

[← All blog posts](#)

Oct 15, 2021 by *Lukasz Erecinski*

October Update: Introducing the PinePhone Pro



Welcome to what will likely be remembered as the most important community update of 2021. Today we're introducing the PinePhone Pro and announcing that both PinePhone Pro and PineNote pre-orders are now open to developers. I am well aware that this announcement will overshadow everything else that I may wish to report on, so I intentionally chose not to include other exciting news in this post. I will write a follow-up piece covering other topics later this month - stay tuned.

But that's not all. All PineTime owners ought to read this month's update concerning InfiniTime. The firmware recently received updates which significantly improve the user experience and BLE connection stability. JF also writes about the project's growth and ongoing project management restructuring.

You can watch the synopsis of this month's community update on YouTube (embedded below) as well as on [Odysee](#) and [PeerTube](#). To stay up-to-date with PINE64 news make sure to subscribe to this blog (subscription widget at the bottom of the webpage), follow [PINE64 Telegram News channel](#), the [announcements channel in Discord](#) as well as our [Twitter](#) and [Mastodon](#).

I'd like to thank [JF](#), [Alex](#) (clover), [Brian](#) (33YN2) and [PizzaLovingNerd](#) for their contributions to this community update.

N.B. Comments on the blog post need to be in English and follow our [Community Rules and Code of Conduct](#).

INVIDIOUS

To see this content, select an instance, or [see directly on YouTube](#).

Region	Domain	Health	Action
--------	--------	--------	--------

Community Update video synopsis

TL;DR

- **Housekeeping**
 - Wiki *getting started* and orphaned Wiki pages help needed
 - KDE turns 25! Happy birthday!
 - Good job on the PinePhone Pro launch team!
- **PinePhone Pro**
 - PinePhone Pro announced - start by reading the [PinePhone Pro website](#)
 - Features RK3399S SoC made by Rockchip for to the PinePhone Pro
 - Ships with 4GB LPDDR4, 128GB eMMC, Gorilla Glass 4™ IPS panel, 13MP main camera, USB 3.0 via USB-C, native video out via USB-C + countless improvements
 - Pogo pins & privacy switches are present: the PinePhone Pro is compatible with the keyboard and back cases
 - Developer pre-orders and *Explorer Edition* coming later this year for **\$399**
 - Developer units are in production due to be delivered soon

- The blog post focuses on the decision-making process and explaining why now is the right time to introduce the PinePhone Pro
- The PinePhone / PinePhone Pro keyboard enters production; keep an eye out for a follow-up post late this month
- **PineNote**
 - PineNote developer pre-orders are now open
 - Dev units ship with EMR pen, magnetic cover and a UART USB-C breakout board for debugging
 - CE /FCC certification is now complete
 - Production is underway and units are being delivered from factory now
 - Biggest software problem relates to initialising the panel under Linux; developers have ideas how to accomplish this, but it remains untested
 - Availability of early adopter units depending on software progress - likely later this year
- **PineTime [by JF]**
 - InfiniTime 1.5 brings a highly requested feature - alarm clock!
 - InfiniTime 1.6 releases just 2 days after 1.5 with a single line commit, which makes all the difference - InfiniTime has now stable BLE connection!
 - Restructuring of the InfiniTime project, creating a Github organization and taking on contributors to help with issue triage, PR reviews and merging
 - A word on donations to the InfiniTime project (and to other PineTime efforts)

Housekeeping

Before we get into the exciting news there are a few housekeeping items on our itinerary. The first of which concerns Wiki contributions. Let me start by thanking everyone who has contributed to our Wiki - I've seen quite a few Wikis and I think that ours is among the best out there. That said, there are a few sections that are either incomplete, neglected or simply haven't received an update in a long time. There is a list of [orphaned pages](#) that need to be integrated into main articles, reworked or completely removed. More importantly, however, the [getting started](#) section - meant as a technical tutorial for complete novices - has not been maintained in over a year. While each device page usually offers a segment explaining how to get up and running, the global *getting started* section remains relevant because it contains information for less technically capable users. If someone is willing to volunteer their time to restructure this section, update it with information and give it an overhaul, then please reach out to me or one of the mods in our chats.

Yesterday our friends at KDE celebrated their 25th birthday. As many of you surely know we're very closely partnered with KDE and have recently been made [patrons](#) by

[KDE e.V.](#) KDE's help and support over the years has been very important to us: most of our devices, including the Pinebook Pro, the PinePhone, and now also the PinePhone Pro, all ship with Plasma installed by default. I [told this story](#) publicly once already, but it's worth repeating - the PinePhone owes its existence to the promise that Plasma Mobile held at the time of the project's inception. Suffice to say we're very fond of KDE and their team members, many of which we have the pleasure to call our friends. So, then, I want to say a deep-hearted happy birthday to KDE developers and their community. Please head over to their forums, social media, chat or elsewhere and make sure they know their work is important to our community and the open source ecosystem as a whole.

Note: This image preview protects your privacy. Click to load the video from the Google servers.

KDE's 25 anniversary video

Finally, I want to say thank you to our team for doing a great job on the PinePhone Pro launch. And I don't mean the people who made the hardware possible, although they deserve a special round of applause for pulling it off under current manufacturing circumstances. To those who put together the trailer and the renders, created the website, put together and tested the pre-order system, and to all those who helped ensure we're on track to have everything ready on October 15th - I am very grateful for your effort and help. Well done.

