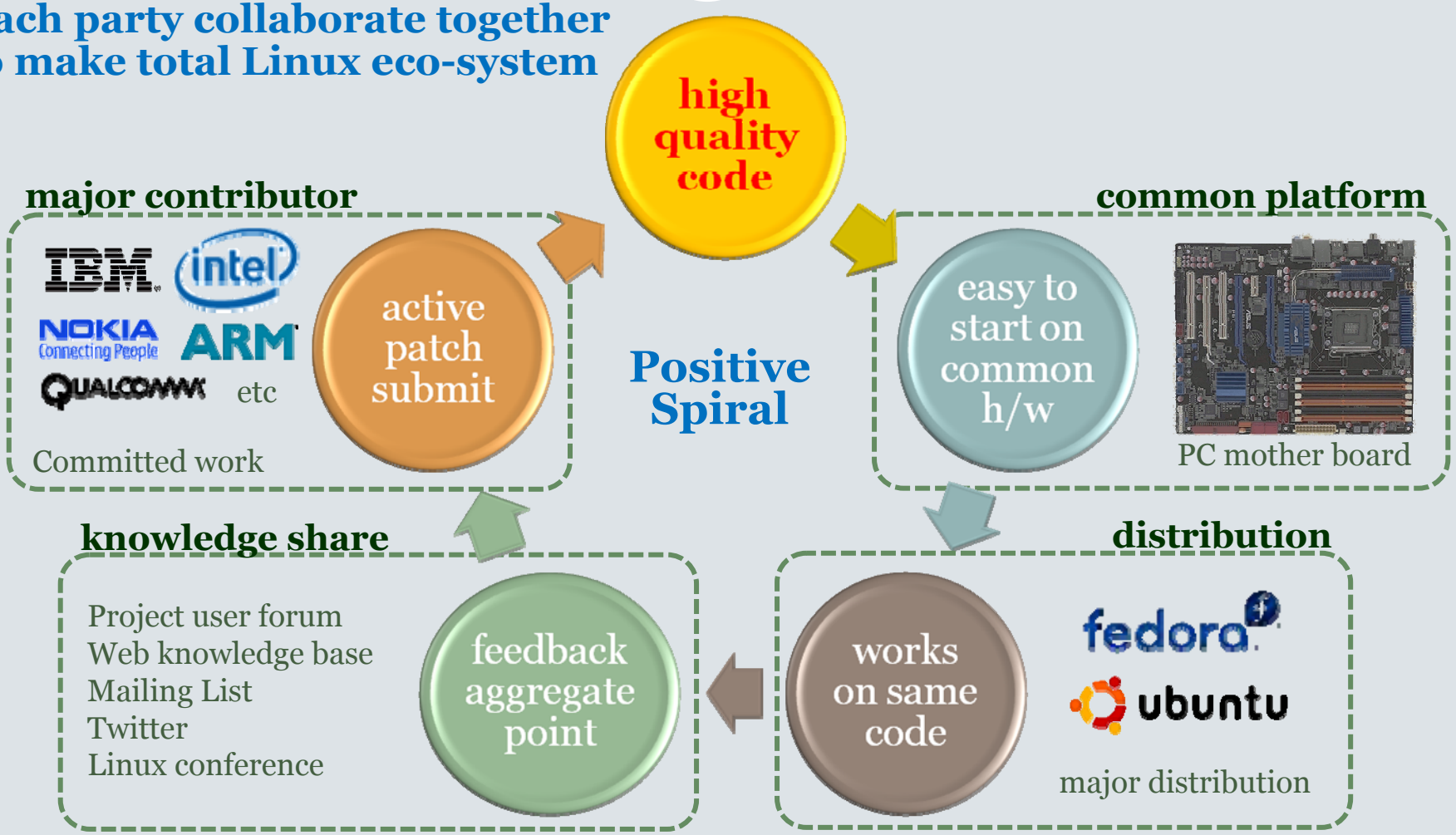
Rescuing SuperH to Linux Commonplace

HISAO MUNAKATA
RENESAS SOLUTIONS CORP.

