Call your NetBSD

BSDCan 2013
Ottawa, Canada

Pierre Pronchery (khorben@netbsd.org)
May 17th 2013

Let's get this over with

- Pierre Pronchery
- French, based in Berlin, Germany
- Freelance IT-Security Consultant
- OSDev hobbyist
- NetBSD developer since May 2012 (khorben@)

Agenda

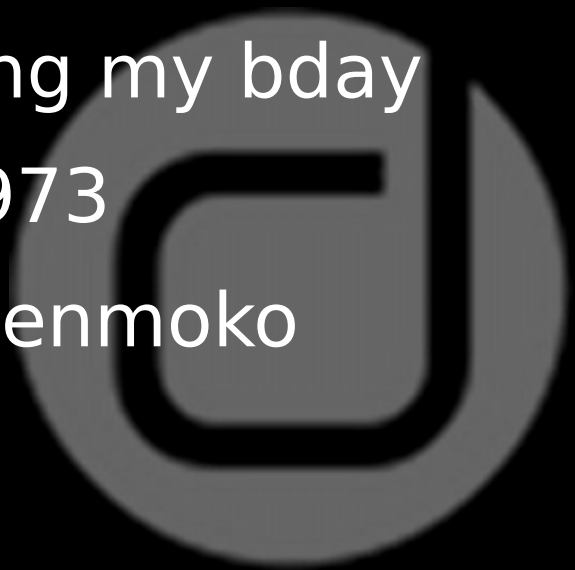
1. Why am I doing this?
2. Target hardware: Nokia N900
3. A bit of ARM architecture
4. NetBSD on ARM
5. Challenges of the port
6. Current status
7. DeforaOS embedded desktop
8. Future plans

1. A long chain of events

- $friend0 gives me Linux CD
- Computer not happy with Linux
- Get FreeBSD CD shipped
- Stick with Linux for a while
- Play with OpenBSD on Soekris hardware
- $friend1 gets Zaurus PDA
- Switch desktop and laptop to NetBSD
- I buy a Zaurus PDA
- I try OpenBSD on Zaurus PDA

1. Chain of events, continued

- $gf gets invited to $barcamp
- I play with my Zaurus during her presentation
- $barcamp_attender sees me doing this
- Begin to work on the DeforaOS desktop
- Get some of it to run on the Zaurus
- Attend CCC Camp near Berlin during my bday
- $gf offers me an Openmoko Neo1973
- Adapt the DeforaOS desktop to Openmoko



1. Chain of events, unchained

- $barcamp_attender was at the CCC Camp, too
- We begin to sell the Openmoko Freerunner
- Create a Linux distribution to support it
- Openmoko is EOL'd and we split ways
- $friend2 gives me sparc64 boxes
- Get more involved with NetBSD
- Nokia gives me a N900 during a developer event
- $barcamp_attender points me to a contest
- Contest is about creating an OSS tablet

1. Chain of events (out of breath)

- Run DeforaOS on NetBSD on the WeTab tablet
- Co-win the contest this way
- $friend3 boots NetBSD on Nokia N900
- Give a talk about the WeTab tablet
- Promise to work on the Nokia N900 next thing
- Apply to BSDCan 2013
- Taste maple syrup for the first time in Canada
- Here I am in front of you

Pictures: Sharp Zaurus

Pictures: Openmoko Freerunner

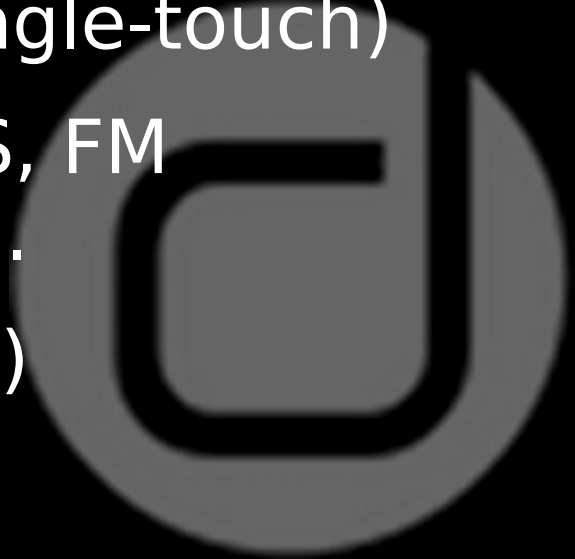
Pictures: WeTab

Pictures: DeforaOS

2. Nokia N900 Mobile Internet Device

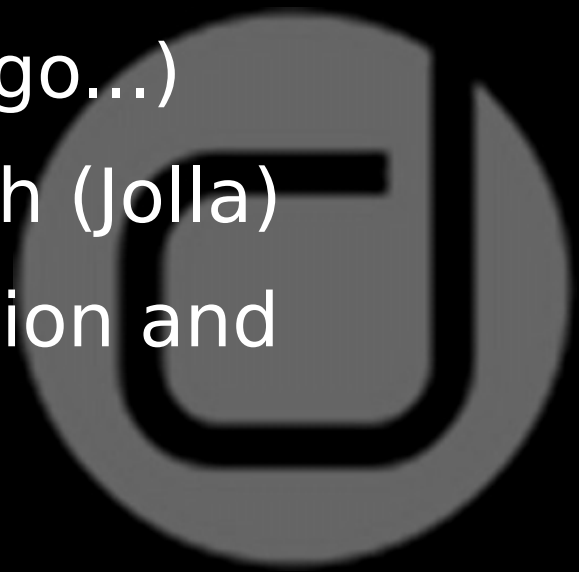
2. Nokia N900

- First of the Nokia N series tablets (N770, N800, N810) to get a GSM chip, available late 2009
- Texas Instruments OMAP3 core
- 600MHz ARM Cortex-A8, 256MB RAM
- 256MB NAND flash, 32GB eMMC flash
- 800x480 resistive touchscreen (single-touch)
- 3G GSM, 802.11bg, Bluetooth, GPS, FM receiver and transmitter, Infrared...
- Shipped with native Linux (Maemo)

