Full Metal Erlang

Porting Erlang to RTEMS,
to run it as close as possible to the real hardware

Peer Stritzinger
Talk at Erlang User Conference, Stockholm 2013

Talk overview

- Introduction to RTEMS
- Demo
- Some Details of the Port
- Status of Erlang on RTEMS
- Introducing Project Grisp

Introduction to *RTEMS*

A Real-Time OS for Embedded Applications

by Thomas Dörfler
embedded brains GmbH
Obere Lagerstr. 30
D-82178 Puchheim
Germany

Full intro to RTEMS slidedeck www.grisp.org/downloads/intro-to-rtems.pdf

RTEMS: What It Is

- Operating System for Realtime and/or Embedded Applications
- Supports same APIs on all major 32 bit architectures
- Allows efficient use of processing time and memory resource
- Reliable realtime behaviour
- Tailored for
  - low memory footprint (e.g. 256KByte RAM, 512 kByte ROM)
  - Low processing power (e.g. 25MHz M68k systems)

RTEMS: What it ain't

- No sophisticated MMU support
  - No virtual memory
  - No memory protection
- No (or limited) multi-user environment
- No access restrictions between internal tasks
  - File system
  - Memory
  - OS objects

Some Real World Applications

Industrial

AGVs (Automated Guided Vehicle Systems)
Navigation and Motor Control

Automotive

High-Speed Data Logger for BMW Superbike ECU (Engine Control Unit)

Space

ESA Herschel Space Telescope
RTEMS controls SMU (Spacecraft Management Unit)

Target Footprint and Requirements

RTEMS can be used on broad range of hardware

- **ROM usage: 96KByte...64MByte**
- **RAM usage: 32KByte...2GByte**
- **CPU clock: 12MHz .. >3GHz**

Realtime Behaviour (given for 25MHz MC68360):

- **context switch: 10 µsec**

System Uptime:

- **10 minutes .. 20 years**

Target Architectures

- RTEMS can be ported to almost all 32 bit architectures
- Some special architectures also supported

Current Architectures:

- PowerPC
- M68K/ColdFire
- I386
- ARM
- BlackFin
- MIPS
- NIOS
- Super-H
- SPARC

Target BSPs

h8300
h8sim

sh
simsh4
gensh1
gensh2
gensh4
shsim

Sparc
leon
leon2
leon3
erc32

hppa1.1
pxfl
simhppa

i386
go32
pc386
ts_386ex
Force386
i386ex

Arm7/9
armulator
csb337
gbag
p32
csb336
edb7312
vegaplus

unix
posix

ColdFire
uC5282
mcf5206elite
av5282
idp

MIPS
csb350
hurricane
rbtx4938
jmr3904
p4000
genmongoosev
rbtx4925
p4000

TI c4x
c4xsim

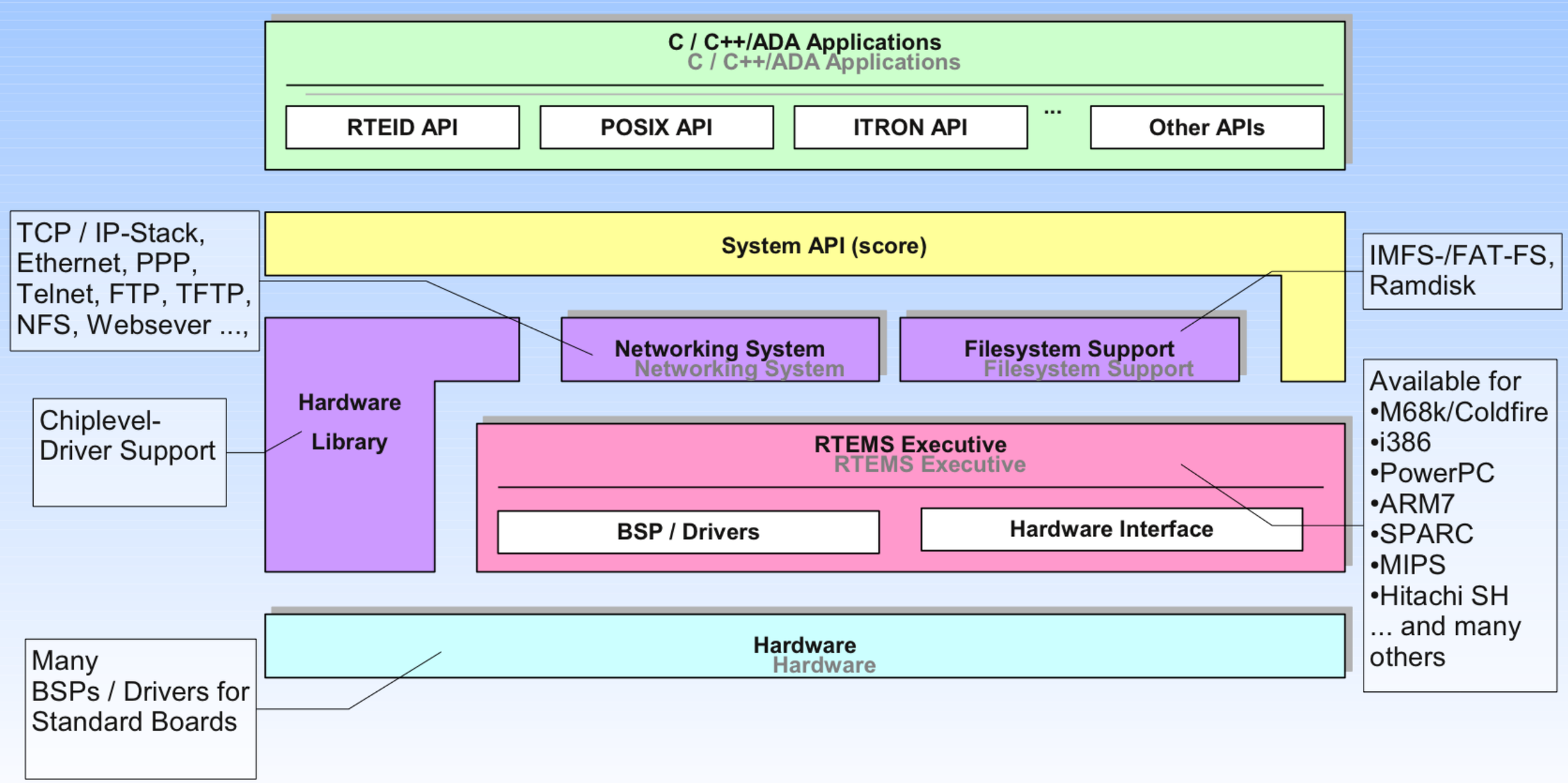
M68k
gen68302
mrm332
mvme162
sim68000
dmv152
gen68340
mvme136
mvme167
efi332
gen68360
mvme147
ods68302
csb360
efi68k

