

Re: which object orient language is most suitable for embedded programming?

Re: which object orient language is most suitable for embedded programming?

Source: <http://coding.derkeiler.com/Archive/General/comp.arch.embedded/2006-11/msg01286.html>

- *From:* Vladimir Vassilevsky <antispam_bogus@xxxxxxxxxxxxx>
 - *Date:* Fri, 24 Nov 2006 20:19:21 GMT
-

Peter Dickerson wrote:

The size of C++ startup code is comparable to the size of a small application.

No true. I have implemented code for a small scientific instrument that uses C++. I wrote the startup code myself

You had to write the startup code yourself...

They pay money for the problems solved, not for the rewriting of startups and standard libraries.

based on the C startup because I needed

to init hardware in C earlier than zeroing BSS. The additional code needed to support C++ was a scan of a table of static constructors before main and destructors after.

Besides that, the C++ startup has a lot of extra functionality that can be used on the special occasions. Such as: `if (OS == blabla)`, `if (library == foobar)`, `if (CPU_version == 12345)` and so on, so forth.

Of course, you can eliminate most of that. How much time it will take? Are you sure there will not be a problem in the future because you removed something essential?

In fact the tables are empty in my case, so that overhead

is wasted. I used GCC for ARM. Note, as discussed earlier this is without RTTI and exceptions – if you're using those then it really is an apples and oranges comparison.

If I don't use exceptions and RTTI right now right here, it does not mean that I will not need all of that later.

Re: which object orient language is most suitable for embedded programming?

Re: which object orient language is most suitable for embedded programming?

There are also compiler related inefficiencies such as the tables of pointers to functions have to be placed in RAM rather than in ROM, etc.

Why? The tables of pointers to functions, commonly called vtables, can, and are, stored in ROM.

You should adress this question to IAR systems.

But, I don't mind that. My goal is developing an application for good and on time. They don't pay for proving a point or saving few bytes.

To me, the whole point of using C++ is avoiding the unnecessary technical knowledge about minor details.

Perhaps. It is never a good idea to think that you an get away with an inferior programmer by providing a 'better' programming language.

Rewriting the startups is definitely not the best use for the good programmer.

VLV

.