Yocto 101

Maciej Borzęcki
01.03.2017
CEHUG Łódź

- Working with embedded systems since 2006
- Since 2016 at RnDity
- Software Architect

In this talk

- Linux in embedded
- Building an image for your device
- Why do it with Yocto

Wide selection of boards

- Raspberry Pi 1/2/3/0
- BeagleBone White/Black/Green
- CHIP
- < $20
  - ~1GHz clock
  - 256-512MB of RAM
  - SD card or eMMC storage
    - NAND flash!!
  - Basic connectivity - Ethernet, WiFi
  - Maybe Bluetooth 4.0
- ~$50-60 - semi industrial products

Wide selection of boards

- Raspberry Pi 1/2/3/0
- BeagleBone White/Black/Green
- CHIP
- < $20
  - ~1GHz clock
  - 256-512MB of RAM
  - **SD card or eMMC storage**
    - **NAND flash!!**
  - Basic connectivity - Ethernet, WiFi
  - Maybe Bluetooth 4.0
- ~$50-60 - semi industrial products

You’ve bought xxxxx board, what now?

- HTPC - who hasn’t tried that?
- OpenELEC/Kodi
- ??

You’ve bought xxxxx board, what now?

- HTPC - who hasn’t tried that?
- OpenELEC/Kodi
- ??

Debian for embedded

- Lots of devices using Debian
  - (or now defunct Emdebian)
- Fully functional RsPI
- **Community support**

Debian for embedded contd.

- How to build SD card/flash image
  - debootstrap
- Reproducibility
  - Control variants
  - Package versions changing over time
- Mirror and freeze repositories
  - Breaking Debian user contract
- Still the most convenient home user solution
- deb cross compilation is quite recent
  - https://buildd.debian.org/
  - sbuild

Buildroot

- https://buildroot.org/
- Since 2005
- Kconfig/Makefile based build
- Used for a while in 2007, never came back
- **Would like to learn more**

Tizen

- Used a bit back in 2013
- Not impressed
- Great for enterprise with infra
- OBS
- Limited use outside of Samsung
- Comes with **mic**
  - MeeGo Image Creator?
  - Anaconda’s kickstart-like files
  - Possible to build SD image, HDD

Yocto & OpenEmbedded

- OE dates back to OpenZaurus ~2006
- All in one solution
- Reproducible builds
- Build contents tracking
- Source code delivery (license compliance)
- SDK
- Images
- Autobuilder
- Wide vendor support (YMMV)
    - Intel, Freescale, Texas Instruments

Taxonomy

- Yocto Project
  - Umbrella project
- Poky
  - Reference distribution
- Bitbake
  - Build tool
- OpenEmbedded
  - Build system
  - OE-core
  - meta-oe

Disclaimer

- Yocto user since 2014
- ~50 commits in main tree
- ~20 in other layers
- Multiple commits to 3rd party packages when porting

