

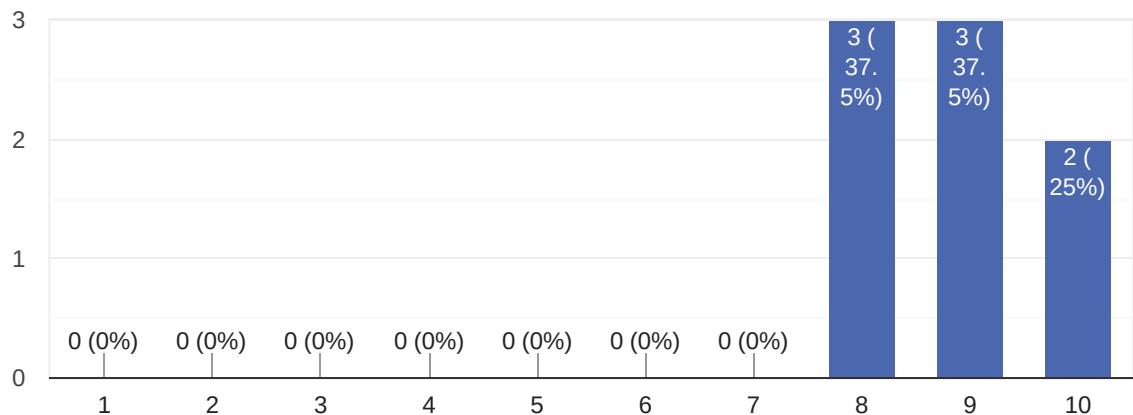
Bootlin training course evaluation

8 responses

[Publish analytics](#)

Overall rating of the course

8 responses



Comments and suggestions

3 responses

Good slides and material, a bit more of improvement on the practical examples.

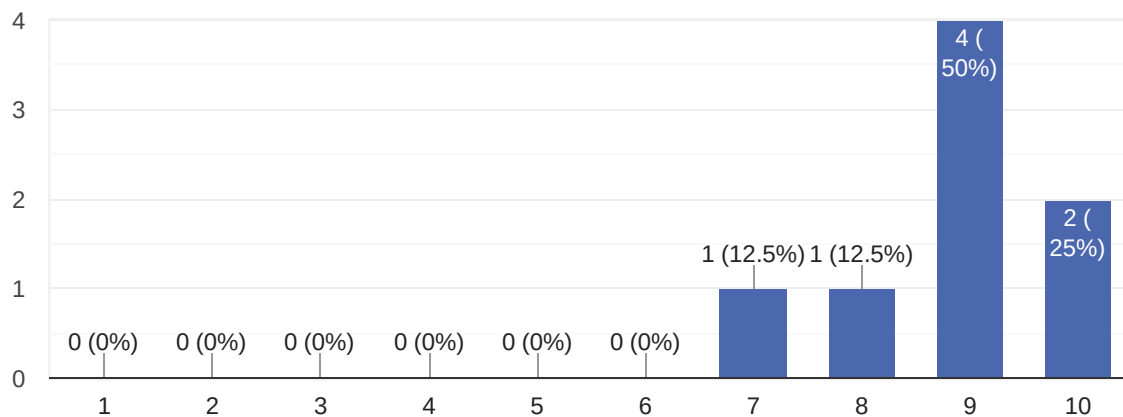
Very fluently explained. First part a bit of recap. Second part very interesting. Third part a bit too practical. Add some modern C++ (`std::mutex`) topics?

The training was very useful and interesting to us. It gave us insights which we can experiment with and apply directly in the products we are already shipping. I think the 3 blocks of 4 hours are a minimum for covering this topic in depth.



How useful were the lectures?