BlackFin
ezKit533

PowerPC
eth_comm
mvme2307
score603e
gen405
mbx8xx
mvme5500
gen5200
mcp750
ss555
dmv177
motorola_ppc
ppcn_60x
ep1a
mpc8260ads
psim
virtex

nios
nios2_iss

SW Structure




hydraprog3
hydraprog3


hydraprog3


Hydraprog 3
Net 1
Net 2
Net 3
Main Pwr


X1
X2
HP3 GW-LCFM
X3

Mainboard
Mini ITX
Intel Atom

FreeBSD, Erlang

Gateways
custom
Freescale
MPC5517

RTEMS

Ethernet

USB

USB

Ethernet Switch / USB Hub

Examples for systems using RTEMS cont.

- Hydraprog-3, Reflashing system for Automotive "Embedded Control Units" (= ECUs)

- RTEMS is used on the gateway boards that talk the actual automotive field bus controls

- All high-level processing is done by Erlang running on FreeBSD, talking via USB2

- Fore more details see my talk at Munich EFL 2013

Examples for systems using RTEMS

- Boschrexroth ID40 RFID System

Demo

Demo

- PowerPC embedded controller MPC 8309
- 128 MB NAND Flash with YAFFS filesystem
- 256 MB SDRAM (16bit bus)
- Ethernet Switch
- Bootloader also on RTEMS

Some Details of the Port

otp/xcomp/erl-xcomp-powerpc-rtems.conf

```
erl_xcomp_build=guess
erl_xcomp_host=powerpc-rtems4.11

erl_xcomp_configure_flags="--without-termcap"

CC='powerpc-rtems4.11-gcc --pipe
        -B/usr/home/peer/rtems-4.11-newgit/powerpc-rtems4.11/br_uid/lib/
        -specs bsp_specs -qrtems -fno-keep-inline-functions -mcpu=603e
        -meabi -msdata -fno-common -mstrict-align -fno-strict-aliasing'

CFLAGS= -g -Wall -O2
LIBS='-ldemo -lnfs -lquicc -lbed -lyaffs2'

erl_xcomp_poll=no
```

Drivers patched into OTP tree

otp/erts/emulator/drivers/unix/spi_drv.c

```
--- a/erts/emulator/Makefile.in
+++ b/erts/emulator/Makefile.in
@@ -787,6 +787,10 @@ ifneq ($(findstring vxworks,$(TARGET)),v
 	DRV_OBJS += $(OBJDIR)/ttsl_drv.o
 endif

+ifeq ($(findstring rtems,$(TARGET)),rtems)
+	DRV_OBJS += $(OBJDIR)/spi_drv.o
+endif
+
 ifeq ($(ERTS_ENABLE_KERNEL_POLL),yes)
 OS_OBJS += 	$(OBJDIR)/erl_poll.kp.o \
 		$(OBJDIR)/erl_check_io.kp.o \
```

Normal Cross-Build

- After patching the OTP tree normal cross-build works

```
$ ./otp_build autoconf
$ ./otp_build configure --xcomp-conf=xcomp/erl-xcomp-powerpc-rtems.conf
        --disable-threads --prefix=/otp

$ ./otp_build boot -a

$ gmake install DESTDIR=$HOME/export/uid-mnt

$ powerpc-rtems4.11-objcopy -O binary beam beam.bin
```

Next Steps

- epmd replacement almost done
  - Only one Erlang node possible per RTEMS
  - Simple epmd replacement written in Erlang
  - Started before net_kernel

Dynamic Linking

- Needed for NIFs
- Convenient for linked in drivers
- Separate build of BSP + C-Application / Beam
- New RTEMS Linker will make this possible

Future

- FreeBSD kqueues in RTEMS
- SMP support for RTEMS (in progress for ARM)
- RTEMS C-Node for smaller hardware
- Directly message RTEMS threads from Erlang

"Erlang on RTEMS" sounds a bit awkward ...

Lets call it ...

Project Grisp

- Erlang on RTEMS
- Supporting Libraries
- Port to Beagle Bone as turnkey evaluation platform
- More to come ...
- Check www.grisp.org

Thanks to

- For funding the Erlang and RTEMS port and being a great partner for innovations like this:

Rexroth
Bosch Group
The Drive & Control Company

- For great RTEMS support, nice hardware design and helping out with the RTEMS Intro slides

embedded brains GmbH®

www.embedded-brains.de

Questions

- www.grisp.org
- www.stritzinger.com
- Twitter/ADN @peerstr
- peer@stritzinger.com