PinePhone Pro

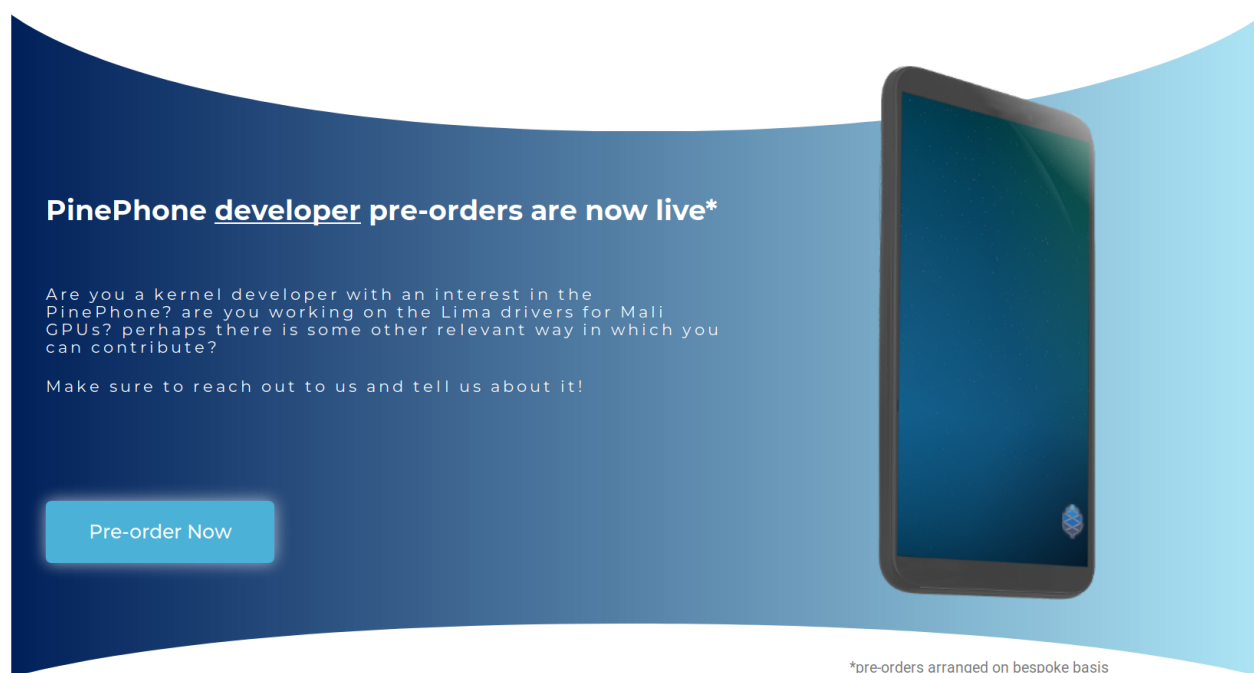
This month I have the pleasure to introduce the newest member of the PINE64 family - the PinePhone Pro. It is a fast smartphone with premium features, built from the ground up to run mainline Linux operating systems. In anticipation of great interest in the nitty-gritty hardware details, we published a dedicated webpage complete with specifications, availability information, device renders and an

extensive FAQ. I strongly suggest you go and read that prior to continuing with this section. Seriously, go read it first.

PinePhone Pro webpage: <https://pine64.org/pinephonepro>

Instead of reiterating the information on PinePhone Pro's website, I'll use this section of the community update to explain why we decided to introduce a more powerful smartphone and discuss some of the key design choices.

Let me start by setting the stage. At the time of the original PinePhone's introduction to the market there were only a handful of mobile Linux operating systems available, none or few of which ran regular Linux with open drivers. The goal of the project was to provide developers with an open, readily available, and inexpensive platform to drive Linux on mobile forward. The PinePhone ended up giving rise to new OSes, such as [Mobian](#) or [DanctNIX](#) (Arch), and helping existing projects such as [Plasma Mobile](#) reach new heights. The PinePhone's introduction also attracted interest from desktop Linux OSes such as [Manjaro](#) and incentivized established mobile projects, such as [UBports](#), to support mainline Linux.



Original PinePhone developer pre-order announcement - a blast from the past :)

Fast forward to today, many of the 20+ operating systems have reached a mature-Beta status. All core functionality has been enabled and we're seeing more complex features being added. Mobile Linux may not yet satisfy the needs of mainstream consumers, but we're now at a point where open-source enthusiasts are willing to give a Linux smartphone a go. MMS messages, Android app support via Anbox or Waydroid, 24hrs of standby time with reliable calls and SMS, accelerated camera viewfinder, and photo post-processing are just some of the features that led people

to start daily driving their PinePhones. Now that software has reached a higher degree of maturity, introducing a fast smartphone with premium features makes sense.



Here it is - beautiful, isn't it?

The announcement of the PinePhone Pro is therefore an acknowledgment that our journey with mobile Linux is entering the next stage. You could say that it denotes a shift from being 'primarily development-focused' to 'technically-inclined end-users centered'. This isn't the most elegant way of phrasing it, but you get the gist. With the PinePhone Pro we deliver hardware that will run current operating systems effortlessly and with some room to spare. But raw performance isn't everything - attention has also been paid to the things that matter the most in a phone used daily: the screen, the camera, storage and RAM capacity, and the overall fit and finish were all central during the design phase. A lot of consideration went into each component choice, and many decisions were discussed with developers over the course of the past year.