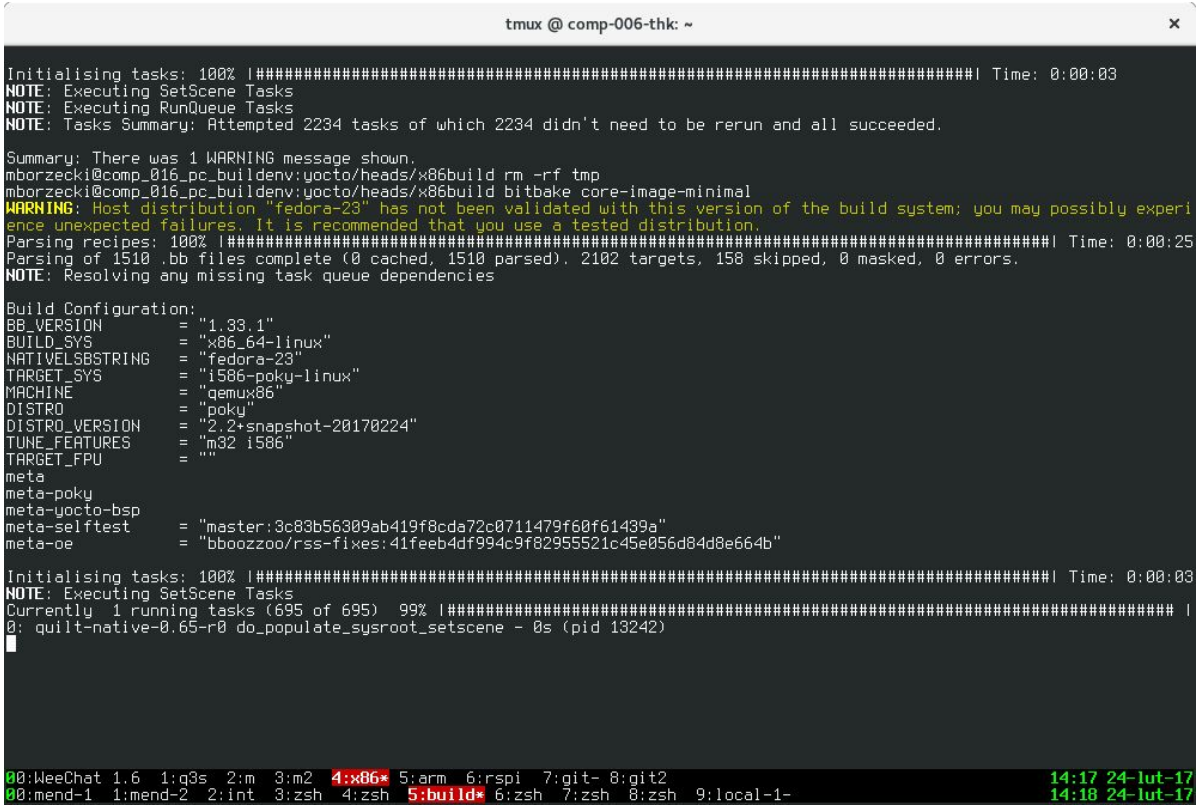
Yocto 101

1. Grab code (git://git.yoctoproject.org/poky)
2. Setup environment:
   ```
   $ source poky/oe-init-build-env qemu-x86
   ```
3. Build a reference image:
   ```
   $ bitbake core-image-minimal
   ```
4. Wait...