Ideal open source eco-system

Each party collaborate together to make total Linux eco-system

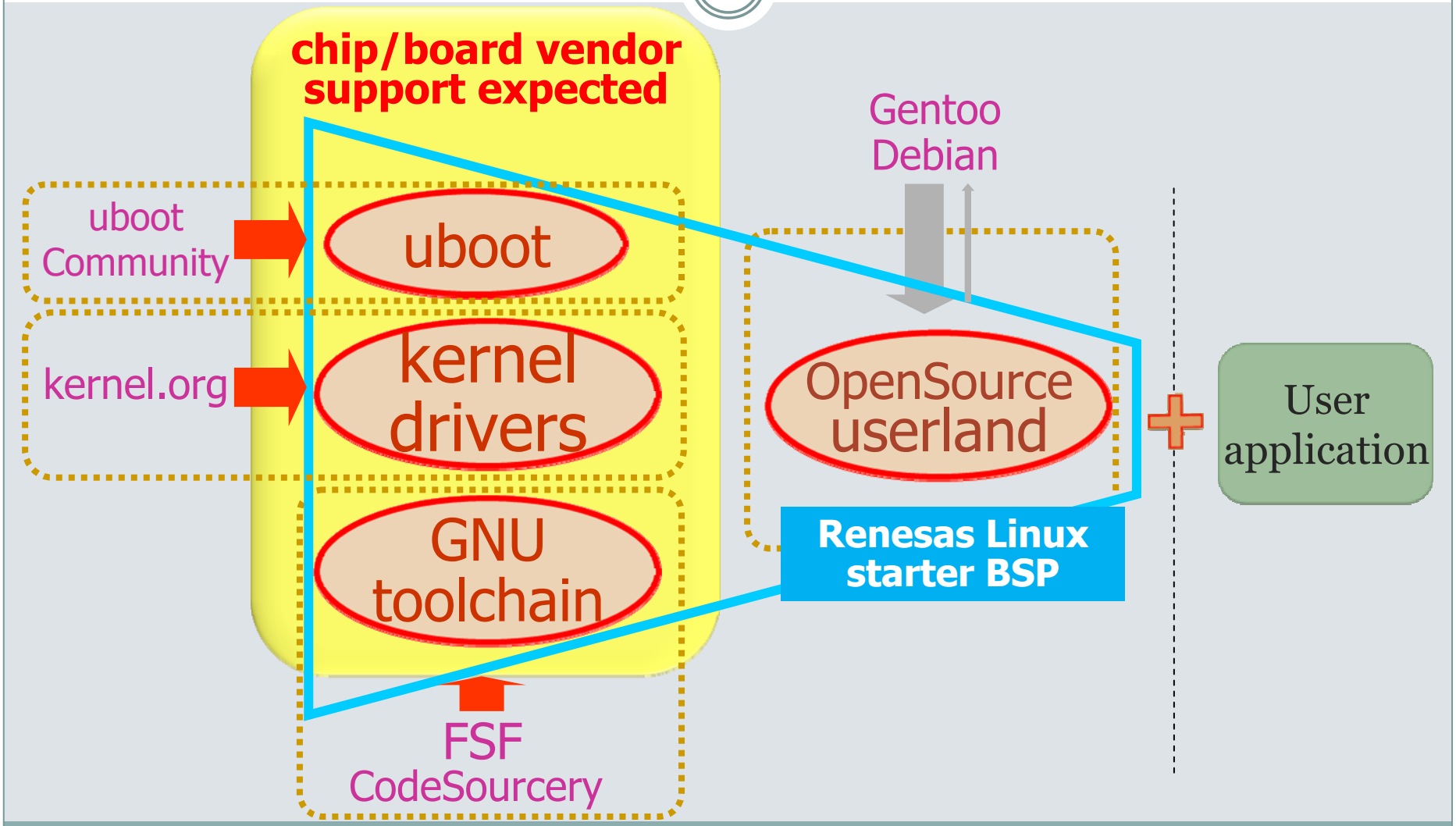


Linux support various non-x86 CPU

- 2.6.31 includes 23 CPU architecture[*1] support, but ...
  - IA32 is a default development environment, and big majority of kernel code are tested only on IA32 (=80x86) environment.
  - All Posix compliant open source program could be run on embedded Linux environment, however its build process (configure, make) does not designed for embedded Linux use.
  - SuperH is a Renesas original 32bit RISC processor designed for embedded products like cell-phone, digital TV and Car navigation system. Linux kernel natively support SuperH (=SH) from 2.6.1. But its adoption is not straightforward like PC Linux.

1: alpha, arm, avr32, blackfin, cris, frv, h8300, ia64, m32r, m68k, m68knommu, microblaze, mips, mn10300, parisc, powerpc, s390, score, sh, sparc, um, x86, xtensa