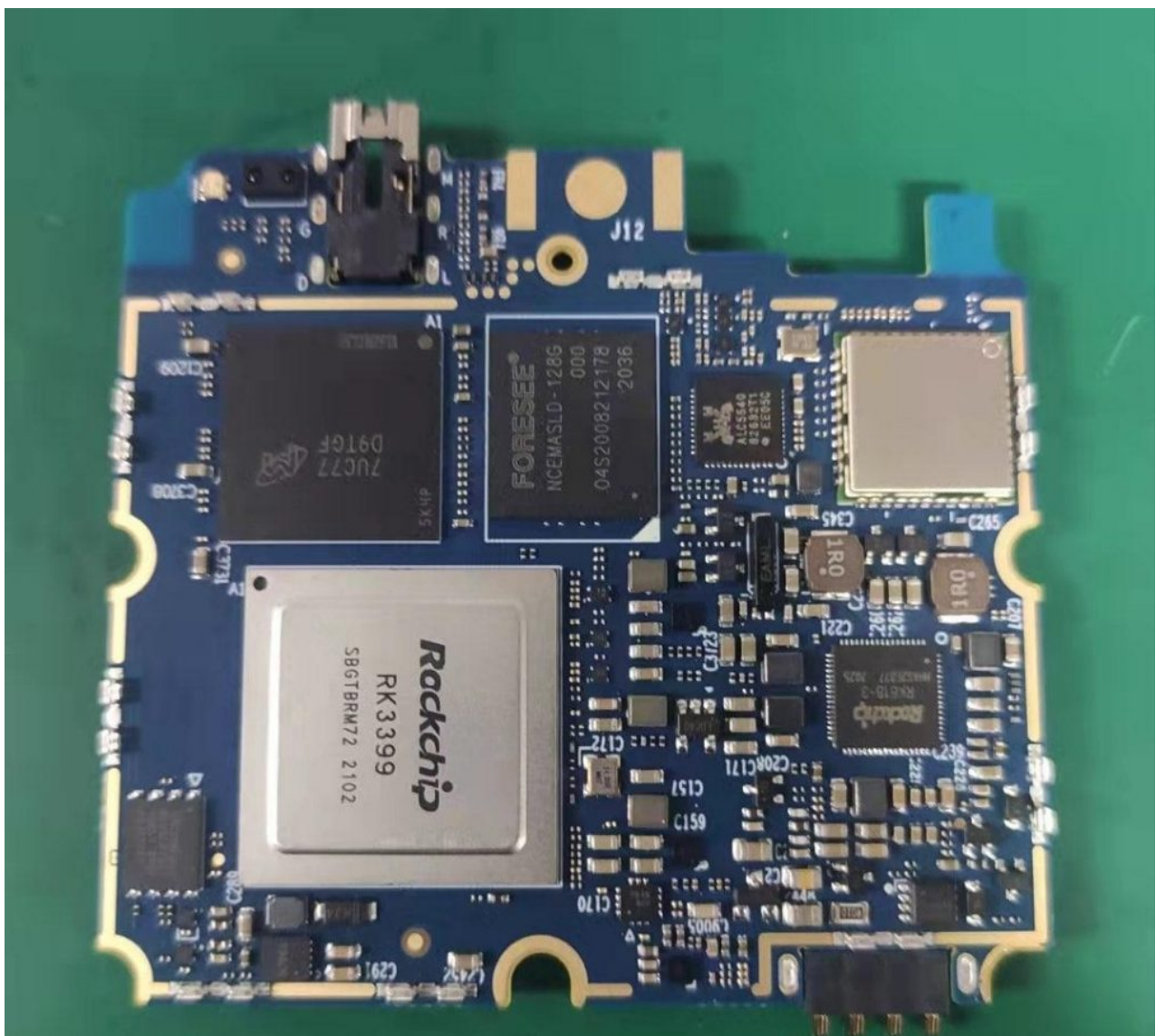
https://www.youtube.com/watch?v=pCxDeMdr_fo

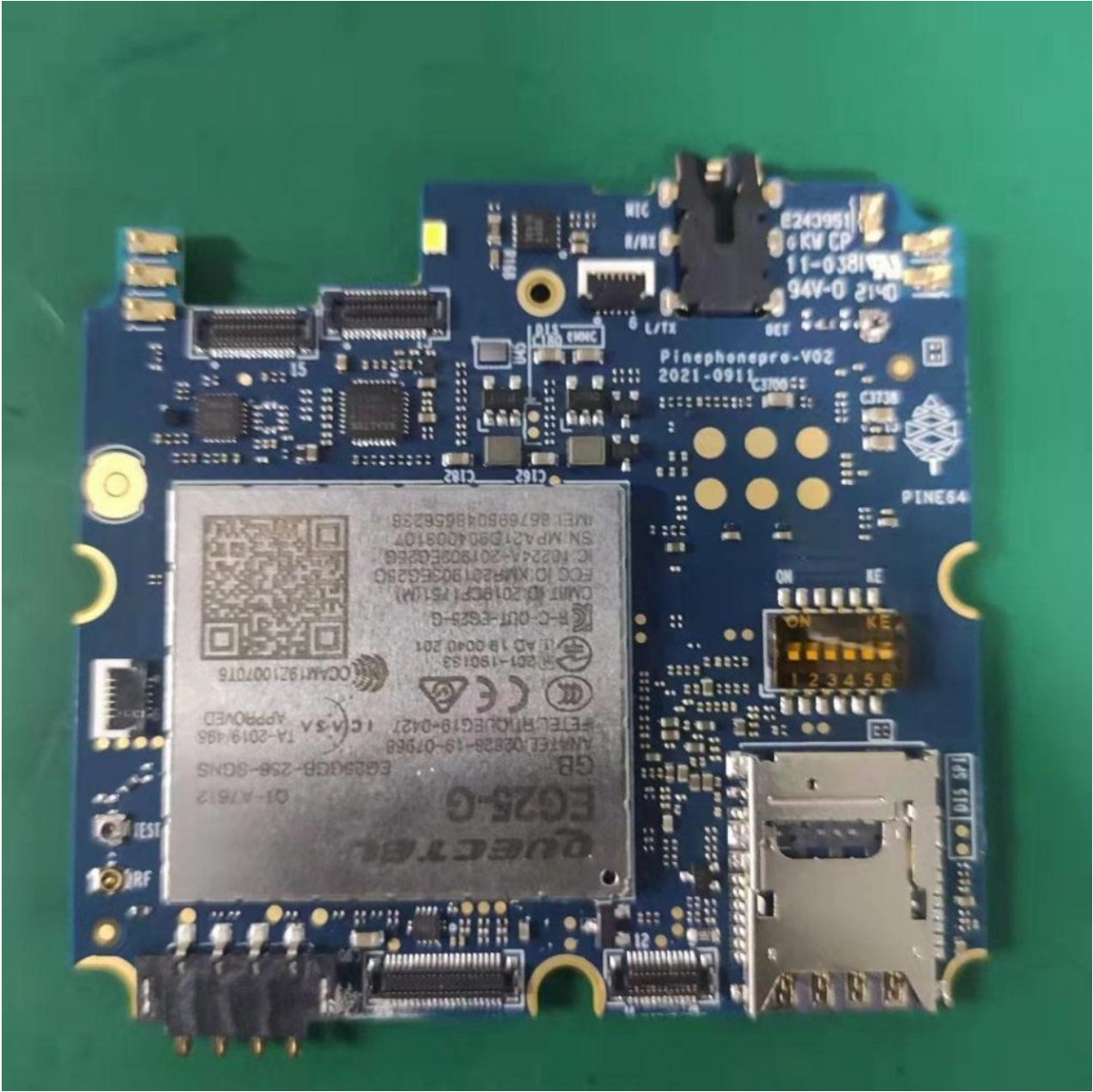
Martijn Braam, a postmarketOS developer and our friend, takes a look at the PinePhone Pro

