

KDE Embedded

WHAT – WHY – NEXT

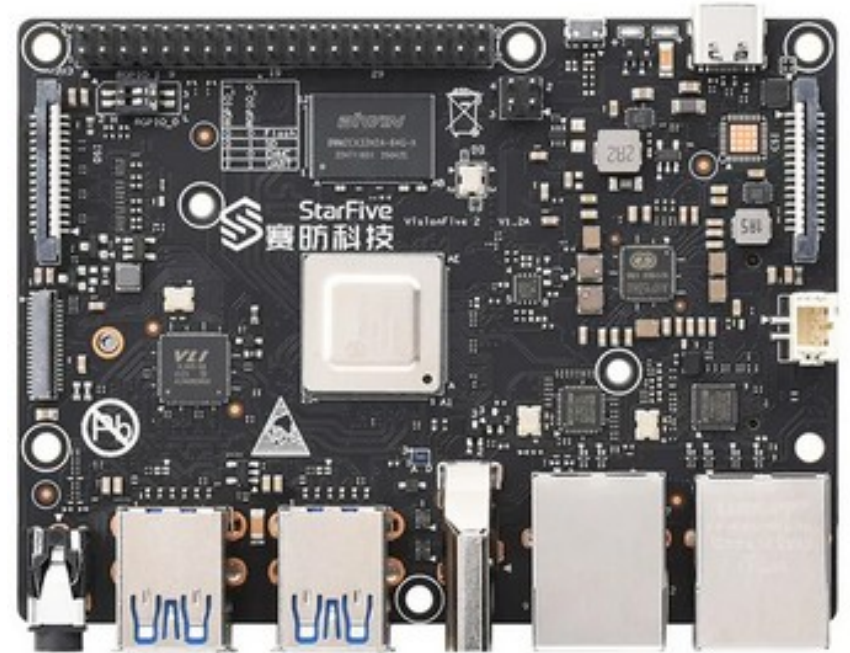
Andreas Cord-Landwehr



WHAT

WHAT DO I MEAN WITH “EMBEDDED”?

- Devices with a defined scope/task
- For this talk: always with a Linux Kernel
- ... with CPU, GPU, and all the nice hardware features we like :)



EMBEDDED:
WHY INVEST TIME?



curiosity / challenge / adventure

it's cool :)



The goals of KDE Embedded (so far) ...

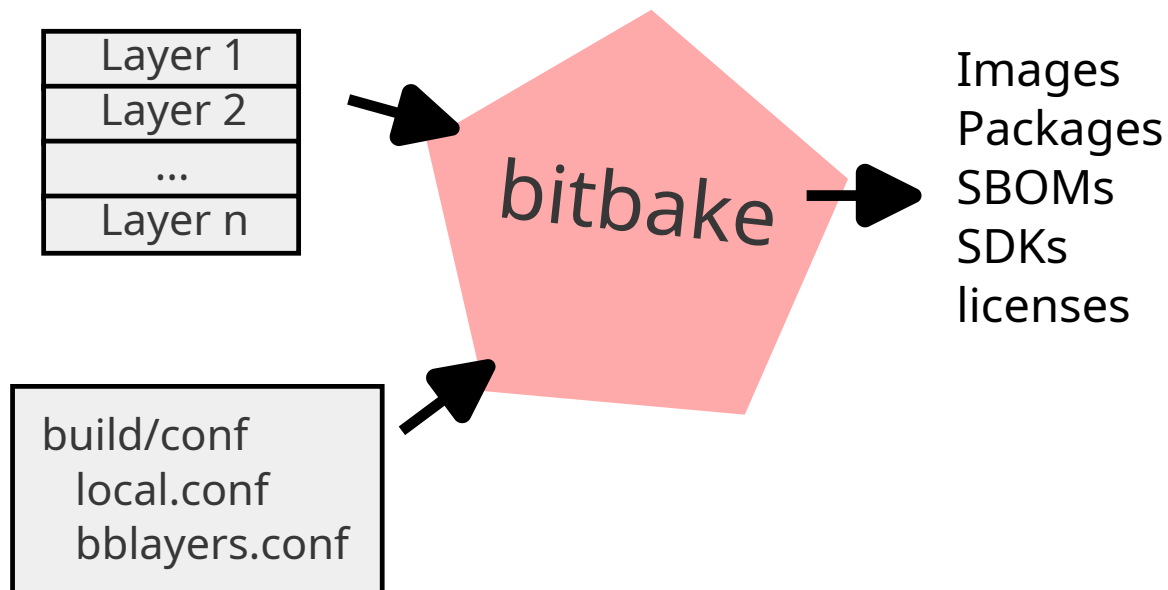
- allow easy integration of KF5/KF6 libraries into industry/automotive ecosystems**
- provide easy to use demo systems for KDE technology on embedded devices (Plasma Mobile, Plasma Bigscreen)**



