

Re: How to develop a random number generation device

Source: <http://sci.tech-archive.net/Archive/sci.electronics.design/2007-09/msg01619.html>

- *From:* John Larkin <jjlarkin@xx>
 - *Date:* Sun, 09 Sep 2007 19:35:43 -0700
-

On Sun, 09 Sep 2007 16:21:24 -0700, MooseFET <kensmith@xxxxxxxx> wrote:

On Sep 9, 1:48 pm, John Larkin
<jjlar...@xx> wrote:
[...]

Watch dogs don't always recover the system from a glitch. If you are storing data in battery back RAM or flash, you need to be sure that wrong values don't cause things to hang in some non-recoverable way.

One of my programming rules is that data should never be able to crash code.

The assignment statement is more dangerous than a GOTO.

Assignments don't crash in assembly, since assembly is untyped. The only math error possible is a divide-by-zero trap, or a stupid pointer. What's important is that program flow doesn't bomb just because some cal table is trashed.

There's nothing wrong with GOTO; hell, Dijkstra didn't even have regular access to a computer, and didn't actually program much. I think in state machines, so GOTO is perfectly logical. In assembly, a conditional branch, or a computed/table driven jump, are the primary control structures.

Nested curly brackets are just as dangerous, or moreso.

John

Re: How to develop a random number generation device