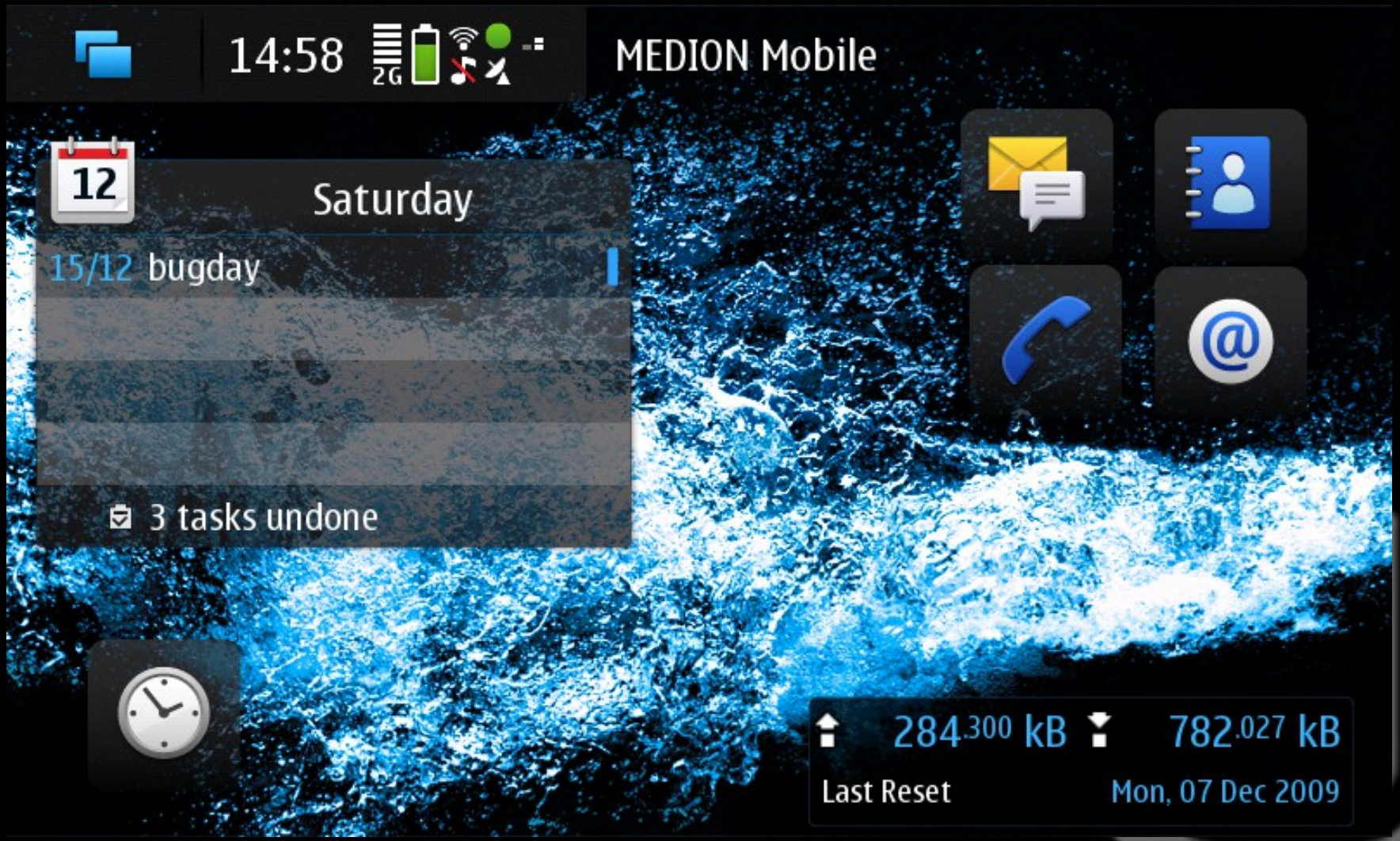


2. Nokia N900: Maemo

- Originally based on Debian GNU/Linux
- Well integrated hybrid tablet/telephony
- Switched from Gtk+ to Qt with the N900
- Most components Open Source
- Good community around the project
- Debatable political decisions (Meego...)
- Follow-up company about to launch (Jolla)
- But for us now: lots of documentation and working code

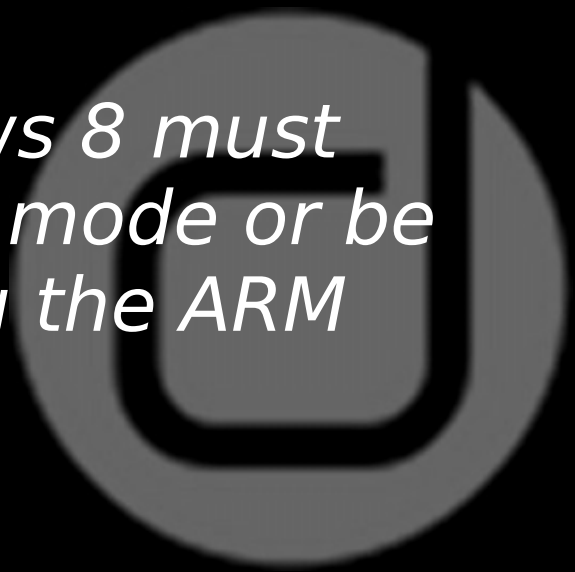


2. Nokia N900: Maemo screenshot



3. ARM architecture crash-course

- Looks a lot like a regular PC: CPU, DDR RAM, storage, internal buses, user input, interfaces, connectors, graphics acceleration...
- Additional constraints: mobile, autonomous (battery), power management, physical size...
- Nothing that prevents running *BSD on it, except for SecureBoot
  « [...] systems certified for Windows 8 must allow secure boot to enter custom mode or be disabled, but not on systems using the ARM architecture » (Wikipedia)