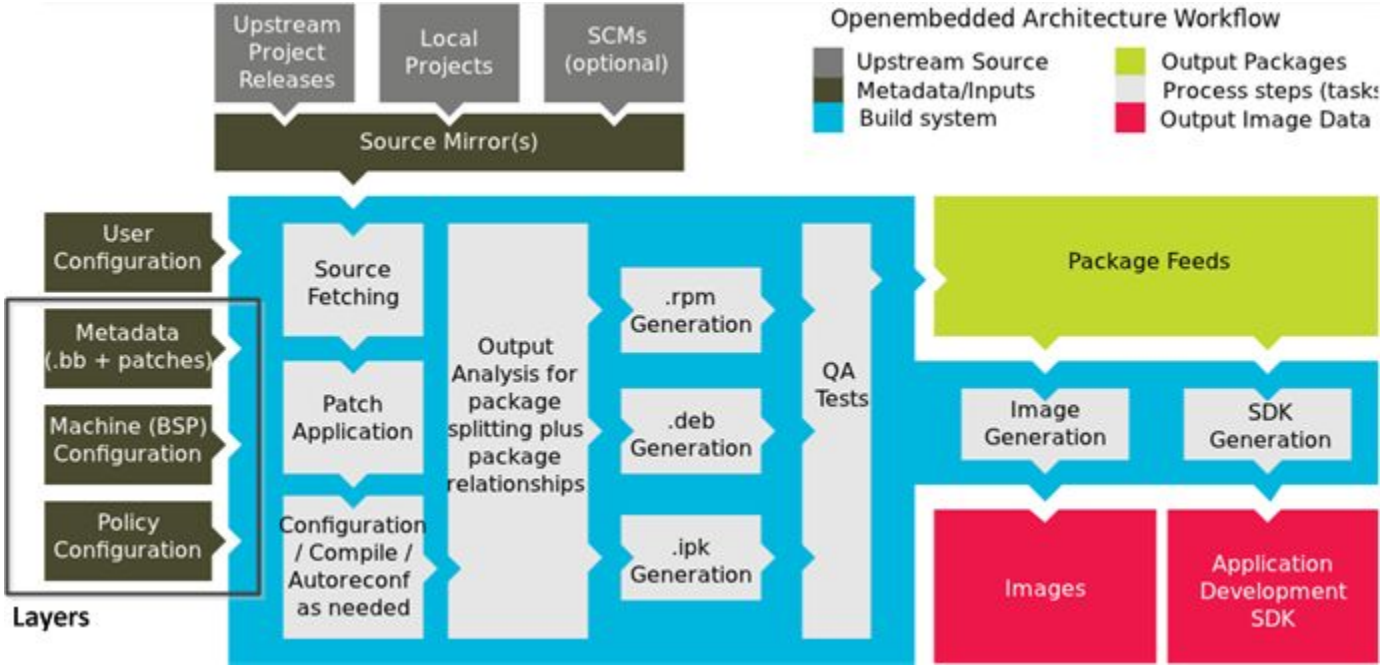
Yocto 101

1. Grab code (git://git.yoctoproject.org/poky)
2. Setup environment:
   ```
   $ source poky/oe-init-build-env qemu-x86
   ```
3. Build a reference image:
   ```
   $ bitbake core-image-minimal
   ```
4. Wait...
5. Wait some more...
6. Start QEMU with your image:
   ```
   $ runqemu qemux86 core-image-minimal nographic
   ```

Yocto 101 condt.

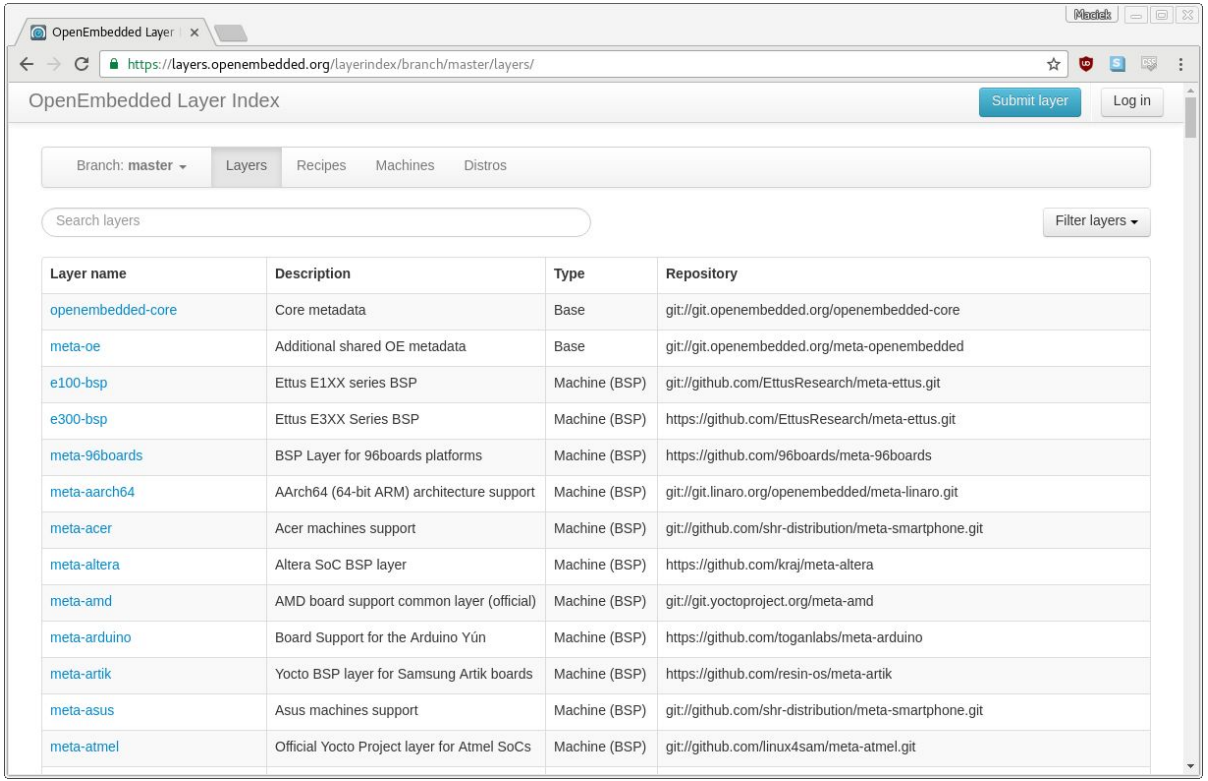


OpenEmbedded Architecture



Yocto Project Reference Manual - https://www.yoctoproject.org/docs/latest/ref-manual/ref-manual.html

