Misra standard for embedded software

Asked 16 years, 3 months ago Modified 12 years, 3 months ago

Viewed 11k times

13

I have a requirement to make a large amount of code MISRA compliant.
First question: Can somebody to give an **estimation** for passing well written code for embedded system based on experience. I understand that "well written" is poorly defined and vague so i ask for raw estimation.
Second question: Any recommendation for tool that can be customizable (i.e allowing suppress specific warnings) and used in automatic build environment (i.e command line interface)
Any other useful suggestions that can help with this task.
Thanks Ilya.

c code-analysis embedded misra

Share

Improve this question

Follow

edited Aug 29, 2010 at 8:31

Pascal Thivent

570k ● 140 ● 1.1k ● 1.1k

asked Sep 15, 2008 at 13:39

Ilya

3,138 ● 4 ● 24 ● 30

This question depends a lot on which version of MISRA you need. The latest 2012 version is still lacking tool support from many companies. – Lundin Dec 20, 2013 at 10:09

6 Answers

Sorted by: Highest score (default)

12

I also highly recommend PC-Lint. If you happen to be compiling your code with Visual Studio I recommend a plug-in 'Visual Lint' from Riverblade. If you cannot compile the code in Visual Studio, you can still run PC-Lint from the command line to good effect.

Some embedded system compilers provide MISRA compliance testing as compiler warnings. I use the IAR compiler for Arm7/Arm9 development. It provides an easy to configure MISRA compliance checklist right in the compiler setup.

It is difficult to come up with a rule of thumb for estimating the time it would take you to make some well written code MISRA compliant. A lot depends on the existing coding habits of the programmers and how closely they follow the MISRA rules in the first place.

Rough estimates:
2 - 3 days to become adept at PC-Lint usage.
Initial pass at making existing code MISRA compliant: 10 to 25 percent of the time spent writing the code in the first place.
Keeping code MISRA compliant: 5 to 10 percent added to

code development. Half of this cost is changing the habits of your coders to follow the 'MISRA way' of doing things. The other half is the extra cost of code testing and inspection to ensure MISRA compliance.

Share Improve this answer Follow

edited Jul 26, 2012 at 15:05

Martin Thompson
16.8k ● 1 ● 39 ● 58

answered Sep 15, 2008 at 15:11

Darryl Buchanan
136 ● 2

Please note that none of these tools support MISRA-C:2012. I recently asked IAR if they supported it and they didn't seem to even be aware there was a new version out... – Lundin Dec 20, 2013 at 10:10

That being said, I would recommend to get a cheap tool such as Lint. It will give hundreds of false warnings, but so will the supposedly "state of the art" static analysers that cost a fortune. Expensive tools like LDRA och Klocwork are equally buggy as Lint when it comes to MISRA checking. – Lundin Dec 20, 2013 at 10:18

8

Making code Misra compliant it not too much of a chore - if you follow fairly good programming practices. You might find some of the pointer rules slightly tricky, if the code you're trying to make comply has some weird and wonderful pointer arithmetic.

I'd second Greg's recommendation for PC Lint, but the open-source Splint is also worth looking at, although between them (and the compiler's warning system), I estimate you'll still only be able to cover 80% of the Misra rules - the rest will probably need to be code reviewed by hand.

Share Improve this answer Follow

answered Sep 15, 2008 at 13:55

Jon Mills
1,875 ● 4 ● 20 ● 28

4

I use PC Lint for static analysis of C and C++ code. It can be configured to show what MISRA rules have been violated, and it has a command line interface.

Share Improve this answer Follow

answered Sep 15, 2008 at 13:47

Greg

3

I have used a commercial tool called [QAC](). The tool is able to enforce [MISRA]()

It has a command-line interface, so you can set it up to run from a automated build environment. The rules to be applied are configurable, but expect to have someone spending some time setting it u. The MISRA enforcement is pretty straightforward and worked well enough. I was told (and this is just 3rd hand) that this is one of the tools some agencies (such as the FDA) use to evaluate code. Like most static analysis tools there is noise (false positives) to deal with. The last time I used it, it didn't have a good means to mark/stop a false positive from occurring again (without changing the code it was complaining about).

I suspect a junior engineer will take up to a week (4-5 days) to get it setup (assuming they are determined to get it working as you want).

On a side note, other commercial static analysis tools likely have MISRA enforcement as well. Reportedly (per their sales rep), Klocwork does.

Share Improve this answer Follow

answered Sep 15, 2008 at 23:19

Zing-
2,167 ● 2 ● 13 ● 12

---

2 Junior engineers should not be assigned to convert code for MISRA compliance. In my experience, they will end up slavishly following the tool's warnings, most of which are false ones no matter the tool. They may be causing more problems than they fix. – Lundin Dec 20, 2013 at 10:15

3

We had a similar problem of retrofitting Misra rules. We had some code quality issues on a large project and decided to use MISRA to improve the code quality.

We use the Green Hills compiler that has support for MISRA C rules. There is also stand alone checkers available. Depending on what you want to do it can be a bit over kill switching on all the rules. We switched one the rule on at a time to give people time to fix a limited number of similar problems else you get totally overwhelmed by the amount of errors.

Since our warnings was generated by the compiler and not by a standalone tool you see the errors as you develop and not only when you run the checker. As we continued developing we got our code compliant and not in one big bang. This also prevent old habits spoiling the new code causing you to having to rework the code again later.

Some times it is difficult to get old code compliant since nobody knows exactly how the code works. I hope you have unit tests.

Share Improve this answer Follow

edited Jul 26, 2012 at 10:08

Jens
72.4k ● 16 ● 130 ● 183

answered Nov 10, 2008 at 11:19

Gerhard
7,031 ● 8 ● 55 ● 83

1

I appreciate that this is an old question, but for the benefit of any other Archaeologists (or searchers), it is important to remember that MISRA provides **guidelines** that should not always be blindly followed.

I commend writing new code with MISRA in mind; therefore it will be a lot easier to stay compliant.

However, this is not always possible - and in particular, when trying to reverse engineer code to meet the guidelines. In this case I suggest that you focus on the Required rules, and treat the Advisories as a bonus... cost v benefit applies here too!

Also, bear in mind that there is a deviation process - it is better to keep clean and maintainable code with a deviation, than to contrive some compliant but illegible spaghetti.

Share Improve this answer Follow

answered Aug 28, 2012 at 13:19

Andrew
2,302 ● 1 ● 28 ● 40