Hey, welcome back to Open Source Voices. I'm Nicole Huesman.

With more than 25 billion intelligent, connected things predicted to hit the market by 2020, the “Internet of Things” is quickly evolving from promise to reality, and open source plays a critical role in this evolution.

Today I'm excited to be joined by two guests at the forefront of this innovation. Anas Nashif, Software Engineer at Intel, and Carles Cufi, Open Source R&D Software Engineer at Nordic Semiconductor. Welcome, Anas and Carles.

Before we dive in, Anas, can you introduce yourself and what you do at Intel?

Yes, I'm Anas Nashif. I've been working at Intel for the last thirteen years, working for the Open Source Technology Center, mostly involved in open source projects. Three years ago I started working on the Zephyr Project.

Carles, tell us a little bit about yourself and what you do at Nordic Semiconductor.

So, I've been at Nordic for around eight years now and even before that I was already involved with Bluetooth low energy and Bluetooth in general, which is the technology that has really marked my career. At Nordic, after spending a few years developing Bluetooth low energy protocol stacks, I'm now in charge of coordinating the open source contributions from Nordic, specifically to the Zephyr Project and other open source projects.

Q: So, Anas, you've worked in this space for a long time. Can you talk a little bit about how open source plays a role in IoT development?

Yeah, sure. So, basically, when we look at IoT—when you mention the word, IoT, to a lot of people—they associate that with makers, with automating stuff, home automation, home security, and so on. The trend that started a few years back was that people wanted to do this stuff on their own. So that's where open source actually made the first interaction with IoT and the maker community—it was with Arduino back then. But this is why you see a lot of community and a lot of contributions coming from different aspects of this industry and the community.

It actually goes back to why open source is important in general. First thing of course is that code sharing and getting people to contribute to open source projects and getting them involved raises awareness and improves the quality of such projects. The second one is security and making sure people actually see what they get. This is something that we see with Linux. Linux has a very good track record with security because everybody can see the code, they can contribute, they can fix, and so on. The other thing, which is the most exciting part, is that you don't get to do things on your own. So, if I'm actually working on a project and want to develop for IoT, the traditional way of doing things is that companies would actually go and develop things behind closed doors—they had to go carry the doors and do everything on
their own. What open source does in this area is it opens the doors for everybody to work on one common thing, and that's exactly what we are doing with this [Zephyr] open source project. It's not led by one company or one organization; it's open for everybody, and besides the members of the Zephyr project, it's open to any individuals who want to contribute to this project.

Q: You mentioned the Zephyr Project. Can you talk more about the Zephyr Project and how it's evolved since its introduction in early 2016?
Yes, it was introduced in 2016, but obviously, we worked on it much before it was launched. We were looking at industrial partners and key leaders in the industry having the same problem we were having. We wanted to develop, or have, a software stack for IoT. We didn't want to introduce something that was industry-specific. We wanted to share and get something out there where we can work with the industry partners, where everybody can contribute with their own strengths and their own areas of expertise. So, the initial code came from Intel, but we were talking with all our partners in the industry and we launched the project back then with three platinum members—Intel, NXP and Synopsis—and soon after, we had other companies join, like Linaro and Nordic, and we have now a few silver members from different areas of the industry. We have been growing in terms of contributors and members. The main problem is trying to avoid fragmentation, where everybody goes and implements IoT solutions on their own in their own way.

Q: It sounds like we're really seeing a lot of participation across the community in contributions to the Zephyr Project. Carles, given your experience with the Zephyr project, what excites you most about it?
Well, the excitement that I feel for the project, and that I think many of my fellow Nordic colleagues feel as well, is that, having seen a landscape in IoT in particular with microcontrollers that has evolved from a very simple one in terms of cores, in terms of technical requirements, in order to drive a small embedded device, growing into a complex, multi-core, multi-peripheral system that requires much bigger code bases and much more effort to develop those, I think that what excites us is seeing other companies having the same issues that Nordic also foresees, and with complexity growing, seeing that we're not the only ones, many others have joined since, sharing efforts to solve these problems. That I think is one of the points that, at least, from my personal perspective, and I know that others feel the same, is exciting—it's the fact that we've come together. We've seen companies that have used our chips, that have shown interest for the Zephyr project, because they also see that, in the future, the complexity is growing at a rate that is not sustainable for a single company to develop. So, this breaking with the fragmentation and joining efforts to develop the software that is as best as it can be, that's really what drives us and excites us about the project.