Layer index


OpenEmbedded Layer
https://layers.openembedded.org/layerindex/branch/master/layers/
OpenEmbedded Layer Index
Submit layer
Log in
Branch: master
Layers
Recipes
Machines
Distros
Search layers
Filter layers
Layer name
Description
Type
Repository
openembedded-core
Core metadata
Base
git://git.openembedded.org/openembedded-core
meta-oe
Additional shared OE metadata
Base
git://git.openembedded.org/meta-openembedded
e100-bsp
Ettus E1XX series BSP
Machine (BSP)
git://github.com/EttusResearch/meta-ettus.git
e300-bsp
Ettus E3XX Series BSP
Machine (BSP)
https://github.com/EttusResearch/meta-ettus.git
meta-96boards
BSP Layer for 96boards platforms
Machine (BSP)
https://github.com/96boards/meta-96boards
meta-aarch64
AArch64 (64-bit ARM) architecture support
Machine (BSP)
git://git.linaro.org/openembedded/meta-linaro.git
meta-acer
Acer machines support
Machine (BSP)
git://github.com/shr-distribution/meta-smartphone.git
meta-altera
Altera SoC BSP layer
Machine (BSP)
https://github.com/kraj/meta-altera
meta-amd
AMD board support common layer (official)
Machine (BSP)
git://git.yoctoproject.org/meta-amd
meta-arduino
Board Support for the Arduino Yún
Machine (BSP)
https://github.com/toganlabs/meta-arduino
meta-artik
Yocto BSP layer for Samsung Artik boards
Machine (BSP)
https://github.com/resin-os/meta-artik
meta-asus
Asus machines support
Machine (BSP)
git://github.com/shr-distribution/meta-smartphone.git
meta-atmel
Official Yocto Project layer for Atmel SoCs
Machine (BSP)
git://github.com/linux4sam/meta-atmel.git

Yocto 101 condt.

- Cross-compilation toolchain
- Set of packages for host
  - *-native
- Set of packages for the target
  - one of RPM, DEB, IPK, tar.gz
- Root filesystem tree (core-image-minimal)
- Disk image
  - wic (*m* in *mic* upside down?)
  - DIY (rpi-sdimg in meta-raspberrypi, sdcard in meta-freescale)
- SDKs
  - cross toolchain - incl. headers
  - eSDK (extensible SDK) - possible to build whole images

Most important pieces

- Machine
  - Usually the target platform, examples: raspberrypi, beaglebone, genericx86-64
  - Platform description - CPU, flash, boot devices, serial console
  - Influences how packages are built and which ones may get installed
- Distro
  - Policy and features at the distribution level, examples: poky, poky-tiny
  - Influences how packages are built and which ones get installed
  - X11 vs. wayland, init system, alsa & PulseAudio, OpenGL ..
- Image
  - Predefined images: core-image-minimal
  - System image features, influences which packages get installed
  - debug-tweaks, dropbear vs. openssh

Most important pieces

```
conf/local.conf:
  DISTRO = “poky”
  MACHINE = “beaglebone”
```

- Machine
  - Usually the target platform, examples: raspberrypi, beaglebone, genericx86-64
  - Platform description - CPU, flash, boot devices, serial console
  - Influences how packages are built and which ones may get installed
- Distro
  - Policy and features at the distribution level, examples: poky, poky-tiny
  - Influences how packages are built and which ones get installed
  - X11 vs. wayland, init system, alsa & PulseAudio, OpenGL ..
- Image
  - Predefined images: core-image-minimal
  - System image features, influences which packages
  - debug-tweaks, dropbear vs. openssh

```
$ bitbake core-image-minimal
```

Code organization

- layers
  - recipes
    - poky/meta/recipes-core/busybox/busybox_1.24.1.bb
    - poky/meta/recipes-devtools/python/python3_3.5.2.bb
  - machines
    - poky/meta-yocto-bsp/conf/machine/beaglebone.conf
  - distributions
    - poky/meta-yocto/conf/distro/poky.conf
- upon layers (mix and match layers)
  - new recipes & extend existing ones
    - meta-virtualization/recipes-core/busybox/busybox_%.bbappend - extends busybox
  - even more machines
    - meta-ti/conf/machine/beaglebone.conf