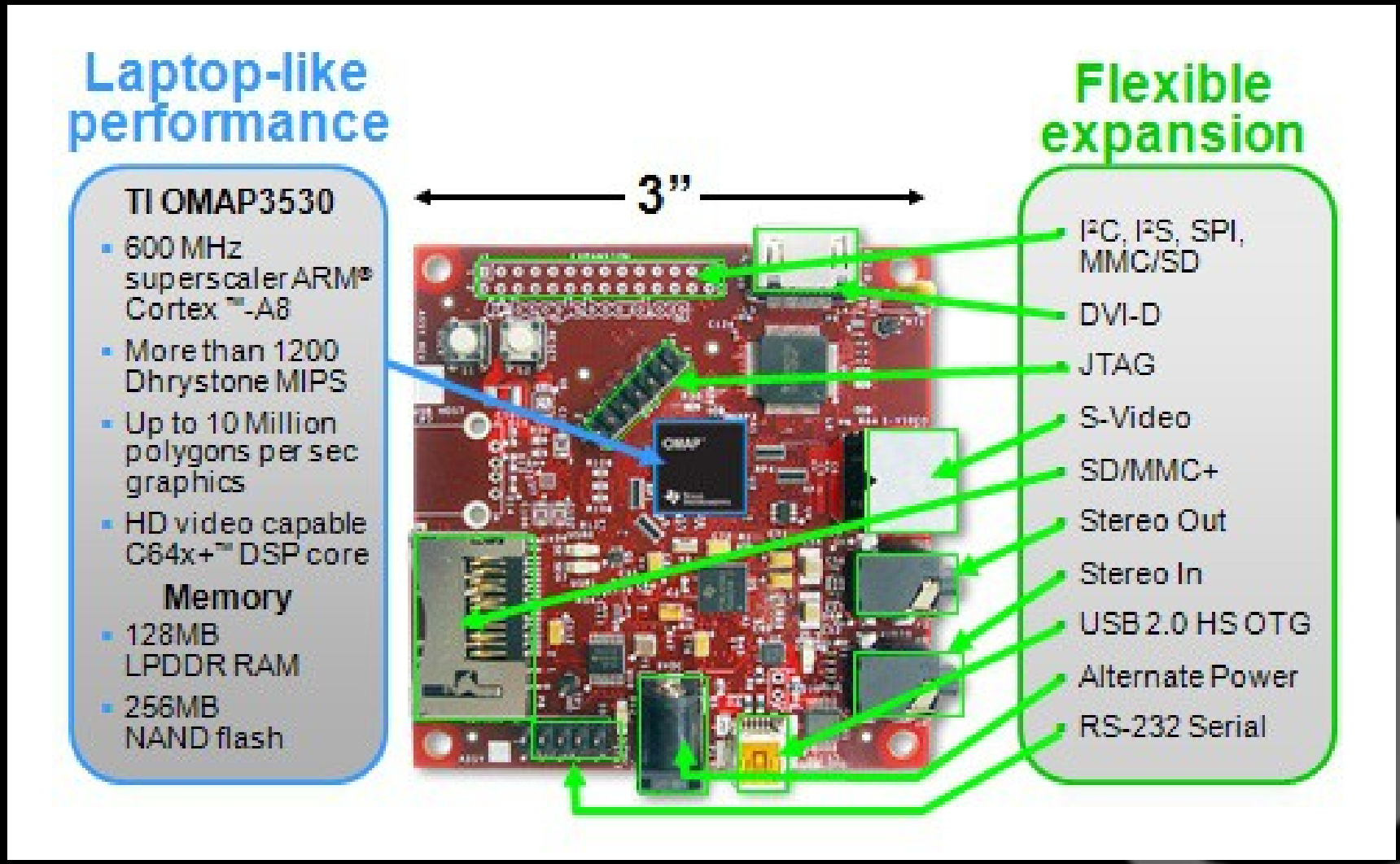
3. ARM architecture 101

- RISC-based, 32 or 64 bits, little or big endian
- ARM (the company) licenses designs to manufacturers
- Different generations, instruction sets and extensions
- As many ARM platforms as there are combinations of manufacturers and instruction sets

3. ARM architecture: TI OMAP (1/2)

- Texas Instruments own platform
- General-purpose ARM core
- Typically lots of different co-processors and extension chips
- Popularized in OSS world by the BeagleBoard (OMAP 3530, 3rd generation)
- Nokia N900 runs on OMAP 3430 (same SOC, different packaging)

3. ARM architecture: BeagleBoard



3. ARM architecture TI OMAP (2/2)

- Main CPU, internal interrupt lines
- Exposes a number of buses: I2C, SPI, GPIO...
- GPIO pins optionally used as interrupts
- Companion processor on some bus, interrupts within the main CPU
- Potentially lots of additional processors and chips on their respective buses, possibly interrupting via GPIO pins

4. NetBSD/evbarm

4. About NetBSD/evbarm

« NetBSD/evbarm is the port of NetBSD to various evaluation and prototyping boards based on CPUs implementing the ARM architecture. NetBSD/evbarm also supports some specific embedded system products based on prototype board designs.