Before moving on to discussing design decisions, let me clear up a few things: the PinePhone Pro is not a second-generation PinePhone, but rather a higher-end PinePhone. The original PinePhone isn't going away anytime soon either - in fact, we have thousands of units in the pipeline. The PinePhone Pro is pogo-pin compatible with the original PinePhone, which means that peripherals, such as the keyboard and add-on back cases, will work with both devices. So if you are getting one of the

add-ons for your PinePhone then don't worry, they'll work with the Pro if you decide to get one in the future.

Finally, let me address some of the key design choices. We chose the RK3399 as our internal development basis for two reasons: we are well acquainted with the platform and it is nearly fully mainlined (the ROCKPro64, our RK3399 development board, recently received [Arm's SystemReady IR Certification](#)). But we aren't using a regular RK3399 chip in the PinePhone Pro. We worked closely with Rockchip's engineers to tweak the original SoC to fit our use case - the result of the cooperation is the RK3399S, made specifically for the PinePhone Pro. RK3399S chips are binned and voltage-locked so that they conform to a smartphone chassis thermal and power consumption envelopes. Aside from great performance, the SoC also features a stellar feature set, including native video output via USB-C, support for high-resolution cameras and USB 3.0 data transfer speeds, just to name a notable few. But choosing such a well-supported platform also means that porting existing OSes to the PinePhone Pro will be fairly trivial. While it may take some time to achieve complete feature parity with the original PinePhone, the road to full system functionality will be relatively short.







A look at what's inside the PinePhone Pro (mandatory disclaimer: images of a prototype and may not reflect end product)

As I previously mentioned, we consulted component choices with the developers. This informed many of our decisions, such as using the main camera sensor on the original PinePhone for the PinePhone Pro's front-facing shooter. Similarly, together with the developers we settled on the main 13MP Sony sensor which has good mainline Linux support, produces nice photographs, and has the right physical dimensions to fit the chassis. Early in the process we decided to go 'all-out' and outfitted the PinePhone Pro with 128GB of flash storage and 4GB of LPDDR4 RAM. Needless to say, we kept all the things our community loves about the PinePhone, such as the removable back, privacy switches and a removable battery (N.B. the battery on the PinePhone Pro has also received an upgrade).



PinePhone Pro

PinePhone

PinePhone Pro (left) next to a PinePhone (right), can you spot the subtle differences?

Lastly, let's discuss decisions that determined the look and feel of the PinePhone Pro. Keeping backward compatibility with the original PinePhone was our priority, so from the get-go, we knew that whatever choices we made they couldn't deviate too much from the original design. The face of the PinePhone Pro features a *Corning Gorilla Glass™* covered in-cell IPS display that produces a nice and vibrant image. It has good viewing angles and gets really bright. The decision to maintain the original PinePhone's screen resolution of 1440x720 was made early on; higher resolution panels consume more power and increase SoC's load, resulting in shorter battery life and higher average thermals. A few extra pixels aren't worth it. As for the back case, it features our trademark understated matte finish. The development unit I had the pleasure to use was considerably more resilient to oily fingerprints than the original PinePhone. It is difficult to describe a tactile experience so I'll leave it at that - I hope that photos do it justice. Finally, the entire device is slightly thicker than the original PinePhone - by 2mm to be precise. The change in thickness was made to accommodate thermal dissipation and the new display.