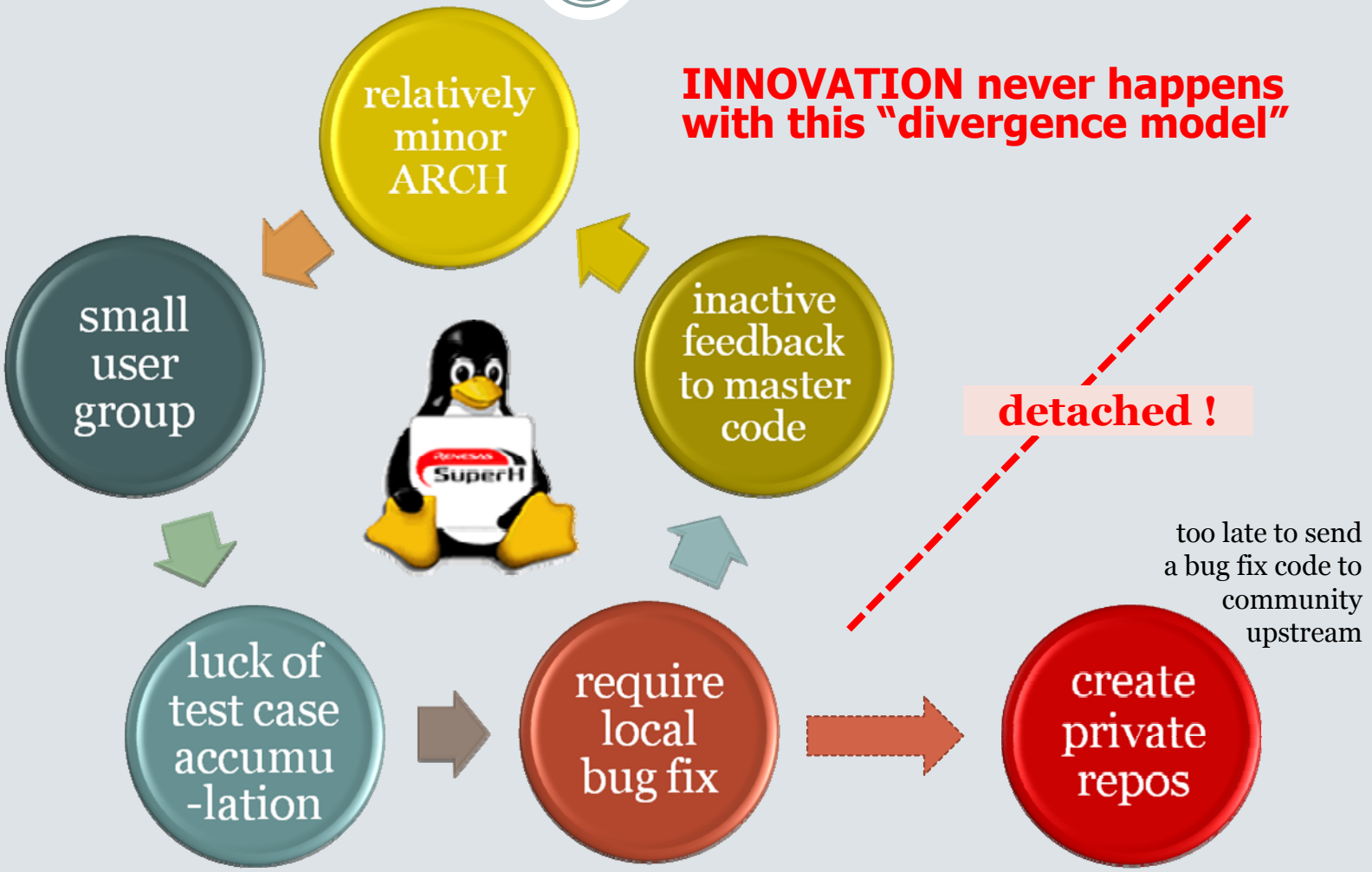
Embedded Linux starter components



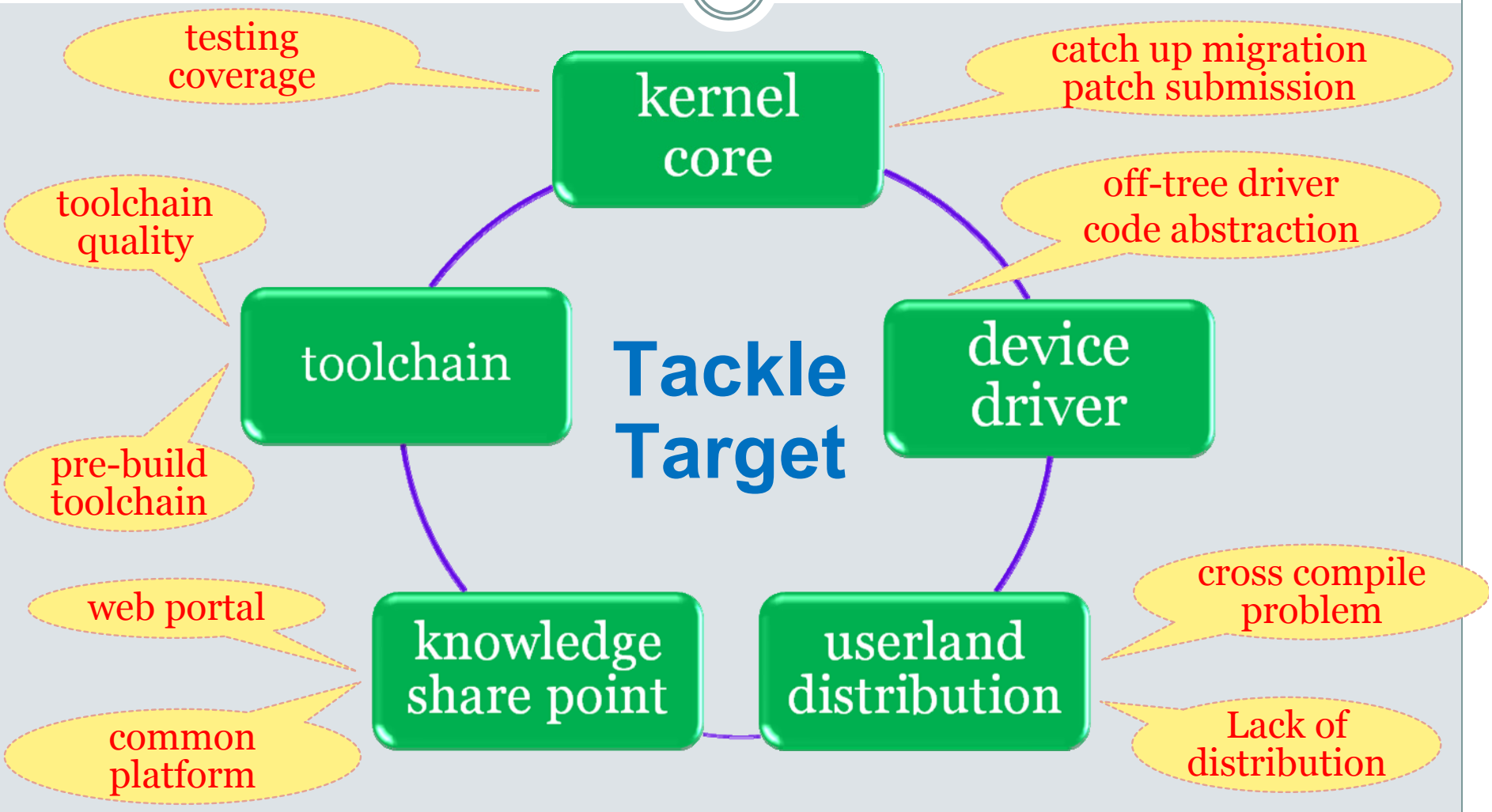
Common embedded Linux headache
(which Renesas had experienced a few years ago)

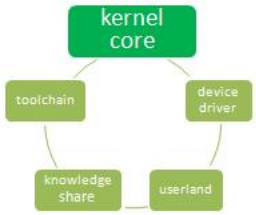
- Hard to find proper target platform that come with Linux BSP
  - Platform support in kernel was not enough maintained, some are neglected
- Lack of document, knowledge share point for SuperH Linux
- Hard to find proper toolchain that support SuperH Linux build
  - Linux build requires glibc support and other extension against default gcc
- Not all new kernel capabilities are available on SuperH
- Hard to find pre-built SH kernel image (no generic binary)
  - SH-3, SH-4, SH-4A, SH-4AL requires different binary due to difference of ABI
- Require off-tree device driver to enable on board device
- No Linux distribution that support SuperH
- Cross compilation of generic open source code is quite problematic
  - autoconf, make sometime does not work with cross build environment
  - Some opensource program like Postgre, Apache uses architecture depended lock mechanism that can not runs on non-x86 architecture but pthread.