Jason Thorpe is the maintainer of NetBSD/evbarm. »

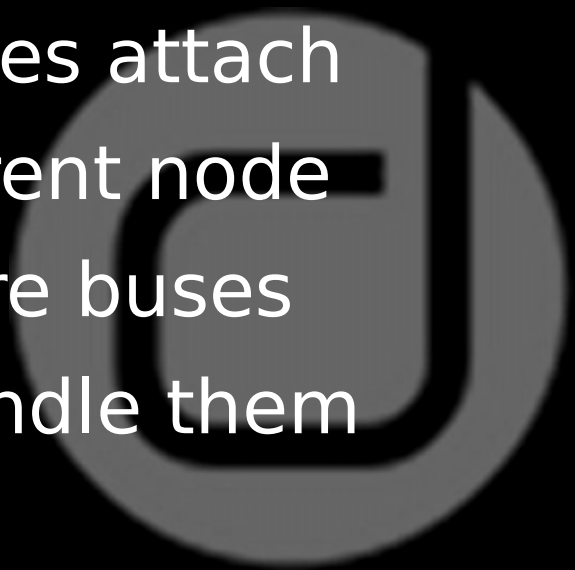
http://www.netbsd.org/ports/evbarm/

4. NetBSD/evbarm: quick glance

```
$ ls -1 src/sys/arch/evbarm/conf | grep
'^[A-Z0-9]\+$'

ARMADILLO210   ARMADILLO9 BCM5301X
BEAGLEBOARD  BEAGLEBOARDXM  BEAGLEBONE
CP3100

DEVKIT8000 DNS323 GEMINI GOLDENGATE GUMSTIX
  HPT5325  IGEPV2 IMX31LITE

INSTALL  INTEGRATOR IQ31244  IQ80310
IQ80321  IXDP425

IXM1200  LUBBOCK  MINI2440 MV2120

N900 NAPPI  NETWALKER  NSLU2

OPENRD OSK5912  OVERO  PANDABOARD PBI
```

4. NetBSD: portable so clean

- Clean so portable
- Machine-independent code whenever likely:
  - Bus access abstractions
  - Same APIs across supported ports
- Device tree model:
  - Virtual root device on which buses attach
  - Each device attaches to one parent node
  - Devices may in turn expose more buses
- Devices register interrupts and handle them



4. NetBSD reminder: interrupts

- Interrupts are expensive: you can only handle so many
- Interrupts occur all the time: there is always something to do
- Therefore:
  - They must be handled fast,
  - with as little code as possible,
  - ...and it is forbidden to sleep: resulting operations must be deferred (workqueues...)

4. NetBSD: can you see it coming?

- Dozens of drivers to write
- Attaching on multiple buses at the same time
- With interrupts lines on separate processors
- What can possibly go wrong...
  - Bad news is: it's pretty bad
  - Good news: it's not so bad

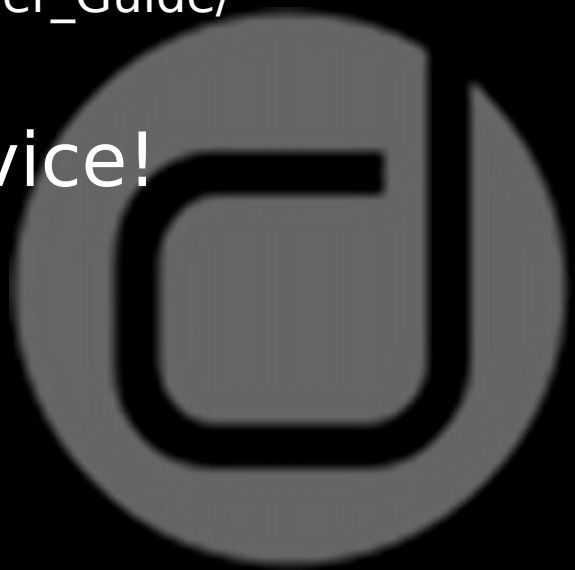
5. A challenging port

5. Challenge -I: preparation

- Make sure the Nokia N900 can boot from the MicroSD card:
  - Developer mode (wiki page unreachable?)
  - Flashing u-boot? (done this long ago)
  - See maybe:

http://wiki.maemo.org/Documentation/Maemo_5_Developer_Guide/Development_Environment/Maemo_Flasher-3.5

- Be careful and don't brick your device!



5. Challenge -I: MicroSD card

```
# fdisk sd0
Partition table:
0: NetBSD (sysid 169)
    start 1, size 7372799 (3600 MB, Cyls 0-458/238/36)
        PBR is not bootable: All bytes are identical (0x00)
1: <UNUSED>
2: Primary DOS with 32 bit FAT - LBA (sysid 12)
    start 7389184, size 65536 (32 MB, Cyls 459/243/41-464/8/56)
3: <UNUSED>
No active partition.
# disklabel sd0
[skipped]
# newfs sd0a
# mount /dev/sd0a /mnt
# cd /mnt && for i in /path/to/sets/*.tgz; do \
   progress -f $i tar xzpf -; done
# (cd dev && ./MAKEDEV all)
```

5. Challenge -I: MicroSD card

```
# cat > etc/fstab << EOF
# NetBSD /etc/fstab
/dev/ld0a  /   ffs rw  1 1
tmpfs      /tmp    tmpfs   rw,-m=1777,-s=16m,noexec
ptyfs      /dev/pts   ptyfs   rw
EOF
# cat >> etc/rc.conf << EOF
rc_configured=YES

clear_tmp=NO
dmesg=NO
hostname=n900.defora.rom
inetd=NO
newsyslog=YES
no_swap=YES
powerd=YES
quota=NO
savecore=NO
virecover=NO
wscons=YES
xdm=YES
EOF
```

5. Challenge 0: patience

- Tedious test procedure:
  - Update MicroSD card
  - Stick in the phone and (re)boot
  - Try, fail and repeat
- No network boot, no serial line
- Reminder: initially I had no keyboard support
- OMAP datasheet 3487 pages
- Companion chip datasheet 947 pages
- One change at a time, do not stick on one bug