PinePhone Pro (left) and PinePhone (right) thickness comparison - it's barely noticeable

That's it, I think I've covered everything. If you have any further questions, make sure to leave them in the comments section. We currently have a small production run of the PinePhone Pro rolling off the factory floor - these units are aimed strictly at developers. The idea is to give devs time to get acquainted with the hardware so that they can start porting their operating systems to the new platform. If you are a developer and want to get your hands on the PinePhone Pro early, then I suggest you promptly place a [pre-order](#) as the number of units is very limited. Early adopters will have to wait a little longer to get their hands on the PinePhone Pro *Explorer Edition*. That said, it won't be an extensive wait - we aim to start shipping the *Explorer Edition* late this year or very early the next (shipping will depend on production scheduling and the CE/FCC certification process).

Note: This image preview protects your privacy. Click to load the video from the Google servers.

I spoke to [Destination Linux](#) about the PinePhone Pro, go have a listen

Before I jump into the PineNote section - I know many of you are waiting for news about the PinePhone/PinePhone Pro keyboard and add-on cases. Due to the PinePhone Pro's launch, and all the preparations which lead up to today, I've had very little time to keep track of hardware and software developments. All I know is that the keyboard has finally entered production. I'll take the next week or so to catch up on everything and write a concise follow-up blog post that will hopefully answer all your questions. I'll also make sure to query developers about any major developments that have taken place in the past month. Keep an eye out for the post around October 30th.

PineNote

It has been three months since we announced the [PineNote](#) - an open e-paper device, based on our range of [Quartz64 single board computers](#). It features a fast quad-core SoC (RK3566), 4GB of LPDDR4 RAM, 128GB of eMMC flash storage and comes equipped with an EMR pen-capable fast refresh e-paper panel. In case you missed it, I suggest you read the [August community update](#) which I dedicated to introducing the new hardware. In time, the PineNote will run Linux (and who knows, maybe even *BSD) with an open software stack, allowing you to do a range of things that are impossible on traditional e-paper tablets. Sky's the limit. But the path to achieving such functionality is a long one, and it will require skilled developers to get PineNote's core functionality off the ground. Existing desktop and mobile environments, such as KDE Plasma, will also need to be altered to make a viable user experience on the unorthodox grayscale display. The bring-up process has already started and a select number of developers have received pre-production PineNotes in the past weeks - but many more contributors are needed to accelerate

the development process.

[Last month](#) I reported on PineNote's current software status in detail, but I haven't had much of a chance to catch up on it since. It is my understanding that the biggest hurdle to overcome remains e-paper functionality - more precisely, getting the display to initialize under Linux. Just prior to writing up this section I scrolled through the PineNote dev chat backlog, and the good news is that developers are actively working towards making this a reality. Aside from the display, development on the RK3566 is going really well and it appears that we'll have a very well-supported SoC on our hands in a matter of months. This process will surely be propelled further by more people from the development community getting their hands on the hardware.



A keyboard with the PineNote? sure why not - via [Danct12](#)

Speaking of the hardware - I am happy to let you know that the initial PineNote production run has successfully rolled off the factory floor and the device has received its CE/FCC certification earlier this month. This means that [we're now ready to start taking pre-orders](#) from developers and contributors who are keen to help us realise the vision behind the PineNote. These units will ship without an operating system installed, with the device in a flashable mode by default. Devices aimed at developers will also ship with the EMR pen, a magnetic cover and a USB-C

UART break-out board to aid with debugging. We will start shipping later this year, possibly as early as late November; keep an eye out for updates on our social media and news feeds as well as the [shipping, stock and availability](#) page.





PineNote production line

As for early adopter units, their availability will largely depend on software development progress. Ultimately, the schedule will be dictated by when the e-paper panel gets enabled on Linux. As I already mentioned, developers have some ideas on how this can be achieved (click [here](#) to see what Caleb had to say about it), so it may be a matter of a few weeks before we reach this milestone. However, admittedly it is also perfectly conceivable that enabling the e-paper panel will prove more of a challenge, and consequently take a longer time. In a sense, the ball is now in the developers' court and the rest of us (including the production team) have to wait for a green light to proceed. As always, I'll keep you updated on how things are progressing in future blog posts and updates.

PineTime (and InfiniTime) [by JF]

In last [month's update](#), we announced the release of [InfiniTime 1.4](#), which greatly improved the reliability and responsiveness of the touch panel and added a nice color picker for the PineTimeStyle watch face. Less than a month later, we published not one but two releases of InfiniTime! These 2 releases have allowed the project to reach a new milestone in terms of **daily driver readiness**. Let's take a closer look at what these releases bring to the table.

InfiniTime 1.5 *Huckleberry*

If I had to guess the feature InfiniTime users missed the most, I would say it's the alarm app. Users would often ask if an alarm app is available in the chat rooms, in response to the toots and tweets I write to announce new releases. My answer was always the same: that we'll have an alarm app when someone will work on it. It's as simple as that. InfiniTime is an open-source project and all contributors are

volunteers working on it in their free time. There's no priority, no roadmap and no set date for the next release. There are only people working on whatever they want whenever they want. Finally [this pull-request](#) was submitted.

Basic alarm app #662

[Edit](#) [Open with ▾](#)

Merged JF002 merged 7 commits into `InfiniTimeOrg:develop` from `mruss77:basic-alarm-app` 15 days ago

Conversation 63

Commits 7

Checks 0

Files changed 14

+537 -2



mruss77 commented 22 days ago

Contributor

😊 ⋮

I added a basic alarm app that allows you to set one alarm to go off once, every day, or on weekdays (M-F).

I know there is discussion in [issue 366](#) about a full-featured alarm app with multiple alarms, which looks great. This basic alarm meets my simple need to wake up on time in the morning.