Can you talk about some of the specific use cases? How are folks out there using the Zephyr project?
Zephyr is flexible and configurable enough that it adapts to a lot of use cases. We've seen people use Zephyr for small, accessory-type devices where you have a small device with a tiny
battery that needs to last for days, if not months, sometimes even years, that connects to a mobile phone, giving you feedback from a sensor that's embedded in the device, that we've seen implemented with Zephyr. But now what we're seeing as well is more complex scenarios with things like Bluetooth low energy mesh or 15.4 base meshes, whether it be thread or any other protocol on top of 15.4, and those are more complex because they require PCIe stack, they require additional security layers, they require complex ciphers, and all of those are perfect use cases for Zephyr because that's where it shows its strengths, having all of this complexity built-in and also having gone through the process of being reviewed by people that come from different areas with different levels of expertise in different types of software development—that is really where we see the golden use cases for Zephyr, although it will obviously always support the smaller ones because being extensible and configurable is really at the heart of Zephyr—that's actually one of the points that excited us at the beginning was when we saw that from a single code base, you could build a tiny project running on our smallest core to a massive project using OpenThread and all of the most advanced technology running on our biggest core, so in that regard, I think Intel and the others who started the project learned the lessons from Linux making it a really, really scalable piece of software.

Q: So, you guys mentioned security, and of course, as more and more smart “things” connect to each other and to the Internet, security becomes a big issue. Anas, can you talk about the project's approach to security?

Sure, since the project has started, one of the main highlights of the project was development had been with security in mind. So, development of the Zephyr project, and producing codes and features was combined with security considerations—so, security was not an afterthought. Also the fact that it is open source, so there are lots of eyes looking and reviewing the code, and this is the core of the Zephyr project. Contributions come in, never mind if you are a big or a small contributor, if you are an individual or if you are a company, your code has to go through the same review process. Everybody gets a say. We have a governance board, we have a technical steering committee, we discuss a range of topics including security, scalability, and so on. In addition, we have a security working group that deals with the security aspects of the project, making sure that we deal with exploits and potential issues that have been introduced, and working with industry leaders in the security domain to ensure we cover all areas that require security considerations. For example, we have a member with a track record in IoT security who has been participating in the project since Day 1 and has contributed a lot. Nordic is participating and has added security-related features to their products. Over the last year, we have added memory protection, isolation, and software updates—both over-the-air and offline. Software updates are a key feature—any project without software updates can't qualify as secure. As much as we pay attention to security and making sure our code is secure, there are always issues and having a mechanism in place to update and push updates to systems in the field—if you don't have that, then you basically don't count these days. Everything is connected, and we are talking about millions or billions of devices that could be deployed, and without having infrastructure in place for that, it becomes unviable in this case. The other thing we have as a part of the whole development process is we do security scans on a weekly basis. Every single line of code that has been
introduced goes through static analysis. We have dynamic analysis as well. We are introducing all kinds of tools around security. As Carles mentioned, scalability is actually a major thing. Being able to build a product from the smallest to the biggest use cases using the same code contributes to security as well because you don't have to go and start something new for a new use case or problem that you have. You use something with a track record, and that actually helps in many ways. Having Zephyr as a complete solution, and not a collection of solutions that you pick and choose pull from different sources, so that's why we have our own IP stack, so it's all well developed as part of Zephyr. We have the Drupal stack, host and controller, and all middleware and all these components work together to create a framework that is secure and scales from the smallest to the largest use cases.

Q: Networking is another big consideration here. Carles, can you talk about how the Zephyr Project is evolving to meet the needs of networking in IoT?

When it comes to networking and IoT, it really is mostly about IPV6, version 6 of the TCP IP suite of protocols, and in that regard, I think Zephyr is pretty well covered. The protocols have been in development for more than a year and a half, and they're now relatively stable and ready for prime time. They are already being used in commercial applications and on top of that what we're doing right now is adding additional features. Among them is TLS. TLS, or DTLS, is important because it allows for secure, encrypted connections over the Internet, which lately, the spread of security certificates, and the fact that most connections to a website are secured using exactly that, is also important to IoT because you often cannot rely on the transport layer being secure, so adding support for those, and I think, in a way, that applications do not have to worry much about it, that adding the encryption feature doesn't require the application developer to rewrite the application or to make major changes has been the goal, and in that regard, that's what the latest efforts are directed towards—getting an API that provides the security credential management, the establishment of encryption, and obviously, doing all of the required encryption. Beyond that, Zephyr integrates very well with OpenThread, which is another open source project, led by Nest, and what we've done with OpenThread is that we've written the necessary code that allows for Zephyr and OpenThread to work together well. We didn't want to re-implement the whole thing, we didn't want to reinvent the wheel, instead we made those two projects cooperate well together so that they complement each other, and now you can run Zephyr with OpenThread and therefore establish a thread network. When it comes to networking, sometimes the term can be misleading. I'd also like to include Bluetooth Low Energy (BLE) Mesh under this umbrella term. BLE Mesh has been proven one of the most popular technologies with our users. There's constantly questions about it on the chat channels or on the mailing list or issues being created. We see a lot of people using BLE Mesh and that's for a wide range of applications—from lighting to industrial control to even simpler relays of data, for example, for smaller control systems. The BLE Mesh implementation in Zephyr is well regarded. In fact, it's been adopted by other open source operating systems, and we are proud of this, and we must continue to make this implementation the best it can be, and we've been doing this by adding things like storage support so you can store the rest of the nodes information and provisioning and etc. And there's more to come.
Q: So, there are so many different vendors in the IoT environment. How do standards play a role?