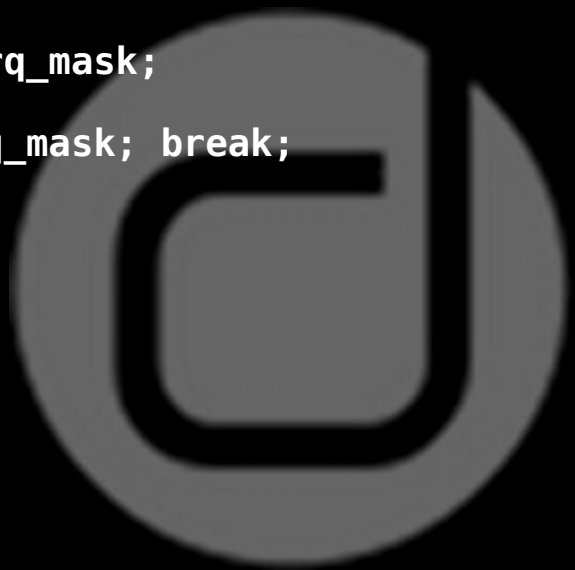
5. Challenge I: differentiate with BEAGLEBOARD

- There was already some support for the BeagleBoard
- This support assumed a console on the serial line (none easily accessible on the N900)
- Likewise it assumed no screen was connected (our only way to debug and interact so far)
- Introduced kernel option OMAP_3430 (not great but not horrible either)

5. Challenge II: GPIO sensors

- Many sensors use the same GPIO pin for interrupting and for data
- Meaning, they are “edge-triggered” on both edges (events are detected on a state transition, on both low-to-high and high-to-low)
- ...omapgpio(4) did not implement them

```
+       case IST_EDGE_BOTH: gpio->gpio_edge_falling_mask |= irq_mask;
+                           gpio->gpio_edge_rising_mask |= irq_mask; break;
```



5. Challenge III: hooking interrupts

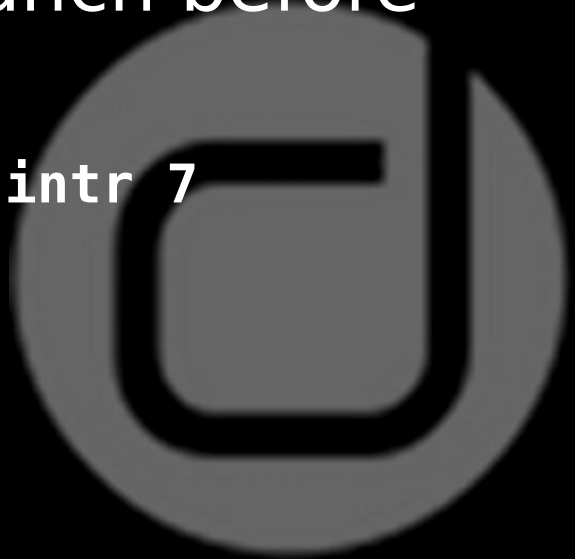
- Now it gets interesting
- On OMAP2 GPIO interrupts are mapped above system interrupts! (using the ARM PIC code)
- Interrupts on this platform are identified as integer values
- In this example pin 0 of GPIO bus 0 is mapped as interrupt 96:

```
omapgpio0 at obio1 addr 0x48310000 size 0x0400 intrbase 96 intr 29
```

5. Challenge III: hooking interrupts

- Considering this, I2C and SPI devices should be able to locate their respective interrupt easily
- In practice it breaks portability, as intr_establish() is a Machine-Dependent call
- ...so the “intr” locators do not exist (rightly so)
- Working in the “khorben-n900” branch before looking for a portable solution:

```
tps65950pm1 at iic0 addr 0x49 intrbase 288 intr 7
```



5. Challenge IV: keypad support

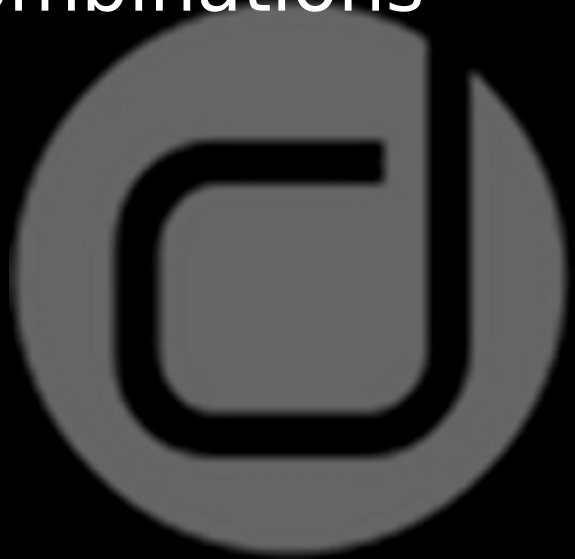
- Keypad is on companion chip
- Companion chip is on the I2C bus
- Companion chip interrupts on the main CPU
- Talking on the I2C bus requires sleeping
- Deferred handling does not prevent interrupts
- The device keeps interrupting so the handler is never called
- Disabling interrupts once registered was not implemented

5. Challenge IV: keypad support

- The companion chip is connected at up to five different addresses, on up to two I2C buses
- The keypad registers are ***not*** reachable from the instance interrupting (different address)
  - tps65950_intr() defers $handler1, returns 0
  - $handler1 uses global $handler3 to wake the keypad, schedules $handler2 at lower priority
  - $handler3 reports the key pressed
  - $handler2 re-enables the interrupt
- I had to patch the ARM PIC code for this