OUR ECOSYSTEM FOR SYSTEM INTEGRATION

Creates customized distro images:

- industry standard
- allows easy port to new hardware
- biggest power: can rebuild everything
- provides all artifacts around an image



YOCTO PACKAGES

OUR YOCTO LAYERS

Packages that we keep updated to be used
by device creators:

- meta-kf5
- meta-kf6
- meta-kde
- meta-kde-demo



100%

KF5/KF6

All KF* libs are provided by
meta-kf5 / meta-kf6

100%

Plasma

All of Plasma, Plasma Mobile &
Plasma Bigscreen

6%

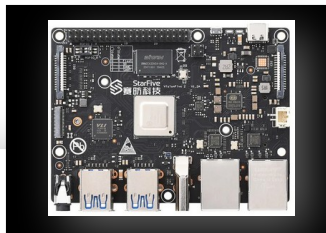
Apps

Currently we package only apps
as needed for demos



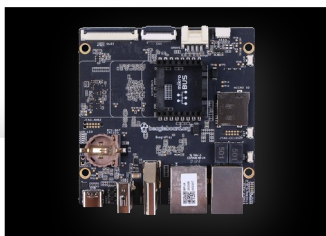
RaspberryPi4

The classic: ARM64 chip with MESA GL + VK driver



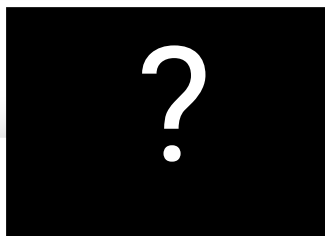
VisionFive2

First RISC-V board + on-board PowerVR GPU



Beagle(Play|V) ?

ARM & RISC-V boards with big community + PowerVR GPUs



...?

A lot is happening in the field

BOARDS

“It only works when it runs on the device”

This collection of boards is what people working on KDE embedded are typically using for smoke tests & demos

MILESTONES

1st meta-kf5 commit 2014

KDE Booth ELCE'19

First presentation of revived meta-kf5/-kde Yocto layers

Plasma @ RISC-V FOSDEM'19

First external contributions, which helped getting Plasma run on Western Digital RISC-V board with external GPU

Plasma @ VisionFive 2 FOSDEM'23

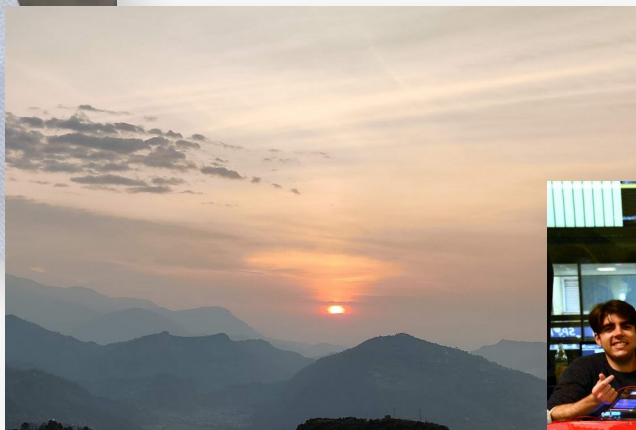
Massive community interest at the booth because we were the first with a running Plasma Wayland-Shell



CHALLENGES



Product



Hardware
Landscape



People

NEXT ~1-3
YEARS



meta-kf*

Our “best-seller”, used in industry products



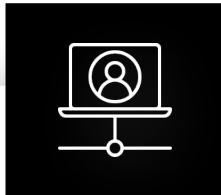
Demoing in fields we never went before

Raise interest for KDE in embedded community



Checkout new devices

Run Plasma & apps as a stack benchmark



Provide a “Pro-Mode” ?