The UI is one screen based off the Timer app UI:



Reviewers

JF002

Riksu9000

Avamander

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

1.5.0

Linked issues

Successfully merging this pull

The alarm clock app pull request

So, why did it take so long for this PR to be submitted? I can only offer my point of view. In my opinion, the alarm app presents 2 challenges. The first one relates to implementing the alarm in a way that it will wake the watch up and notify the user at a specified time. The other challenge is the design of the user interface. The UI needs to allow the users to specify an alarm time and the days the alarm should be triggered. Some developers are comfortable with lower-level development but are really bad at UI (I fit in this category), while others can do wonders on the UI side but struggle with low-level layers. Perhaps it is a bit of an exaggeration but it isn't far off the truth either.

I therefore really like how [mruss77](#) approached implementing this new functionality - he created the first version with basic features (only 1 alarm, with very few options and customizations) which are already functional and left room for improvements in

further releases. So, now we have our first alarm app! It allows you to specify time and days (once, every day, or weekdays), to enable/disable the alarm and to check when the next one will be triggered. And, obviously, the app will notify you when the alarm is triggered with a small vibration.



A look at the alarm clock app - picture via [Nico Cartron](#)

Another important addition to this release is the [rework of the BLE advertising strategy](#). In Bluetooth low-energy (BLE), advertising allows a device to broadcast *advertising* messages to announce itself to the other devices. These devices can then detect advertising messages when they are in scanning mode. [James](#) worked on refactoring the advertising logic in InfiniTime so that it now follows state-of-the-art recommendations, which should (and do) improve the interoperability with more BLE-capable devices as well as connectivity with the companion apps.

Last but not least, [Tim](#) ensured that the [date and time are persistent](#) across reboots. It means that InfiniTime is now able to recover the date and time when the watch is reset (either manually or due to a bug). I think this is a nice quality of life improvement as users have (had, to be precise) to regularly reboot their watch to recover the BLE connectivity (more about this later). This was a bit tricky as the NRF52832 (the CPU that's running the PineTime) does not provide any register or memory areas guaranteed to keep content when the CPU resets. We also cannot afford to store the time in either the internal or external flash memory, as it would wear them far too fast. Fortunately, Tim found a way to store the date and time in RAM and to ensure this location will not be erased when the watch reboots!

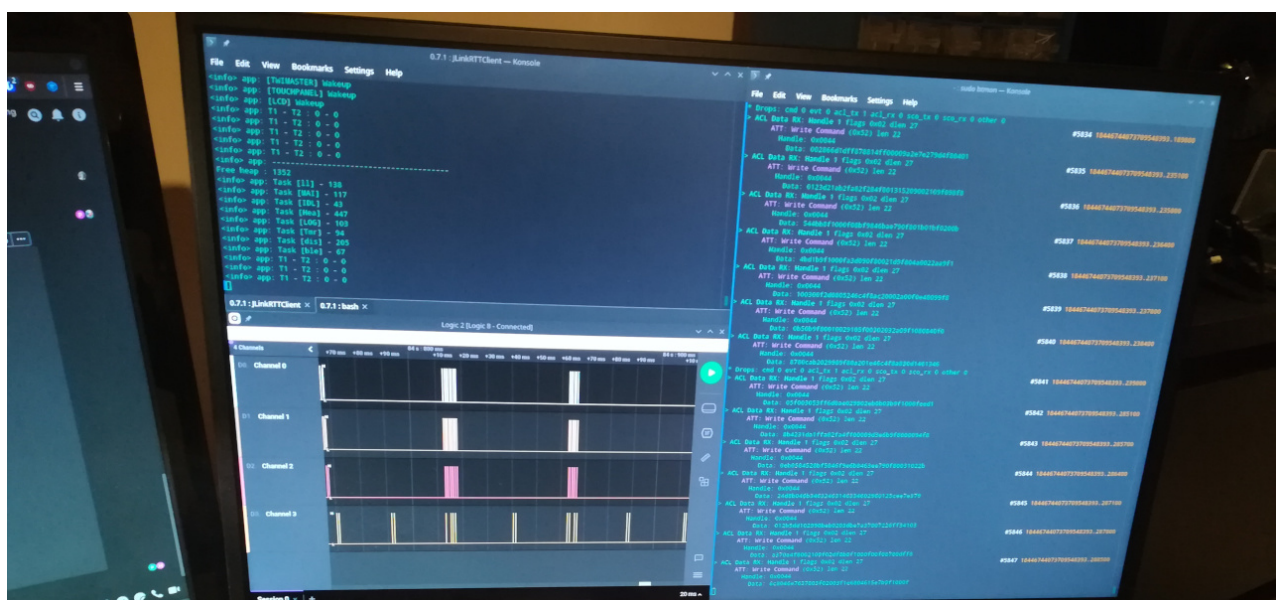
InfiniTime 1.6 Ice Apple (we have a BLE fix!!!)

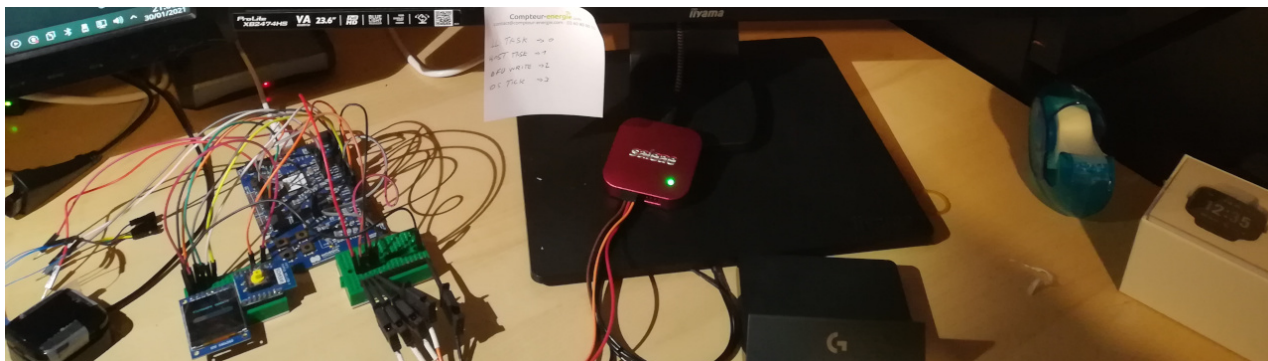
Only 2 days after the release of 1.5 I decided to publish InfiniTime 1.6. This release is really small in terms of the number of commits (only 1 commit to be precise) and changes (a single line of code was modified), but it is ****huge**** in terms of its impact for end-users: BLE connectivity now works a lot better than before!