Code organization

- layers
  - recipes
    - poky/meta/recipes-core/busybox/busybox_1.24.1.bb
    - poky/meta/recipes-devtools/python/python3_3.5.2.bb
  - machines
    - poky/meta-yocto-bsp/conf/machine/beaglebone.conf
  - distributions
    - poky/meta-yocto/conf/distro/poky.conf
- upon layers (mix and match layers)
  - new recipes & extend existing ones
    - meta-virtualization/recipes-core/busybox/busybox_%.bbappend - extends busybox
  - even more machines
    - meta-ti/conf/machine/beaglebone.conf

```
$ bitbake-layers show-layers
$ bitbake-layers show-recipes
```

Code organization

- layers
  - recipes
    - **poky/meta/recipes-core/busybox/busybox_1.24.1.bb**
    - poky/meta/recipes-devtools/python/python3_3.5.2.bb
  - machines
    - poky/meta-yocto-bsp/conf/machine/beaglebone.conf
  - distributions
    - poky/meta-yocto/conf/distro/poky.conf
- upon layers (mix and match layers)
  - new recipes & extend existing ones
    - **meta-virtualization/recipes-core/busybox/busybox_%.bbappend** - extends busybox
  - even more machines
    - meta-ti/conf/machine/beaglebone.conf

```
$ bitbake-layers show-appends
```

Code organization

- layers
  - recipes
    - poky/meta/recipes-core/busybox/busybox_1.24.1.bb
    - poky/meta/recipes-devtools/python/python_2.7.9.bb
  - machines
    - **poky/meta-yocto-bsp/conf/machine/beaglebone.conf**
  - distributions
    - poky/meta-yocto/conf/distro/poky.conf
- upon layers (mix and match layers)
  - new recipes & extend existing ones
    - meta-virtualization/recipes-core/busybox/busybox_%.bbappend - extends busybox
  - even more machines
    - **meta-ti/conf/machine/beaglebone.conf**

Directory structure

```
poky
├── bitbake
├── documentation
├── meta
├── meta-selftest
├── meta-skeleton
├── meta-yocto
├── meta-yocto-bsp
└── scripts
meta-ti
├── conf
├── licenses
├── recipes-bsp
├── recipes-connectivity
├── recipes-core
├── recipes-devtools
├── recipes-graphics
├── recipes-kernel
├── recipes-ti
└── scripts
```

```
meta-openembedded
├── contrib
├── meta-efl
├── meta-filesystems
├── meta-gnome
├── meta-gpe
├── meta-initramfs
├── meta-multimedia
├── meta-networking
├── meta-oe
├── meta-perl
├── meta-python
├── meta-ruby
├── meta-systemd
├── meta-webserver
└── meta-xfce
```

Recipe

```
SUMMARY = "Simple helloworld application"
SECTION = "examples"
LICENSE = "MIT"
LIC_FILES_CHKSUM = "file://${COMMON_LICENSE_DIR}/MIT;md5=0835ade698e0bcf8506ecda2f7b4f302"

SRC_URI = "file://helloworld.c"

S = "${WORKDIR}"

do_compile() {
     ${CC} helloworld.c -o helloworld
}

do_install() {
     install -d ${D}${bindir}
     install -m 0755 helloworld ${D}${bindir}
}
```

Recipe (meta/recipes-extended/bc/bc_1.06.bb)

```
SUMMARY = "Arbitrary precision calculator language"
HOMEPAGE = "http://www.gnu.org/software/bc/bc.html"

LICENSE = "GPLv2+ & LGPLv2.1"
LIC_FILES_CHKSUM = "file://COPYING;md5=94d55d512a9ba36caa9b7df079bae19f \
                    file://COPYING.LIB;md5=d8045f3b8f929c1cb29a1e3fd737b499 \
                    file://bc/bcdefs.h;endline=31;md5=46dffdaf10a99728dd8ce358e45d46d8 \
                    file://dc/dc.h;endline=25;md5=2f9c558cdd80e31b4d904e48c2374328 \
                    file://lib/number.c;endline=31;md5=99434a0898abca7784acfd36b8191199"

SECTION = "base"
DEPENDS = "flex"
PR = "r3"

SRC_URI = "${GNU_MIRROR}/bc/bc-${PV}.tar.gz \
           file://fix-segment-fault.patch "

SRC_URI[md5sum] = "d44b5dddebd8a7a7309aea6c36fda117"
SRC_URI[sha256sum] = "4ef6d9f17c3c0d92d8798e35666175ecd3d8efac4009d6457b5c99cea72c0e33"

inherit autotools texinfo update-alternatives

ALTERNATIVE_${PN} = "dc"
ALTERNATIVE_PRIORITY = "100"

BBCLASSEXTEND = "native"
```