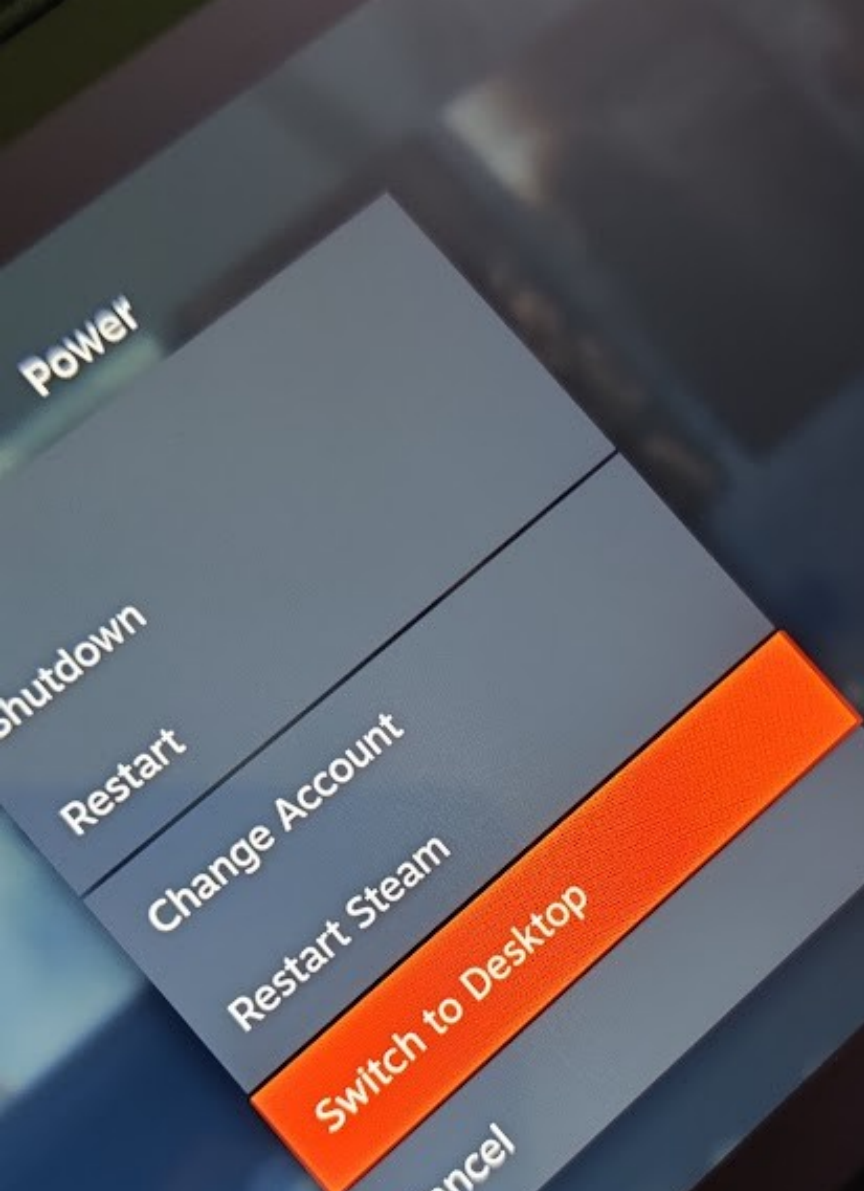
This is just an idea yet, but I think it makes sense...

PRODUCTS

MAKE SOMETHING THAT MAKES SENSE

Beyond the KF packages there is not much usage by industry; is that “okay & expected” or do we strive for more?

IDEAS?
→ **BoF**



PRODUCTS

I LOVE MY STEAMDECK

Steamdeck implements an awesome idea: Let the user easily switch from the main app of the system to a powerful management/ debugging/ benchmarking system.

→ this sounds like a general super helpful feature for many embedded devices

→ could this be a general feature for KDE embedded?