Typical negative spiral
(that likely create off-tree local repos)

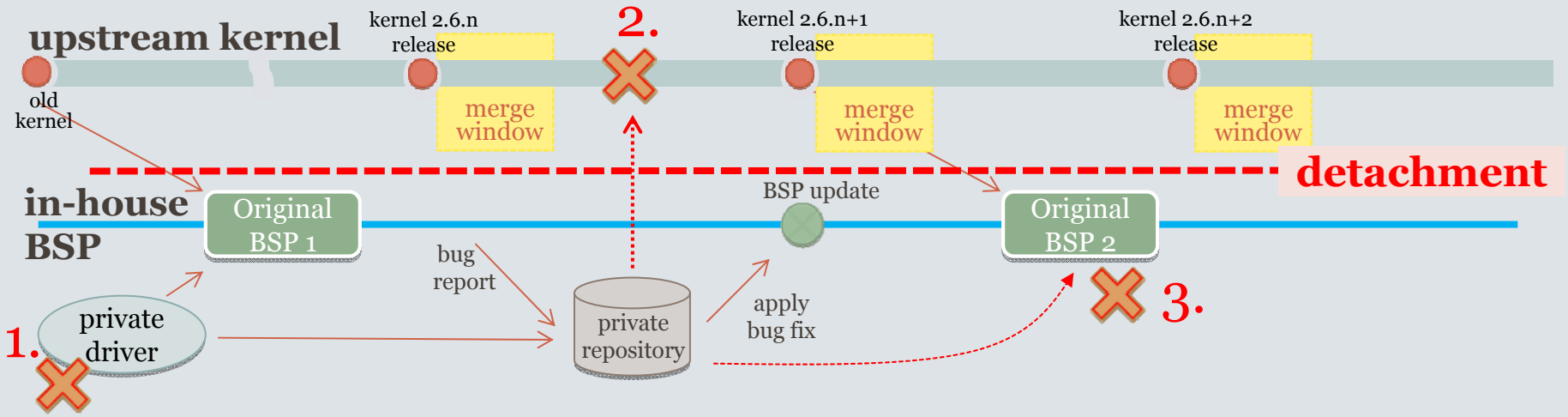


Gap analysis (to renovate negative loop)

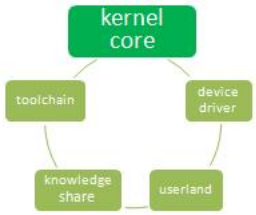




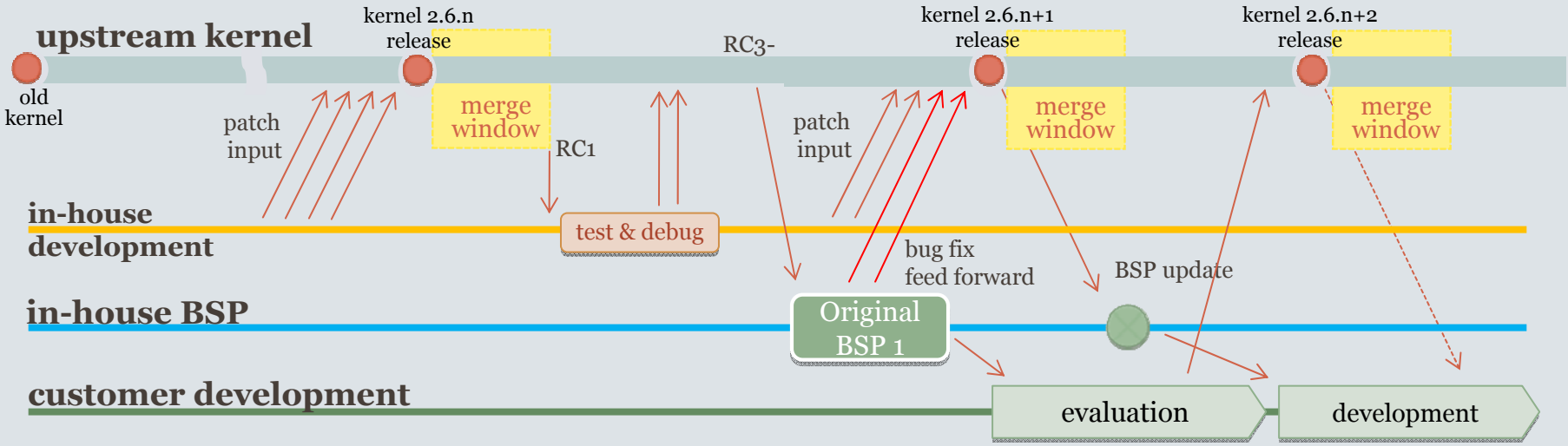
SH kernel development (3 years ago)



- Use already released kernel (3 – 6 month old kernel at earliest)
- [1] Need to add private device driver as they are not merged in kernel
- Manage private repository to collect bug fix patches and private driver
- [2] Upstream community do not accept patch for outdated old driver/kernel
- [3] In the event of Linux BSP update, we simply pull newly released kernel, but it include neither private driver nor local bug fix patch.

