

Re: Learning Code...Fast

Source: <http://coding.derkeiler.com/Archive/General/comp.programming/2004-06/0569.html>

From: David (*FlyLikeAnEagle_at_United.Com*)

Date: 06/10/04

Date: Thu, 10 Jun 2004 02:58:08 GMT

Hi Daniel,

On Wed, 9 Jun 2004 15:15:25 UTC, mcdan@csi.com (Daniel McCarty) wrote:

> "David" <FlyLikeAnEagle@United.Com> wrote in message
news:<rOdGr40LMPU3-pn2-Z8LjOceuG5E4@localhost>...
> >
> > [...] You may be able to do a top-down approach if the project is
> > fairly well organized. [...]
>
> Hi David,
> Thanks very much for the advice. I should mention that it's an
> embedded systems project, so there are a lot of magic numbers, memory
> addresses and hardware intricacies sprinkled throughout the code.
> That may be the toughest part. As for the project being in Zinc, yes,
> it's a pretty readable format. (If the dice of history rolled a
> little differently we might have all been using Zinc instead of Java,
> but that's a topic for a different day.)

I spend about half my time on embedded projects (Forth and C).
If you can get the hardware diagrams or an explanation of the hardware
addressing that will help quite a bit. It doesn't make much sense that
Q4 is set here and reset there. When we find out that Q4 enables
an audio driver or turns on the DTMF decoder it makes the surrounding
code and magic numbers a bit less cryptic.

Protocols are important too. I have many serial protocols that
are pretty much state machines. Without some document to show the
whole conversation and order used, making changes is a risky task.

Embedded programs also have certain traits that application
programmers don't have. They rely more on state transitions and
very small code fragments. The order of memory accesses can be
very important. Initialization and other tasks may be distributed
in an unfamiliar pattern. We also use watch dog timers and other
hardware transitions to observe and control the system behavior.

comp.programming: Re: Learning Code...Fast

comp.sys.embedded may also be a good place pose some of your questions.

> *Part of the first thing that I do may have to be to write the
> project documentation, if only for my own sanity.*

That is generally my first step when approaching a new project. I can keep quite a bit in my head, but there are generally a lot of paper napkin-like drawings that help piece a project together. They are really important in embedded interfaces where the details are critical.

Its too bad that most of this documentation never quite makes it into the code base. I try to document what I can, usually in text description files, that sit alongside the source code.

Even the compiler and build procedures can be very important to document.

> *Thanks again,
> Dan*

David