

Re: Disobeying jet engines – why?

Re: Disobeying jet engines – why?

Source: <http://sci.tech-archive.net/Archive/sci.electronics.design/2008-01/msg03925.html>

- *From:* Glen Walpert <gwalpert@xxxxxxxxxx>
 - *Date:* Fri, 25 Jan 2008 13:53:26 GMT
-

On Thu, 24 Jan 2008 11:44:52 -0800, John Larkin
<jjlarkin@xx> wrote:

On Thu, 24 Jan 2008 13:18:35 -0600, Damon Hill
<damon1SIX1@xxxxxxxxxxxxxxxxxx> wrote:

Didi <diditgi@xxxxxxxxxx> wrote in
[news:f121757f-1671-4ebe-892d-625ea1c236b6
@h11g2000prf.googlegroups.com](mailto:news:f121757f-1671-4ebe-892d-625ea1c236b6@h11g2000prf.googlegroups.com):

<http://news.bbc.co.uk/1/hi/england/london/7206596.stm>

Anyway, to me this sounds like some popular office OS has made its way into the cockpit and was out for lunch with the HDD while the pilot was trying to talk it into delivering his commands to the engine controllers...

Windoze is NOT qualified for critical applications like this. It's something pretty specialized and focused for the application. I suspect the investigation will either find a hardware fault or operator error (misconfigured autopilot). Airbus has had something similar happen, resulting in a crash and loss of life.

<http://www.youtube.com/watch?v=-kHa3WNerjU>

--Damon Real life doesn't have a 'reset' button

I know some of the guys who do the engine control computer firmware for the Pratt&Whitney engines. They use our gear to simulate engine

Re: Disobeying jet engines – why?

sensor signals to the control computer, and run weeks/months of scripts to verify the firmware in all sorts of situations.

Their ECC's use no OS at all, just basic bare-metal state machines.

You mean that they roll their own RTOS rather than buying one known to work. You can make a state machine controller with FPGAs and no OS, but a uP needs some sort of RTOS in order to function even if they don't call it that and it is integral with the rest of the code.

I recall a crash investigation a few years back (written up in Embedded Systems Design IIRC) where the crash was attributed to ECC software errors. Code analysis by a third party revealed something like 4000 errors (deviation from accepted high reliability software practice) in the code, many of them serious, resulting from a complete lack of adequate software and testing standards.

My guess is that the latest crash from loss of engine control will also be found to be a result of bad software which probably did not handle some unusual exception (such as a sensor failure) properly.

.