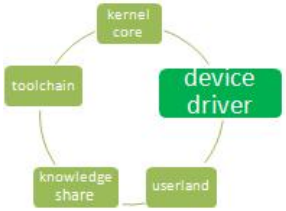


kernel development (now)

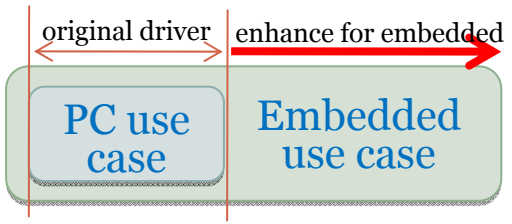


- Before merge window open, we send patch as much as possible so that these request could be merged at next version kernel.
- Once RC1 released, we test RC version kernel to stabilize our code.
- Create new BSP with RC version kernel, and revise it when its official release
- If any issue found, we send patch for next (n+1) release as well as own BSP

[Now our kernel development activity is fully synchronized with upstream]

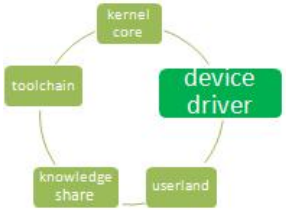


device driver development



stage	category	location	comment
I	orphan local driver	off-tree private repo	You need to add private driver to boot kernel. This makes embedded Linux fragmented.
II	platform specific driver	in-kernel platform specific code	community reviewed code, but still isolated in platform specific location. Hard to catch up future kernel migration.
III	kernel common driver	in-kernel common code	community reviewed, and fully integrated with common driver code. Driver will be maintained at every kernel migration easily

- As kernel space device driver API is heavily maintained, all driver code should be reviewed at every kernel version migration to apply appropriate kernel/driver interface.
- Linux common driver is the best (and may be the only) place to maintain driver for long time.
- Some common driver might not support embedded usage because original driver is written for PC. You should request original driver author to enhance his code to support extended embedded use.
- You should eliminate platform specific driver as much as possible, because these driver are quite easily discarded at future kernel migration. It might take a few month to negotiate with original driver author, but it is worth spending time for this generalize process.



Example : SH7724 Linux driver status

off tree | patch | merged

2.6.31

- tw9910 (V4L2)
- mt9t112 (V4L2)
- CEU (V4L2)
- ak4642 (ALSA)
- FSI (ALSA)
- DA7210 (ALSA)
- Platform support
- DMA Support

SH7724 CPU
CMT clockevent
kexec_jump
kernel enhance

2.6.31-rc7

- tw9910 (V4L2)
- mt9t112 (V4L2)
- CEU (V4L2)
- ak4642 (ALSA)
- FSI (ALSA)
- DA7210 (ALSA)
- Platform support
- DMA Support

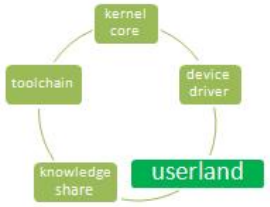
Platform support
CPU idle
U-Standby
kernel enhance

2.6.32-rc4

- tw9910 (V4L2)
- mt9t112 (V4L2)
- CEU (V4L2)
- ak4642 (ALSA)
- FSI (ALSA)
- DA7210 (ALSA)
- Platform support
- DMA Support

R-standby (queued)
kernel enhance

Now you can boot SH7724 reference platform without private driver.



SuperH Debian support status

(We work with Debian project to add SuperH support)

http://buildd.debian-ports.org/status/architecture.php?a=sh4

Package(s): Go

[avr32] [hurd-i386] [m68k] **[sh4]**

Buildd status of sh4

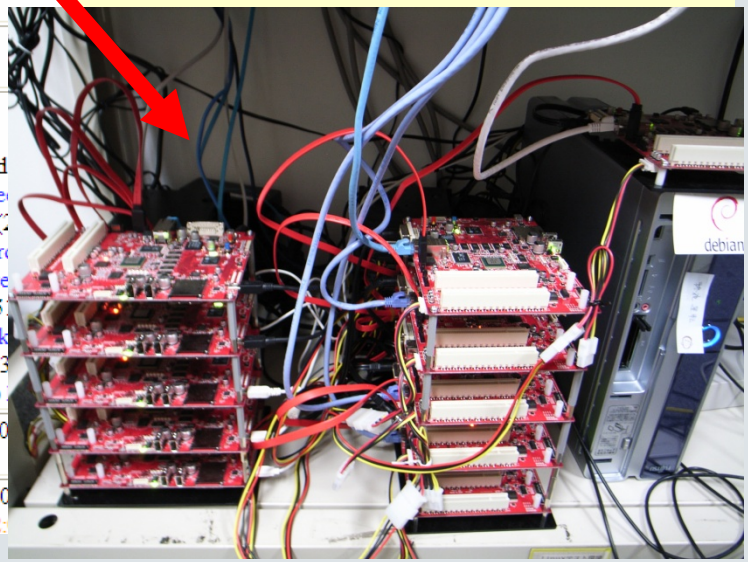
10 target test boards

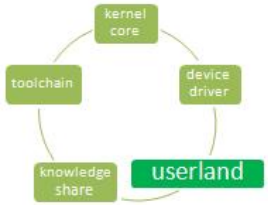
Restrict on buildd: **[all]** [yamashiro] [mutsu] [hiei] [kongou] [amagi] [nagato] [huso] [hyuga] [yamato] [musashi]
[all] [only out-of-date]

The time indicates for how long a package is in the given state. A/B means that on A out of B other architectures whe the the build succeeded. the name indicates the build deamon used for last build.