8 responses



Comments and suggestions

2 responses

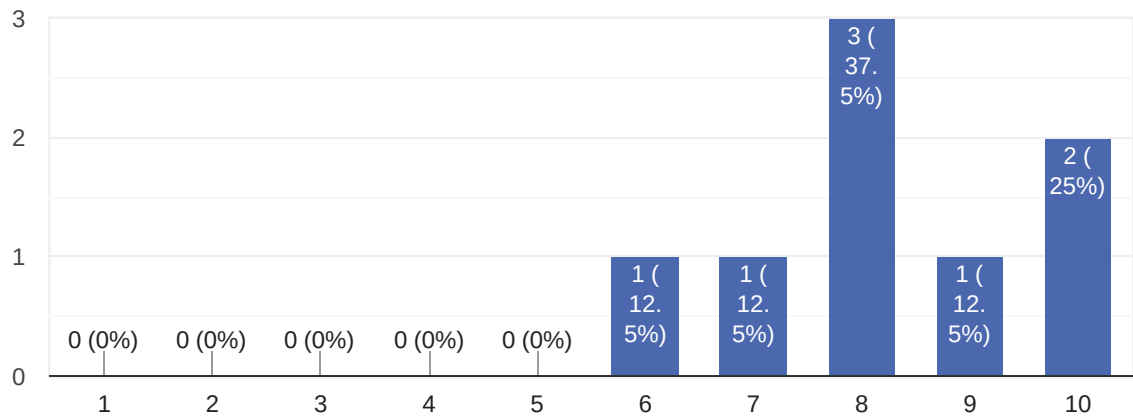
Not immediately useful for the work I'm doing, but very interesting to know the fundamentals where applications are built on.

Very well presented. Also, it was very clear the presenter is very experienced with the topic, and knows what he's talking about.



How useful were the practical demos?

8 responses



Comments and suggestions

5 responses

This is the part that needs improvement. I suggest to have some preset use cases documented in the labs so one can follow them offline, specially syntax of commands and combination of tools to see pefromance metrics is important to learn quickly the concept and analyze the result. The trainer did very good on showing some down to earth practical examples, just suggest to have few of them documented would be great.

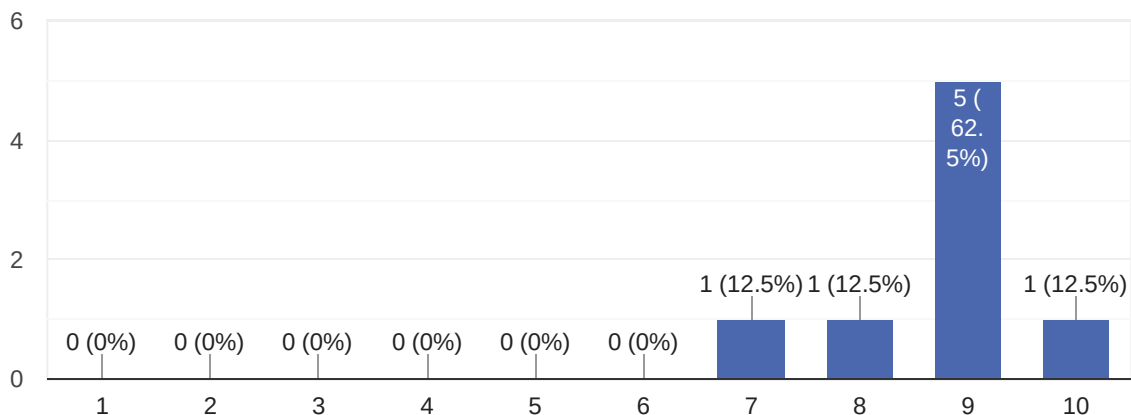
If possible I would add a demonstration of a Thread RT program using a peripheral that is measurable with a scope. Like a simple gpio toggling(if this doesn't involve too much overhead to go through vfs etc.) and show the scope's signal to underline the drift under stress, maybe with persistency sampling on scope. It can even be a 10Khz square wave, so the scope can be a crappy usb scope. This way you can have the scope's display onscreen and share it like you already share the desktop.

You can't help it when participants have never done the demos themselves, which can make it sometimes harder to follow. Maybe ask for some simple tasks to do up front as preparation?

The demo's are ok, but it would be more interesting to try and solve some problem yourself in the labs like in the Yocto training. For example startina out with a sample

How would you rate the overall organization of the course?