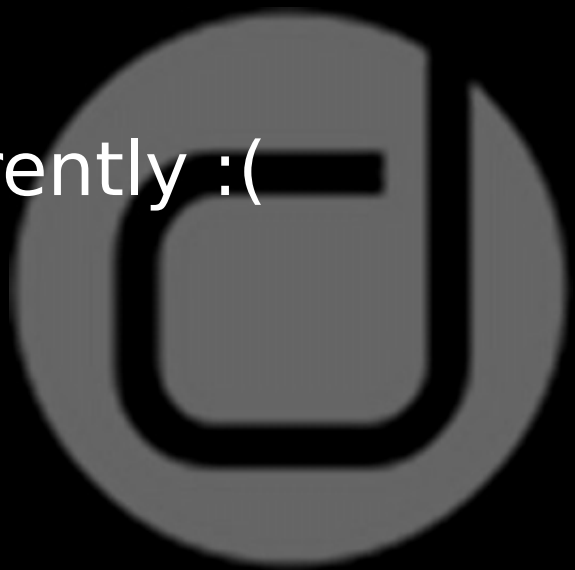
5. Challenge V: keypad support

- We have no speed locators
- ...meaning we typically assume the lowest clock speed available on a given bus
- Nice for stability but I need to wait one second between each key pressed :(
- (and I haven't implemented key combinations yet by the way)



5. Challenges: minor issues

- cngetc() is a busy loop IIRC (sucks CPU and therefore battery life)
- The battery slowly drains out (no support for charging yet)
- Some calls (like sysmon_pswitch_event()) are not documented as requiring execution outside of interrupt context
- I need a crash course on SPI apparently :(



6. Current status: the code

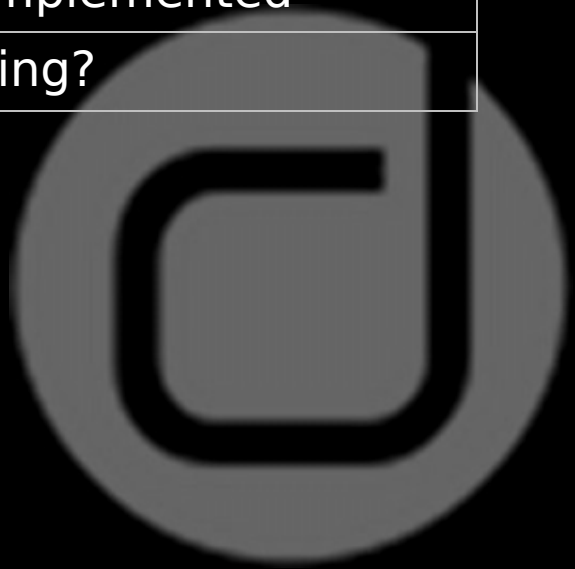
- Getting as much as possible into HEAD
- Created the “khorben-n900” branch on May 7
- Also using Git for “convenience”, hosted on Github at https://github.com/khorben/khorben-n900

6. Current status: OMAP 3430

Device component	Driver	Status
DMA	omapdma(4)	Working?
Framebuffer	omapfb(4)	Working
GPIO	omapgpio(4)	Working (interrupts)
I2C	omapiic(4)	Working (polling)
Interrupt controller (ICU)	omapicu(4)	Working
NAND	omapnan(4)	Working?
SDHC	sdhc(4)	Working
SPI	omapspi(4)	In progress (interrupts)
System Control Module	omapscm(4)	Working?
System timers	omapmputmr(4)	Working?
Watchdog timers	omapwdt32k(4)	Working?

6. Current status: TPS65950 (companion chip)

Device component	Driver	Status
Audio	tps65950pm(4)	Not implemented
GPIO	tps65950pm(4)	Unknown
Keypad	tps65950pm(4)	Working (basic)
LEDs	tps65950pm(4)	Working?
Power management	tps65950pm(4)	In progress
Real Time Clock	tps65950pm(4)	Working?
USB	tps65950pm(4)	Not implemented
Watchdog	tps65950pm(4)	Working?



6. Current status: sensors (GPIO)

Sensor	Driver	Status
Power supply	n900acad(4)	Reports AC adapter events (not consistent)
Audio jack insertion	n900audjck(4)	Reports a hotkey (working)
Camera button	n900cambtn(4)	Reports hotkeys for focus and capture (working)
Camera cover	n900camcvr(4)	Reports hotkey (working)
Keypad slide	n900kbdsld(4)	Reports hotkey (working)
Lock button	n900lckbtn(4)	Reports hotkey (working, spurious events)
Proximity sensor (front)	n900prxmty(4)	Reports hotkey (working)

6. Current status: I2C devices

Device	Driver	Status
Accelerometer	stmemsis(4)	In progress (mostly works)
Audio codec	?	Not implemented
Battery monitoring	?	Not implemented
Headphone amplifier	?	Not implemented
LED & vibrator driver	lp5523led(4)	In progress
Light sensor	?	Not implemented

6. Current status: SPI devices

Device	**Driver**	**Status**
Panel controller	?	Not implemented
Touchscreen	tsc2005ts(4)	In progress
Wireless interface	wilink(4)	Not implemented