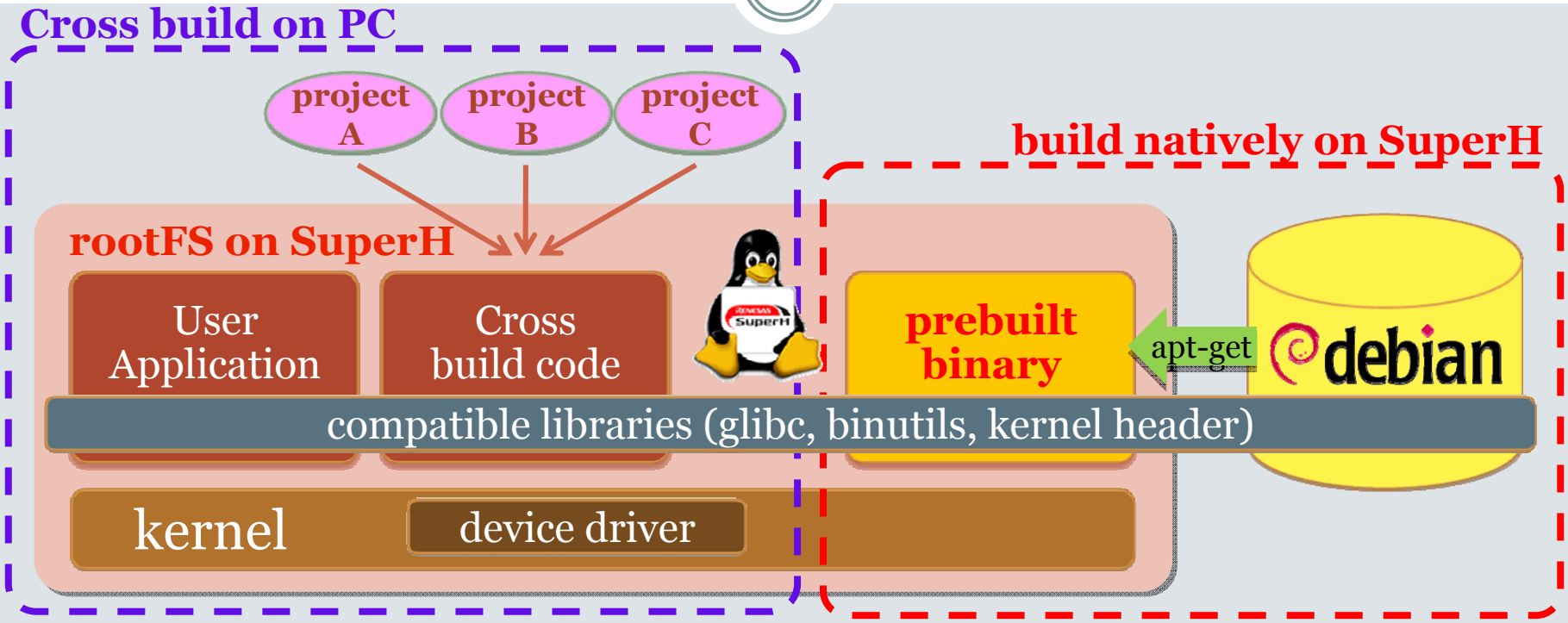
BD-Uninstallable	1439	*Too many results, cannot display*
Building	48	**State: Building (48)** linux-latest-2.6 (8d 10:56, 0/1, musashi) passenger (7d 21:41, 0/0, kongou) kdemultimedia (6d 17:08, 1/1) fenix (5d 14:28, 1/3) telepathy-glib (5d 13:19, 1/2) python-visual (5d 02:03, 0/0) e 05:37, 2/2) soya (3d 18:18, 1/1) coinor-csdp (3d 09:28, 0/0) qt4-x11 (2d 18:13, 0/1) eglibc (st (1d 05:28, 0/0) ocamlpam (23:30, 2/2) meta-gnome2 (20:59, 2/2) rumor (20:30, 1/1) firebir (13:36, 0/0) gst-plugins-good0.10 (09:45, 0/0) libgda4 (09:29, 2/2) libgweather (09:00, 1/2) re gtkboard (06:33, 2/2) gupnp (06:20, 2/2) libkipi (06:02, 2/2) heimdal (05:53, 0/0) librest (05:5 1/1) potracegui (05:02, 1/1) hotswap (05:01, 1/2) dar (04:28, 0/0) gpe-calendar (04:27, 1/1) k (04:04, 2/2) skim (03:59, 2/2) klineakconfig (03:32, 0/0) kst (03:21, 2/2) lineak-kdeplugins (03 (02:19, 2/2) vidalia (02:13, 0/0) guidedog (02:08, 2/2) kitty (01:51, 2/2) knetfilter (01:18, 1/1)
Dep-Wait	6	lisaac (8d 07:42, hyuga) mozart (8d 07:41, hyuga) a2ps (7d 22:41, amagi) freedink-dfarc (4d 0 (18:43)
		gambc (8d 22:01, 0/2, musashi) gauche-c-wrapper (8d 22:01, 1/1, yamashiro) gcl (8d 22:01, 0 0/2, nagato) g-wrap (8d 22:01, 2/3, huso) pgocaml (8d 22:01, 1/2, kongou) slang-slirp (8d 22:

Official Debian packages must be compiled upon real (native) hardware. Renesas will host Debian repository server to distribute these binary

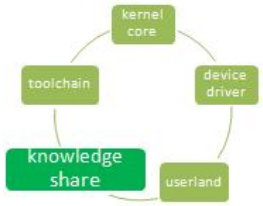




SuperH Linux rootFS
(prebuilt Debian packages + cross build environment)



- Official Debian package must be compiled on native environment
- Cross toolchain is suitable for kernel and user application build.
- You may want to cross compile open source project that are not in Debian.
- So Glibc/binutil compatibility between native and dross is required in this case.

