent psychopath who knows exactly where you live.

>>(unknown)

>ROTFLMAO!

>

>The second one reminds me of the oldest son's admonition:

>Document/Indent/Document/Indent/Document/Indent

>

>He'd look at stuff I'd written (I quit at Pascal) and chew my ass

>every which way from Sunday ;-)

On my software face, today, I try to code for readability for "the next guy" (which, most of the time, is me :-). To do this, satisfying optimal execution speed / minimal memory requirements, is IMHO the real "Art of computer programming". It was a real long way to see software as kind of literature, and to be proud of the technical excellence, which can be achieved along the way and is invisible most of the time.

You can't impress someone with such code. It's understandable, and it's working – not really an impressive achievement. You'd get a better name and a better reputation when fighting (and winning). The best compliments you'll get with such code are comments of coworkers like "hey, in this 25k lines, you made an obvious mistake in module xxx, which only took me about an hour to fix after the clients call". Really another kind of comment than begging a software god for assistance.

For each single case, it's economically not really the best way. You can get more money and earn more reputation, when the customer gets proofs for difficulties and pays for their solution. In the long run,

sci.electronics.design: Re: Horrid Serial Dacs

I lost some customers, because they didn't appreciate me as a fighting winner, but I won much more long-time clients appreciating an unobscured and working solution. IMHO, that's what a craftsman (and that's what an engineer is, just on maybe a different level) is for – delivering something working and satisfying without any hassle. Without vanity. Without bothering the client.

Andreas

--

Every program has at least one bug and can be reduced by at least one line. By induction, then, every program can be reduced to a single instruction, and that will be wrong.