

# Re: Atmel AVR development tools

---

*Source:* <http://sci.tech-archive.net/Archive/sci.electronics.design/2008-02/msg00117.html>

---

- *From:* dbvanhorn <[microbrix@xxxxxxxxxx](mailto:microbrix@xxxxxxxxxx)>
  - *Date:* Fri, 1 Feb 2008 08:30:42 -0800 (PST)
- 

The AVR bootloader has to contain the subroutines for erasing and programming. Those subroutines can be activated accidentally by an electric glitch, software bug, cosmic ray, unholy spirit or something like that. Although this event is not very likely, I have observed the failure rate due to that at the order of one per 5k units per year in the harsh EMI environment.

Theoretically, I agree, but that would be true of anything really, and any routine in the system.

My biggest experience with bootloaders was millions of credit card terminals.

The feature was WELL worth it, any problems that the bootloaders gave us were so far down in the noise as to be unobservable. Our harshest environments were Ski resorts and Las Vegas/Reno casinos. Cold dry air and industrial carpeting make for some impressive ESD problems. These systems kept their operating program in SRAM, and the loader/interpreter in ROM, so it was a bit safer that way.

I like how the AVR implements the loader, and both the loader, and the lockbits can be used to prevent writes to the loader itself. In my case, the system may end up with a bunch of small peripheral processors who all need to be updated from the core processor, so I'm pretty much forced to use it, or forego the ability to do an upgrade without disassembly and special tools.

In my loader, when the loader boots, it checks the lock bits, and if they aren't set properly, it sets them.

Then while loading, it examines the page addresses, and prevents any attempt (even though it wouldn't work) to write to the loader space. Wild app code could try to overwrite the loader, but I don't use the SPM instruction anywhere in my application code, so there isn't a routine that could accidentally run, and even if it did, it would be prevented by the lock bits.

The routines up in the loader space exist, and could trash the app,

Re: Atmel AVR development tools

but I'd have to have multiple and very exact failures to cause them to over-write the loader itself. As long as that's true, I'm good. Systemwise, I have bigger threats.