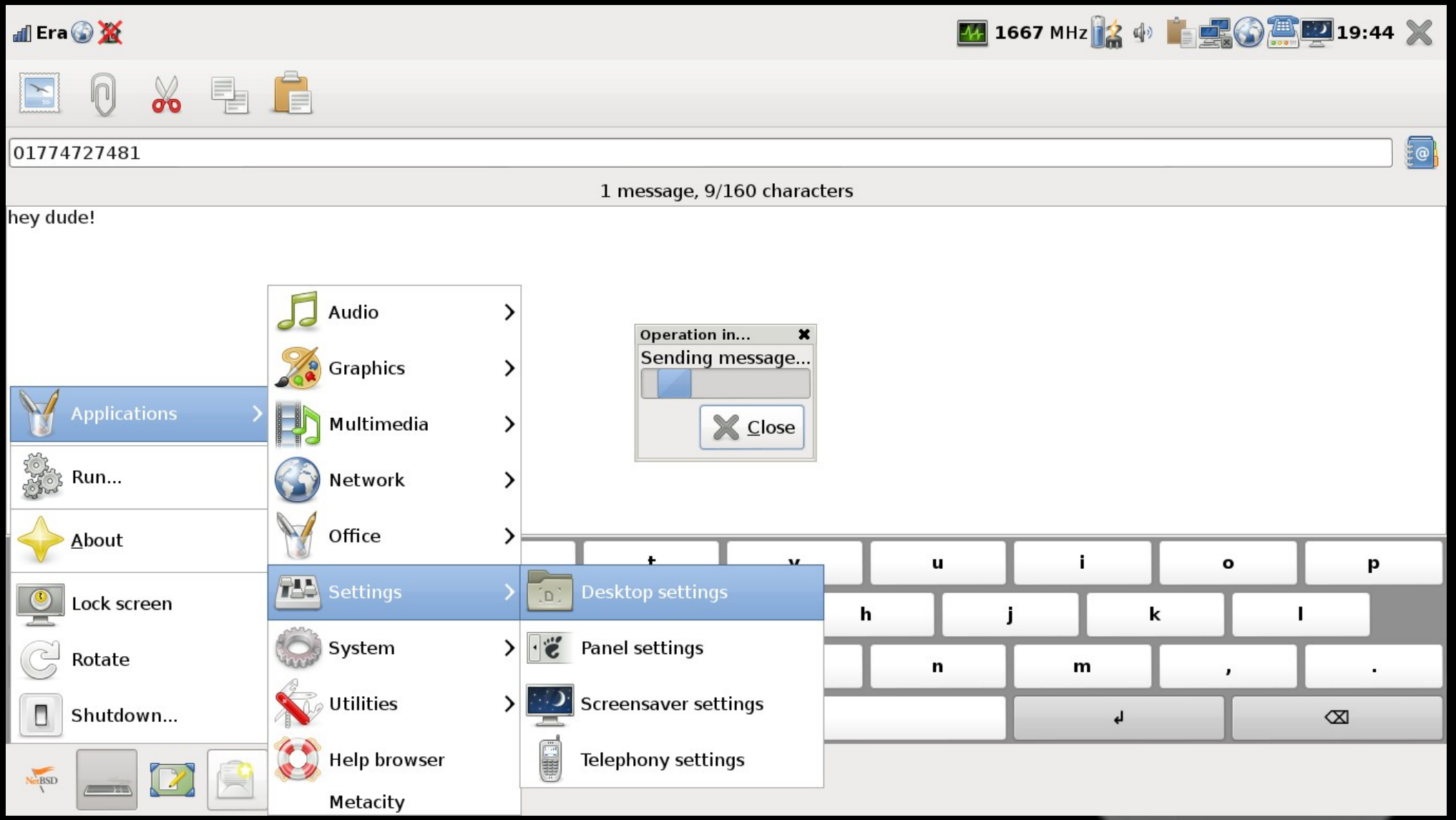
6. Current status: other devices

Device	Driver	Status
Battery charger	?	Not implemented
Bluetooth	?	Not implemented
Camera (back)	?	Not implemented
Camera (front)	?	Not implemented
GPS	?	Not implemented
GSM	?	Not implemented
LED torch	?	Not implemented

6. Current status: summary

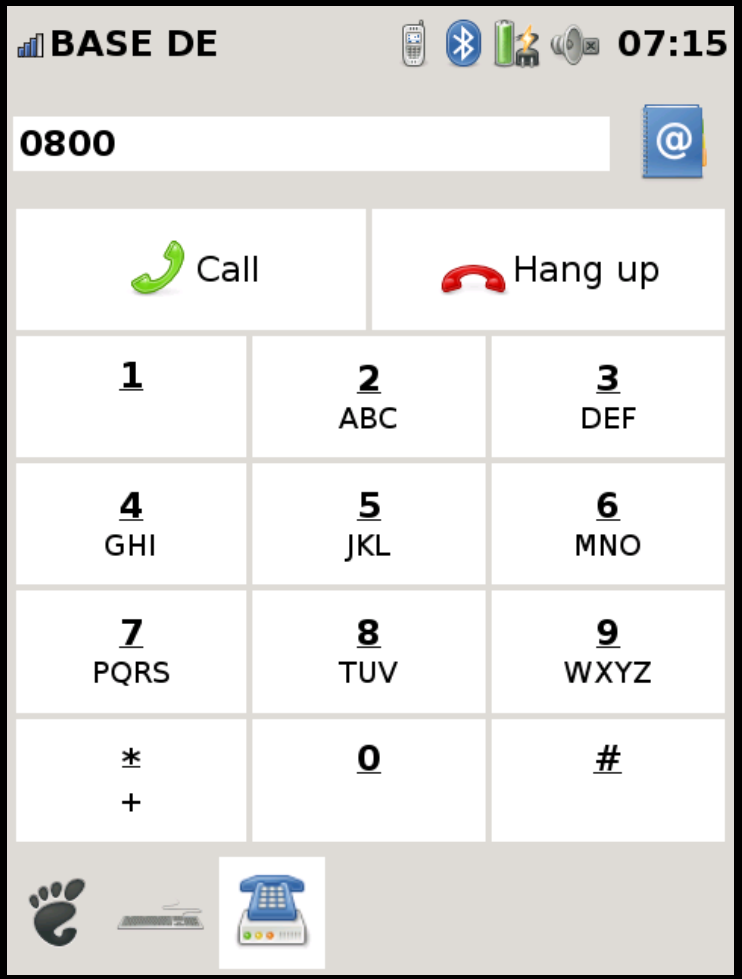
- It boots multi-user
- It starts X (xdm)
- Keyboard works (but super slow, no key combinations)
- Touchscreen doesn't work yet
- Not much else works at the moment :(