The majority of members are participants in many of the standards groups, and we try as much as possible to drive our roadmap and our features based on these standards. So, we don't try to come up with our own solutions. This goes back to security, as well as interoperability. Just like Carles mentioned Bluetooth Mesh and OpenThread, machine-to-machine, end-to-end, and OCF can also be deployed on top of Zephyr using an existing implementation. Whenever solutions and solution implementations of certain standards exist, we try to work with these and work with the project to make it fit on top of Zephyr, and we try as much as possible to follow the standards. Beyond standards for networking and Bluetooth, we're talking about standards for coding guidelines, C standards, how to write code, best known methods in the open source community, how to do things, so we don't try to go and do things our own way, we try to adopt, and we try to participate in conversations with industrial standards groups or other open source communities and make sure that Zephyr is well positioned to be a leader in these areas.

Q: Thanks for helping us understand where the Zephyr Project is today. Let's shift a bit now and talk about what's next for the Zephyr team.

A couple of things that we're really excited about that are coming up—and they're already being developed now, but obviously, full support for them will come in later releases. In particular, the ARM architecture, which is very popular among embedded developers, especially Cortex-M line of ARM processors, they've recently released a new version of their architecture called ARM vm8 and they have support for something called TrustZone. That is essentially a mechanism to ensure the reliability and security of applications running on an embedded processor. This gives you the opportunity to isolate the extreme sensitive code—it's a little bit like a secure enclave where the secure code runs in its own isolated partition and then the unsecure code doesn't have access to certain hardware peripherals, memory areas, and all of that is done by the core. We were probably one of the first, if not the first, open source project to support the cores natively as in building for them and adding support for some of their hardware features. We're expanding now to full support for the TrustZone technology as well as additional features that come with it. Beyond that, I'd like to mention mcu-boot, which is a boot loader that uses Zephyr to build. It's not directly part of the Zephyr project, but it works tightly coupled together. We also have plans to add a lot of new features to this bootloader, which we want to become a little bit like the standard where we shift towards having a single open source bootloader. And there's more, such as better Windows support. Windows has always been important for us, I would go as far as to say key for us, so being able to build, debug, flash, the whole development cycle needs to be as good on Windows as on Linux, so I think better Windows support, features that catch developers’ eyes and also make them more productive.

Q: And Anas, what are you looking forward to?

All of the things that Carles mentioned. Plus one thing that's happening in the Zephyr project is, when people think about IoT, they usually think of sensors and stuff like that, things that
are used for home automation and so on, but it's actually bigger than that, it's basically about everything connected. So, we are moving the project in this case to support complex industrial use cases. One thing, for example, is multiprocessor systems where Zephyr works alongside Linux, supporting industrial protocols, supporting time sensitive networking, supporting virtualization where Zephyr works alongside Linux to do real-time sensitive activities and run secure instances for open applications. So, we are growing beyond things like microcontrollers into supporting a wide variety of use cases, and as Carles mentioned, enhancing the developer experience, whether it's on Windows or Mac or supporting different IDs, and most importantly, collaborating with other existing projects and making Zephyr part of whatever projects deliver. For example, this has already happened with JavaScript, with MicroPython, and a few other projects that are using Zephyr to do open source development and to develop their own use cases. So, we don't want to be working in a silo; we want Zephyr to be used in other projects just the same way we are using other projects' code and software.

Q: Thanks! For developers who want to dive in, where can they learn more?
Developers can go to zephyrproject.org website, which has all the information needed. The project itself is developed on Github. They can join us on IRC, chats, mailing lists, and different forums. Everyone is welcome to join our open meetings. Feel free to join our discussions, ask questions and look through the documentation. They can also join the discussions in our weekly meetings.

Thanks so much for joining us today. It's been great to talk about where the Zephyr project is today and where it's headed, and we're looking forward to hearing more in the future.

Until next time. Thanks for listening.