Let's go back in time to April this year. At the time Lukasz and I felt that InfiniTime

reached many milestones (with the integration of the step counter and heart-rate sensor) and deserved an official 1.0 release. The only reason why I hesitated was BLE connectivity, which was not as stable as I wanted. BLE would work for a time and then simply stop. I couldn't figure out why that was happening, and I didn't know if other users were experiencing this issue too. As you know, we finally decided to announce InfiniTime 1.0 knowing that we could fix the issue in future releases. This proved to be the right move. InfiniTime and the whole PineTime project gained a lot of momentum and InfiniTime 1.0 release allowed PINE64 to ship the PineTime as an 'end-user' product rather than a 'developer kit'. Most importantly, the majority of users were happy with this firmware release. Buuuut I still noticed the BLE issue, and it was increasingly more obvious I wasn't alone. Many users were reporting BLE connectivity issues, saying they had to reset their watch to be able to reconnect to the companion app. I've been chasing this issue ever since and I spent countless hours trying to understand why the BLE radio would stop emitting data after a few hours. Other developers quickly joined me [on Github](#) and in the chat rooms. They also spent a lot of time testing and debugging. In the end, we managed to prove that most of the BLE-related code is rock solid, but we still couldn't understand why it was failing after roughly one day of usage.

So, why was this issue so hard to debug and fix? Debugging wireless communications can be very tricky since you cannot connect an oscilloscope or a digital analyzer using wires to figure out what is happening or what went wrong. I tried to set up a [BLE debugging environment](#) hoping to learn more about the issue... with no luck. Also, the BLE stack ([NimBLE](#)) is a very complex piece of code that works very closely with the hardware. Without going into too much detail, the BLE stack needs **real-time performance** to ensure that timing constraints from BLE are met and it relies on hardware **timers**, **RTC**, and **interrupts** to control the BLE radio. This strong coupling between the hardware and software, along with strong timing constraints, makes debugging a lot harder than usual.





Debugging - wires, wires and more wires

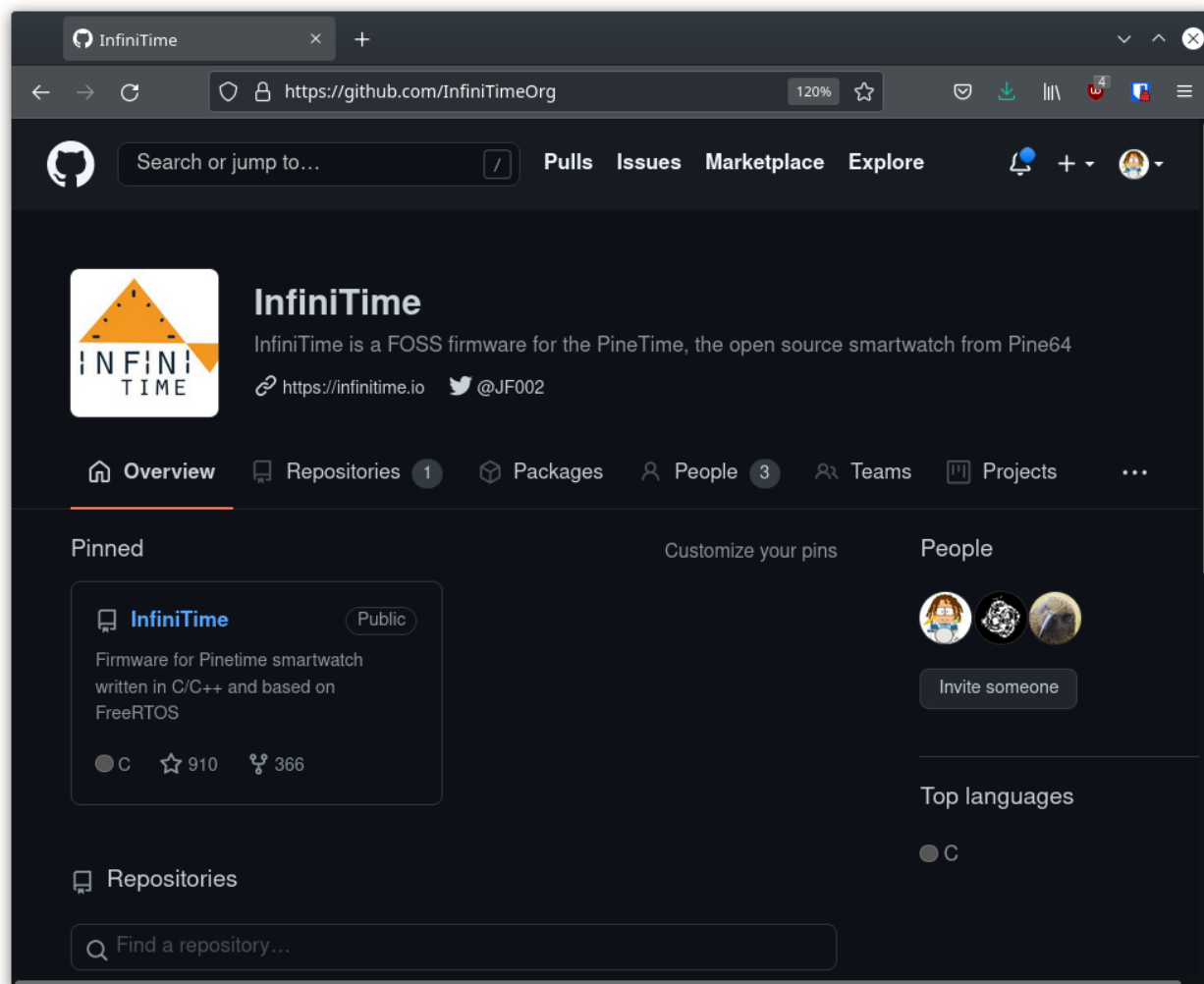
In the end, the [analysis and the fix](#) to the issue was provided by [Daniel Jackson](#). Daniel found that a bug in the BLE stack would cause an overflow in the advertising logic which would stop advertisement messages from being sent. This fix is small and easy, but finding that specific issue in the whole code was like finding a needle in a haystack. I can't help but be amazed by this mind-blowing collaboration between multiple developers from the open-source community. This was long-term work, it was tiring and frustrating, but we never gave up and finally understood what was going on and found a fix. Thanks to everyone who worked on this issue, and to all of you who supported us and tried to provide help and feedback - you're awesome!