8 responses



Comments and suggestions

5 responses

The first time login process is a bit confusing. I remember I have to register an account via a purely French language website, which is challenging for me. Could be nice if it is simplified.

I did not give a 10 here only because the platform for demos was changed last minute (BBB to ST32), there were good reasons as explained but I would have prepared much better. One suggestion maybe is to pick a more popular platform that supports multicore (RBP ?)

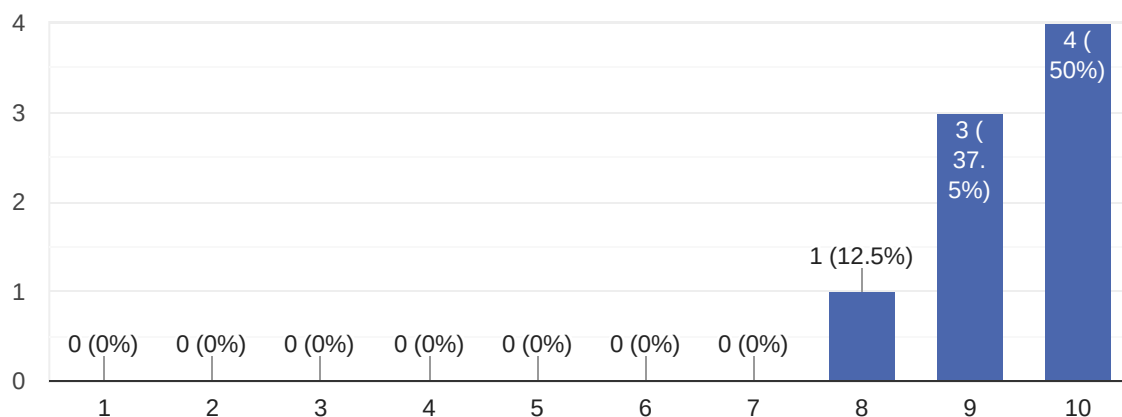
The only thing I would write "bigger" in Eventbrite is that you will receive the link from your instructor. I knew this from the course of Graphics, but at the time I had to ping you and I couldn't find on Eventbrite the link.

Internal communication in our company was lacking. At first I dismissed the December mail as phishing.

Very well structured, starting from theoretical background, and gradually moving to practical application.

How would you rate the trainer?

8 responses



Comments and suggestions

4 responses

The trainer is very friendly, patient, and willing to help.

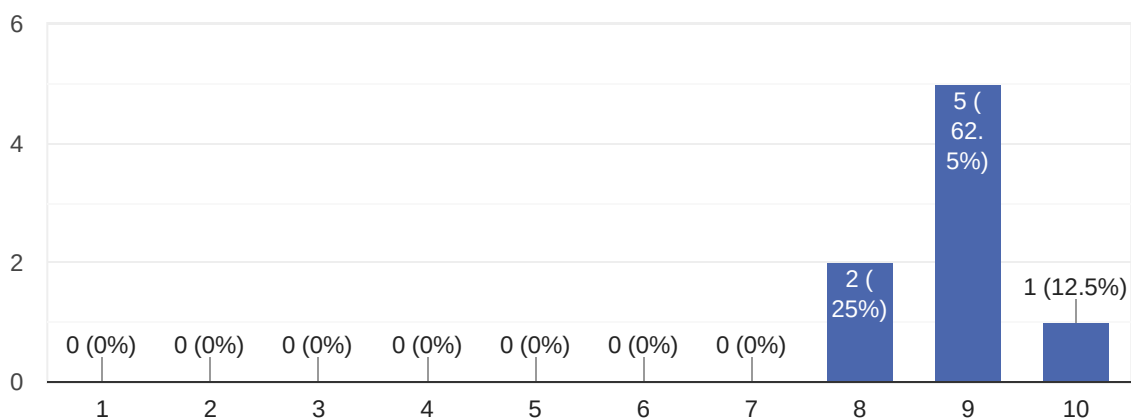
I've liked Maxime a lot overall. Same was for Paul.

Clear and comprehensible language. Regularly stops to poll for questions. Prompt response to questions. Self criticism in order to help improving lecture material.