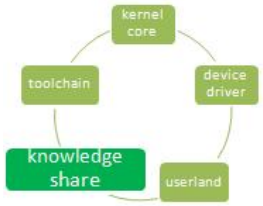


Renesas open source portal

http://oss.renesas.com/

Renesas launches our first official open source portal that distribute

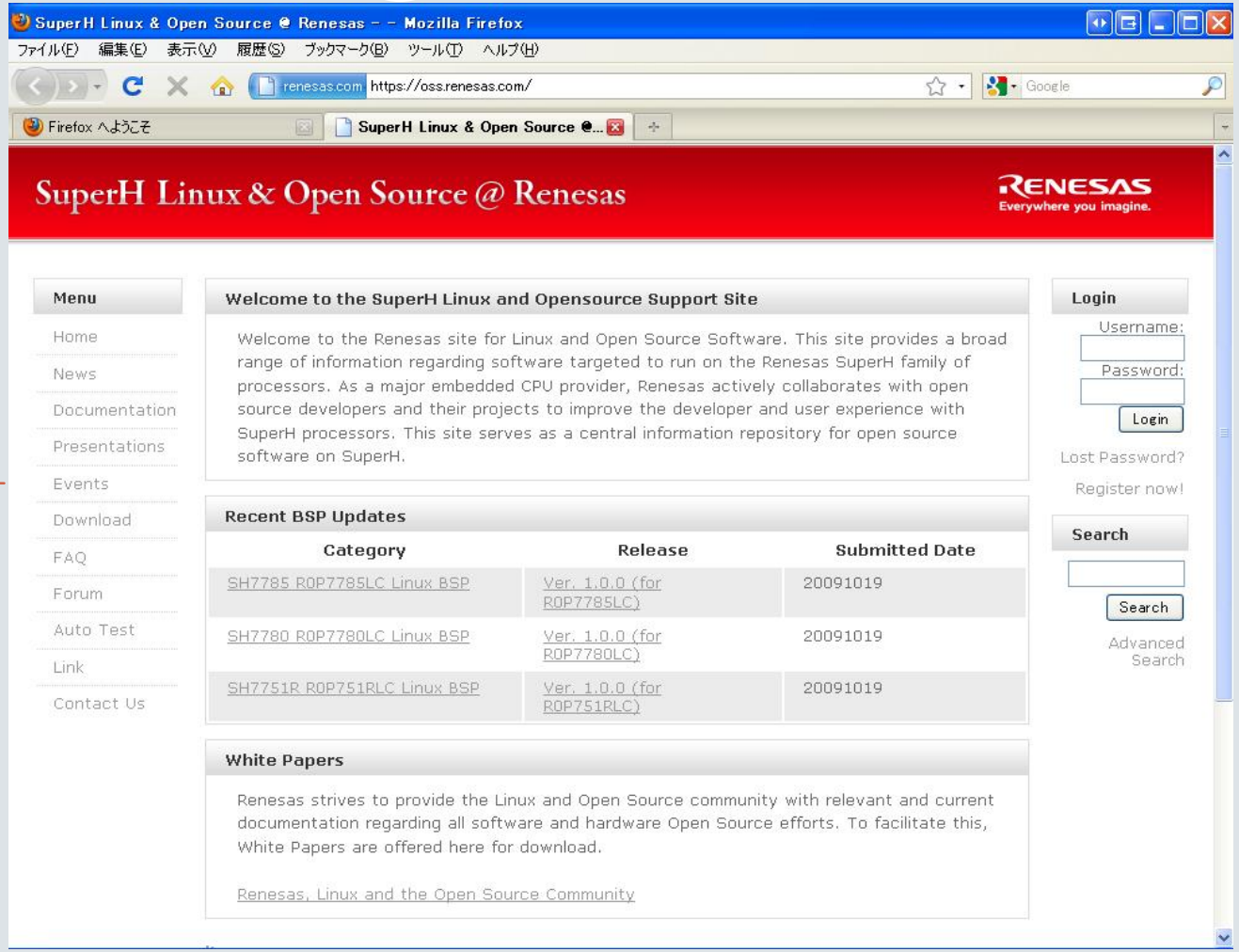
- Download
  - pre-built Linux BSP (kernel [vanilla] + userland [Debian], toolchain)
  - Full source code for our Linux BSP
  - Pointer to public space asset related to Renesas open source activity
- User Forum (common knowledge share point for SuperH Linux users)
- Document / FAQ
- Automated test report (kernel built, LTP, lmbench etc)
- Renesas open source related event info and its materials
- Bug report and tracking

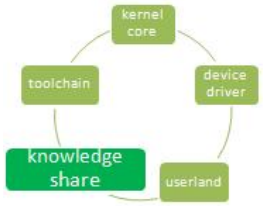


Renesas open source portal (Top)