Project organization

Since I am the person who started InfiniTime (the project was initially just called *my project* and later *PineTime-JF*) I am the only one in charge. I have all the powers, I take all the decisions and I do whatever I want with the project. I do, of course, encourage users to provide feedback and other developers to submit pull requests, but in the end I'm the one who manages all feedback and decides which modifications make it into the codebase of InfiniTime. But with great power comes great responsibility as they say... As the project gained popularity and momentum, I couldn't help but feel overwhelmed by the quantity of feedback, issues and pull requests. Responding to questions and issues reported by the users, reviewing and merging the nice pull requests, all take a lot of time and energy, and there isn't enough time in a day (after work and family) to manage the project in an efficient way.

Without knowing it, I became a [benevolent dictator](#) and, at the same time, the project's bottleneck. Many issues were left unanswered and it often took me several weeks to look at new pull requests. And that's fine, that's how open source development works! But that's not the most comfortable position for me to be in. I cannot afford to spend so much time working on InfiniTime. More importantly, I also don't want users and contributors to become frustrated by the relative slowness of the project.

I've been thinking about this situation for some time and recently decided to create a [Github organization](#). Organizations on Github allow inviting other contributors to the project and giving them specific access and permissions. The goal is to onboard a few contributors to the new *core team* so that I'm not the only one allowed to triage and manage issues as well as review and merge pull requests.



InfiniTime [Github Organization](#)

This is a major change in the project organization. We'll have to learn to work together in this new way, but I'm confident that together we'll manage to do great things. As of today I invited 2 contributors who I know from the project on Github and the chatrooms. I have known [Avamander](#) since the beginning of the PineTime project, and he has always given me a lot of useful advice on how to manage and organize InfiniTime. He also contributed nice features like the Music control app. [Tim aka geekbozu aka Geekboy1011](#) is very active on the chat rooms and worked a lot on the BLE issue we fixed in 1.6, and made many contributions to features like the persistent clock. Thanks to both of them for accepting the challenge and joining me in the new project structure. They are the first members of the new InfiniTime organization, but we'll probably end up inviting more members later on to help with

issue triage and PR reviews.

The new InfiniTime project URL is now <https://github.com/InfiniTimeOrg/InfiniTime>.

Github should take care of redirecting the old URLs to the new ones, but I recommend you update your bookmarks and *git remotes* to ensure you'll always point to the right URL

A word about donations

I'm always shy away from talking about money and donations in the context of working on open source projects. Open-source development, in my view, is all about collaboration, contribution, sharing and most importantly of all - fun. Money should not be an important part of it. However, at the beginning of this year many people started asking me how they could donate to thank me for my work on InfiniTime. At first I was a bit reluctant but finally created an account on Liberapay. I did so promising nothing in return (except that I'll probably invite my wife to the restaurant to make up for the time I spend working on the project) and yet, some people chose to anonymously donate money!

I received far more donations than I anticipated and I continue to receive very generous sums. I have to admit this is a nice additional source of motivation and I thank everyone who has donated so far. But this is not a one-man show project - InfiniTime and PineTime are successful thanks to the work of many people. This includes developers working on the firmware (InfiniTime and WaspOS, for example), companion apps (Amazfish, Gadgetbridge, Siglo, etc.), users and testers who provide feedback, as well as bloggers and content creators who write and talk about the project online. If you want to donate money to the PineTime project, feel free to have a look at profiles of these people and thank them for their amazing work!

Keep in mind that money is not the only way to motivate and encourage developers and contributors - provide feedback, share your experience and enthusiasm and thank them for their work. These little things will make any developer happy!

That's if for this the community update this month. Keep an eye out for the follow-up news blog post at the end of this month.

INFINITIME , NEWS , PINEPHONE , PINEPHONE-PRO , PINETIME

- [September update: Hurdles and Successes](#)
- [Introducing the PineNote](#)
- [April Update: New Developments](#)
- [January Update: Happy New Gear!](#)
- [December Update: The Longest One Yet](#)