HARDWARE LANDSCAPE

RISC-V

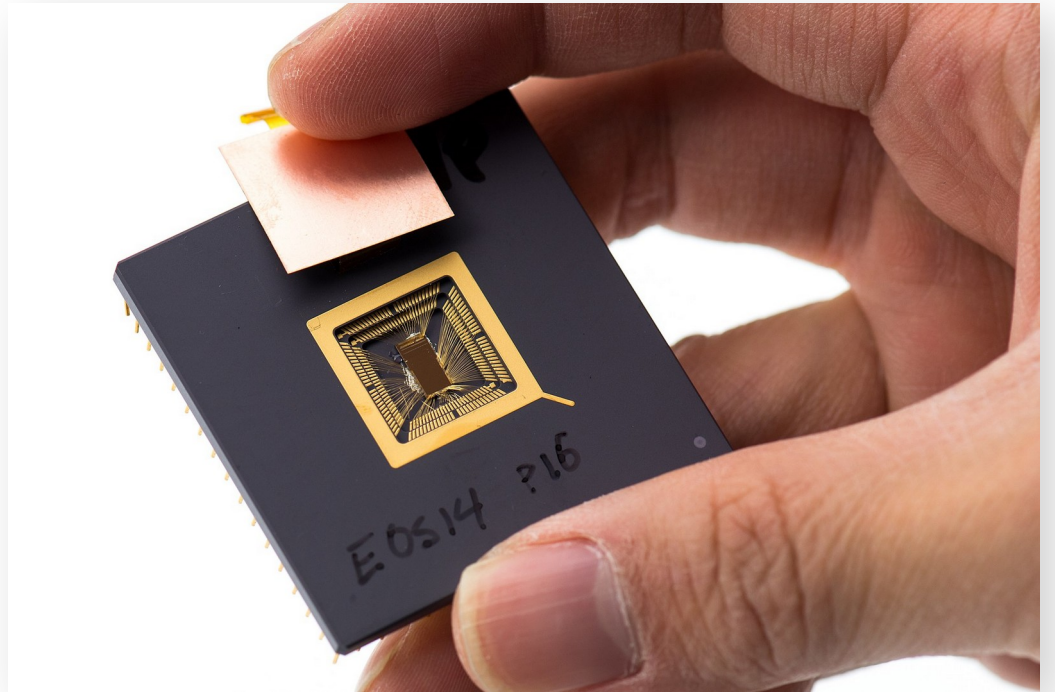
THAT topic for dev boards. Everyone has one in their pipeline, just this week Beagle Foundation announced a new one!

CPU wise closing to what we know from ARM/x86

GPUs

The main pain on getting Plasma running.

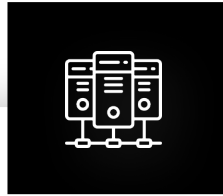
Especially on RISC-V only binary blob/non-MESA drivers available as of now... but improving





Read-only rootfs

Because of complexity & robustness



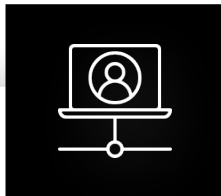
Standard-Tooling

Update mode, system init, build system... → switch to established FOSS



Security & Safety Awareness

few years ago embedded people were more relaxed



Multimedia Complexity

Pipewire & libcamera to integrate complexity

TRENDS

TOPICS ON EMBEDDED TRACKS

Needs of embedded devices are different than on “ordinary” desktops: They trade flexibility with robustness, security and fault-tolerance. But they still have desktop like problems in complexity.

PEOPLE

DON'T SCARE PEOPLE AWAY BY THE FIRST BUILD

Yocto is not known for being easily approachable,
but for its steep learning curve...

First Steps

Bootstrap Config

<https://community.kde.org/Yocto>

1. create build container
2. setup Yocto via repo & init script
3. build, wait, flash and your cross fingers during the first boot ;)

Collaboration

Matrix: [#yocto:kde.org](#)

New matrix channel exists since last
FOSDEM, come and join!

New People

We are few...

And need you! If you have
general interest or already use
our layers, come and join!



STARTER KIT

Getting Started

Get a supported device & follow the wiki

During this adventure you will experience fear, frustration and absolute joy when you flash your system to an SD card, put it into the device AND it actually starts and displays something on the screen!

Shopping List

SD-card : ~ 15 €

USB-to-RS232 Converter : ~ 5 €

SD-card writer : ~ 10 €

Dev Board : ~80-150 €



<https://community.kde.org/Yocto>

DEVELOP WORKFLOW

EMBEDDED DEVELOPMENT IS DIFFERENT

To speed-up development, we use desktop systems to develop & test natively and only WHEN it is ready, test and fix software on the devices.

Thus:

- host computer is the main work-horse
- new software is developed and tested exactly like any KDE app
- host computer provides cross-compile toolchain to build software for target
- and then (hopefully) we only have to fix the “minor” tweaks for making the software run proper



CONTINUOUS INTEGRATION

MAKE CONTRIBUTION SIMPLER

Transform Yocto Version Compatibility from Manual to Automatic

Big number of supported Yocto versions and manual fresh build
requires TBs of free space & CPU time

- need merge check builds for layers
- coverage for main release tracks



AND THEN?

SPREAD USAGE

Can we make the KDE meta layers more easy to use? How to attract more users? (or is every KF5/KF6 user in Yocto land already using them?)

NEW ADVENTURES

Come to the BoF or discuss your ideas during Akademy!





Homework

Tell me where KDE makes sense in
the (Linux-)embedded field



Add-On

Go and get you some HW and play
around with :)

Q/A?