7. DeforaOS embedded desktop



7. DeforaOS simulated on a phone

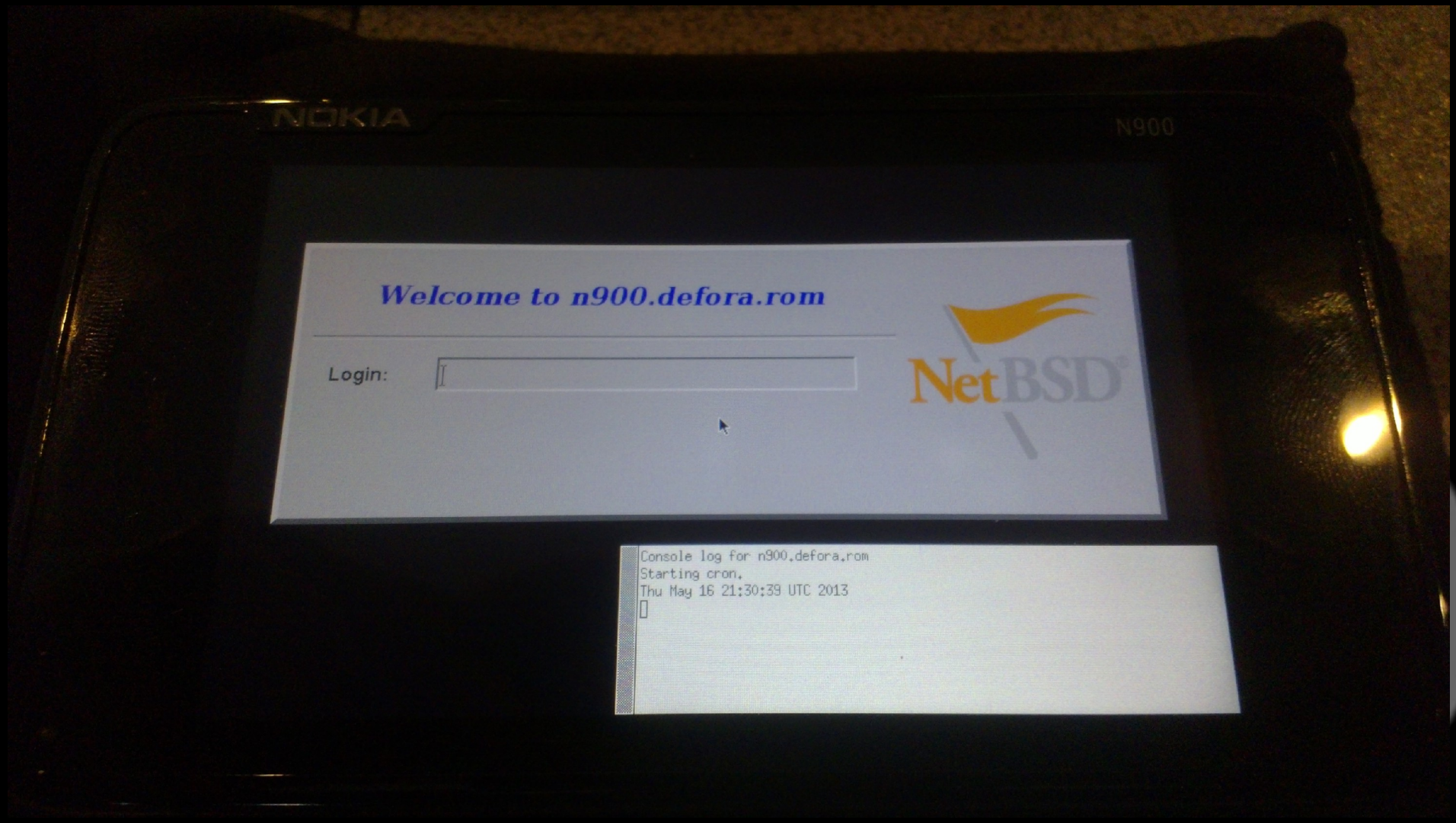
7. DeforaOS on a real phone



7. Booting NetBSD/N900

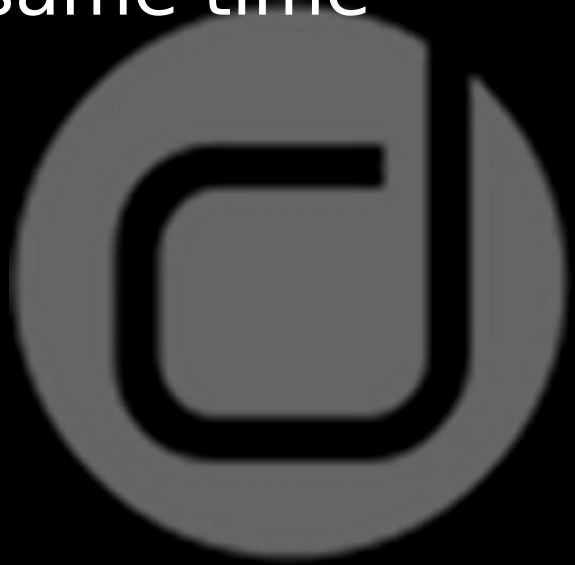
7. DeforaOS on NetBSD/N900

- Almost there (got X11 running yesterday)



8. Future plans

- Get rid of these “Not implemented” cells
- Get as much as possible into NetBSD's main tree of course
- Try to polish the existing kernel support in time for NetBSD 7
- Keep working on DeforaOS at the same time
- Support moar devices:
  - E-book reader (thanks spz!)
  - Nokia N9 (OMAP 3630)



Thank you!

To the many of you who helped:

aful, alphacc, annek, bapt, bearstech, carebear, ckab, danl/bsdcan, fbz, guigui2, imil, jared, joerg, laf0rge, manu, mbalmer, mbouyer, nokia, olivier_, oz, rhaamo, riz, rkujawa, shaddai, spz, wip, xpo, ze, zecrazytux...

Thank you!

- Hope you learned something today
- If not, plenty of other talks coming :)
- To reach me:
  - Pierre Pronchery <khorben@defora.org>
  - khorben_ on #netbsd (Freenode)
  - http://www.defora.org/
  - http://www.duekin.com/ (consulting)
- See you soon!