Recipe (meta/recipes-extended/bc/bc_1.06.bb)

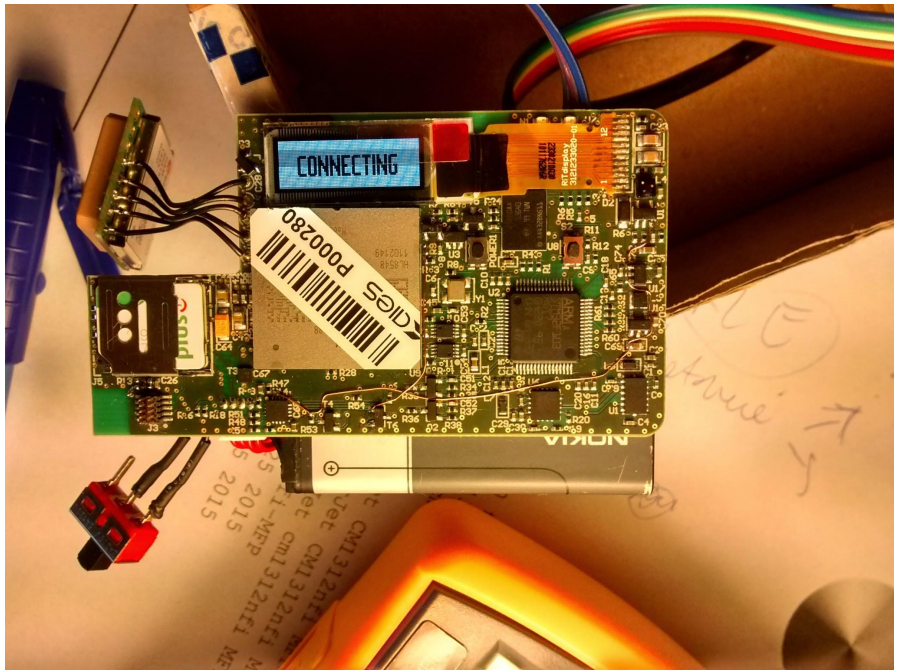
```
SUMMARY = "Arbitrary precision calculator language"
HOMEPAGE = "http://www.gnu.org/software/bc/bc.html"

LICENSE = "GPLv2+ & LGPLv2.1"
LIC_FILES_CHKSUM = "file://COPYING;md5=94d55d512a9ba36caa9b7df079bae19f \
                    file://COPYING.LIB;md5=d8045f3b8f929c1cb29a1e3fd737b499 \
                    file://bc/bcdefs.h;endline=31;md5=46dffdaf10a99728dd8ce358e45d46d8 \
                    file://dc/dc.h;endline=25;md5=2f9c558cdd80e31b4d904e48c2374328 \
                    file://lib/number.c;endline=31;md5=99434a0898abca7784acfd36b8191199"

SECTION = "base"
DEPENDS = "flex"
PR = "r3"

SRC_URI = "${GNU_MIRROR}/bc/bc-${PV}.tar.gz \
           file://fix-segment-fault.patch "

SRC_URI[md5sum] = "d44b5dddebd8a7d7309aea6c36fda117"
SRC_URI[sha256sum] = "4ef6d9f17c3c0d92d8798e35666175ecd3d8efac4009d6457b5c99cea72c0e33"

inherit autotools texinfo update-alternatives

ALTERNATIVE_${PN} = "dc"
ALTERNATIVE_PRIORITY = "100"

BBCLASSEXTEND = "native"
```