The course was very well given.

How did the course meet your learning objectives?

8 responses



Comments and suggestions

2 responses

Pleasantly surprised. I expected more low level kernel and driver topics, which I'm not that interested in. I'm happy this was not the case. It wasn't boring at all. That means a lot to me.

Learned a lot.



What part(s) of the course did you like most?

7 responses

Demos, labs, and knowledge points from an experienced preemptive RT programmer.

All was good, just as I mentioned a bit more of improvement in the practical examples.

All

Explanation of all the different tools of profiling, diagnosing.

The flexibility of the trainer to zoom in on specific topics that we're asked

The ftrace tool.

For me the part about the tooling was the most interesting. Analyzing problems with preempt-rt is not trivial, the tooling that was demonstrated will definitely help.

What part(s) of the course did you like least?

6 responses

Would be better to give more explanations on the tools used in the demo and lab. Some commands and options are new to me.

Can't really say

It would have been great if some sort of hardware simulation platform was used so we could easily follow the labs along with the instructor.

The lab showcasing all the profiling and diagnosis tools. Too long. But then again, there is a lot to demonstrate.

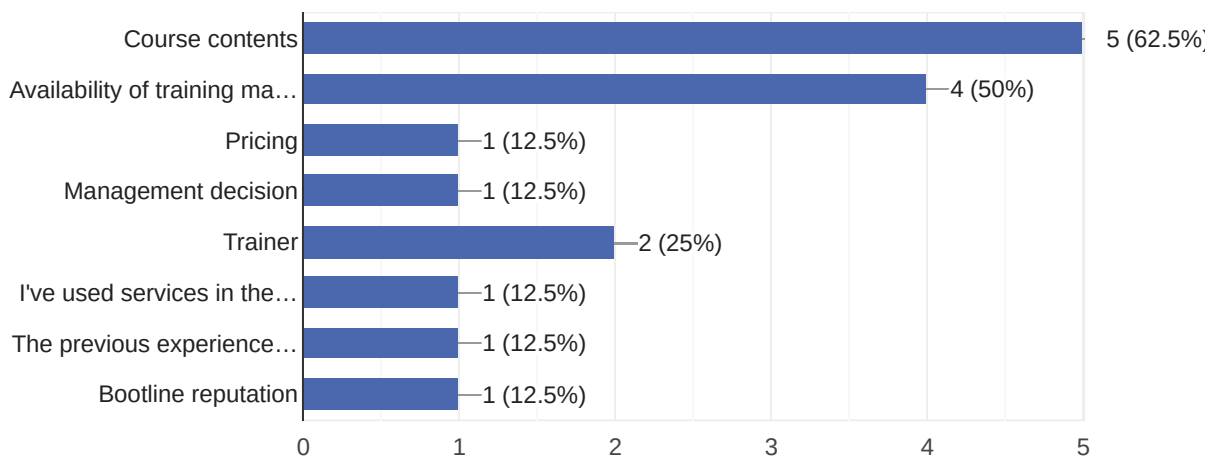
-

/



What reasons prompted you to choose a Bootlin course?

8 responses



Comments

3 responses

My time zone is US Eastern and it would be great if the training courses can start like 30 minutes later (8:30am in my time).

Right now I would like to sign up for the driver development course in March but for some reason that is going start at 7:30am, which way too early especially for a 7-day course.

Was inside the company not informed I was subscribed.

Bootlin is a very respected company in the field of embedded linux. Would not have thought about another company for this specific topic.



Further training needs?

5 responses

Linux kernel programming.

Application programming for my father's company developers, specifically in Qt5 for embedded Linux or other possible alternatives that I honestly don't.

alsa configuration. audio and networking (RTP, controlling jitter, ..)

Zephyr

"Heterogeneous computing". Many SoCs tend to have some co-processors these days. Think about the cortex-m processors found in modern i.MX and TI processors. A course on best practices for building drivers to interact between linux kernelspace/userspace and the firmware running on the coprocessor would be very very useful.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#).

Google Forms