autotools
cmake
meson*
pip/distutils
cpan
npm

Some devices

Workout gadget

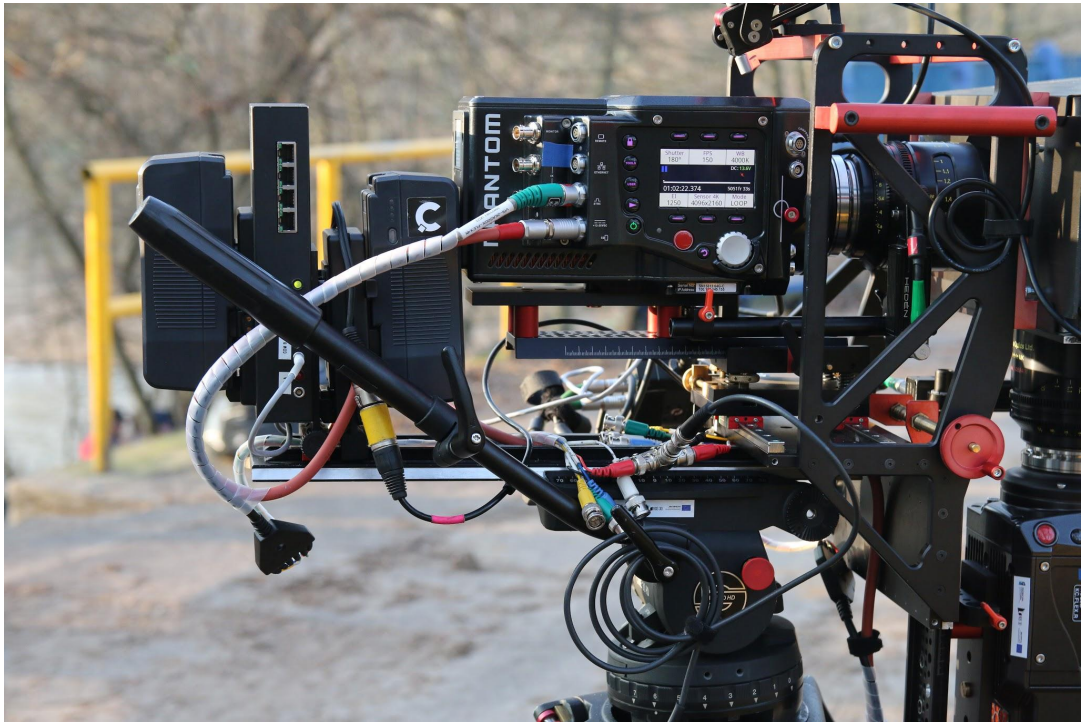
- Accelerometers
- Gyroscopes
- Heartbeat
- Position (10Hz)
- On board storage
- 3G connectivity
- STM32F1x



Data hub

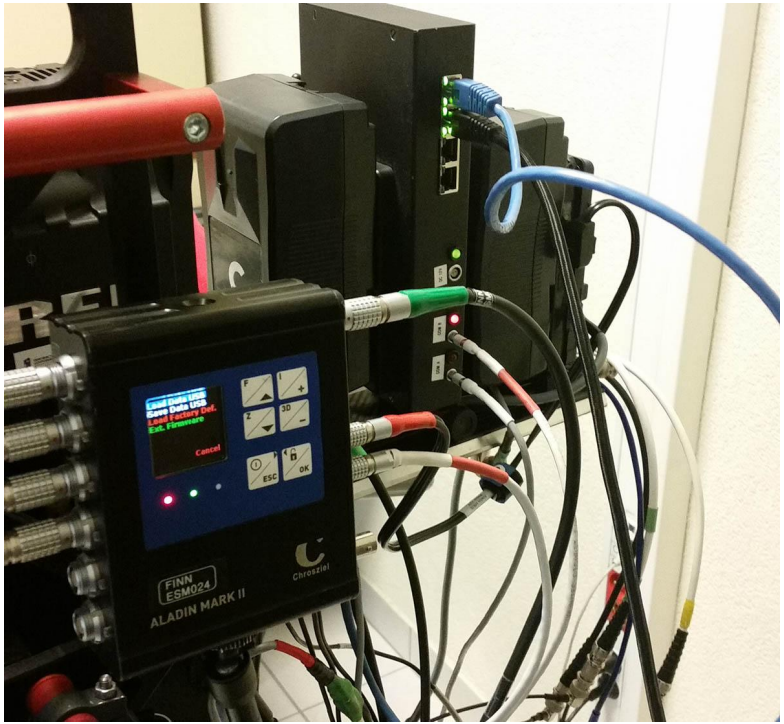
- Storage
- Recharge station
- Firmware update

Rig Controller



Rig Controller - Mobile App

Rig Controller



Demo

Q&A