http://oss.renesas.com/

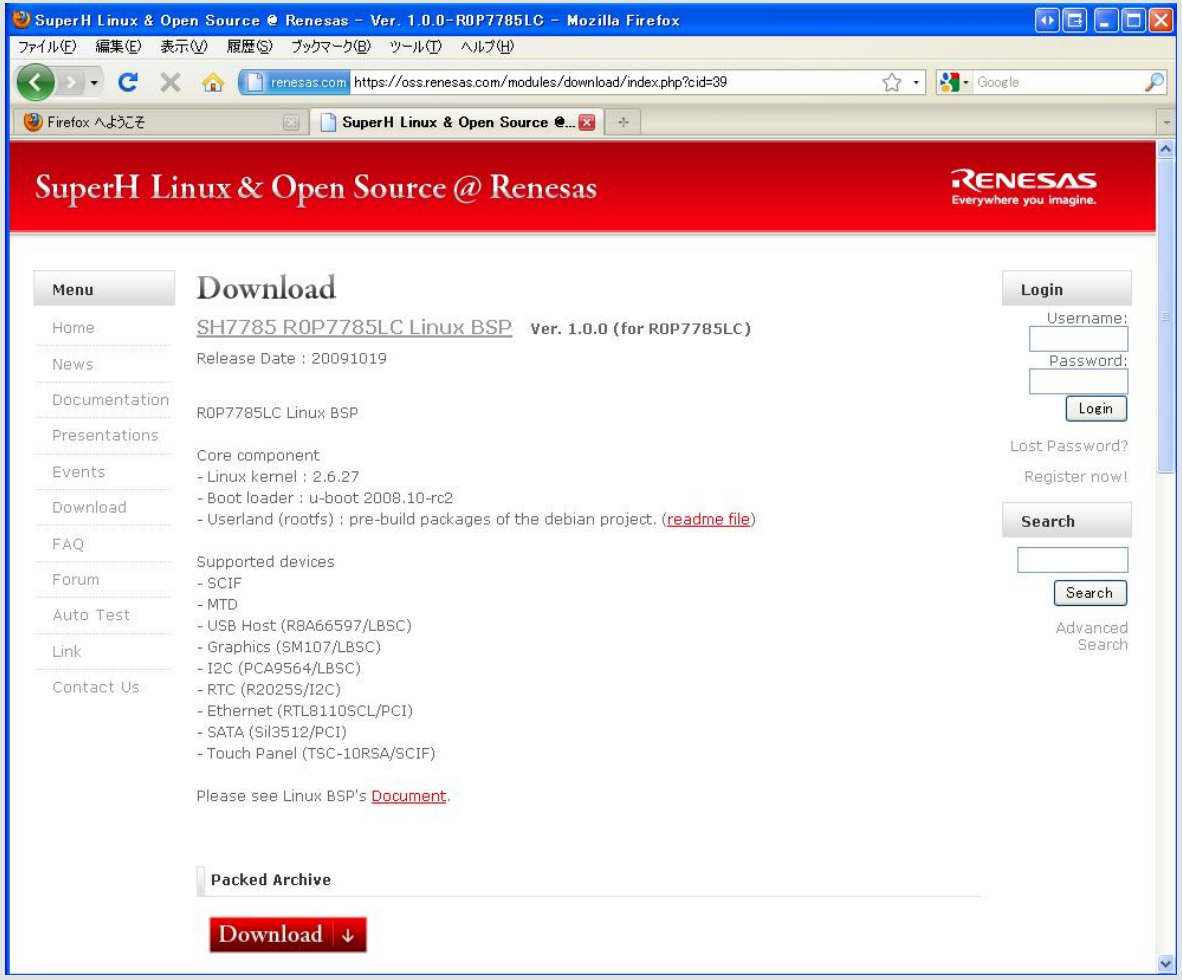
- News
- Documentation
- Presentation
- Events
- FAQ
- Forum
- Auto Test
- Link
- Contact Us





Renesas open source portal (download)

http://oss.renesas.com/

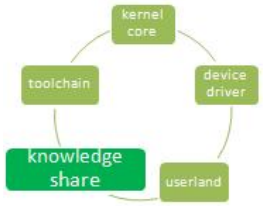


prebuilt binary

- uboot
- kernel
- Debian
- toolchain
- rootFS image
- Sample code

full source code

- kernel
- All package



Linux reference platform
[Renesas Multimedia Solution]

Processor Key features

SH-Mobile R2R (32bit RISC @500MHz)
64MB NOR Flash / 256MB DDR2-SDRAM

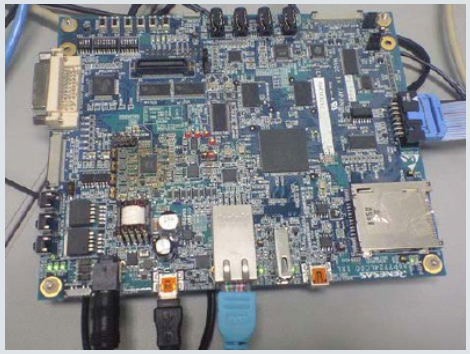
USB HOST : Type A Connector
HOST/Function : Type mini A/B Connector
2ch Camera Interface (support 720p HD input)
Video Processing Unit (VPU5F) for 720p enc/dec
Sound Processing Unit (SPU) for MP3, AAC+ etc.
2D Graphics Engine
JPEG Processing Unit (JPU)
2ch SD card Interface
10/100Mbps Ethernet MAC

Platform Key features

PoE(IEEE802.3af) class3(12.95W)
Motion Sensor
DVI, Composite Video (NTSC/PAL)
7' WVGA, Touch panel (Option)
Camera Board for 720p (Option)

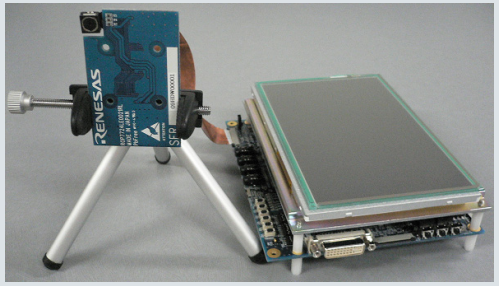
Target application

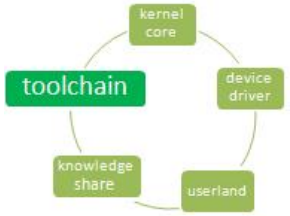
IP Surveillance Camera
V2IP (Video and Voice over IP)
Video Conference System
Portable Navigation Device
Graphical/Video User Interface



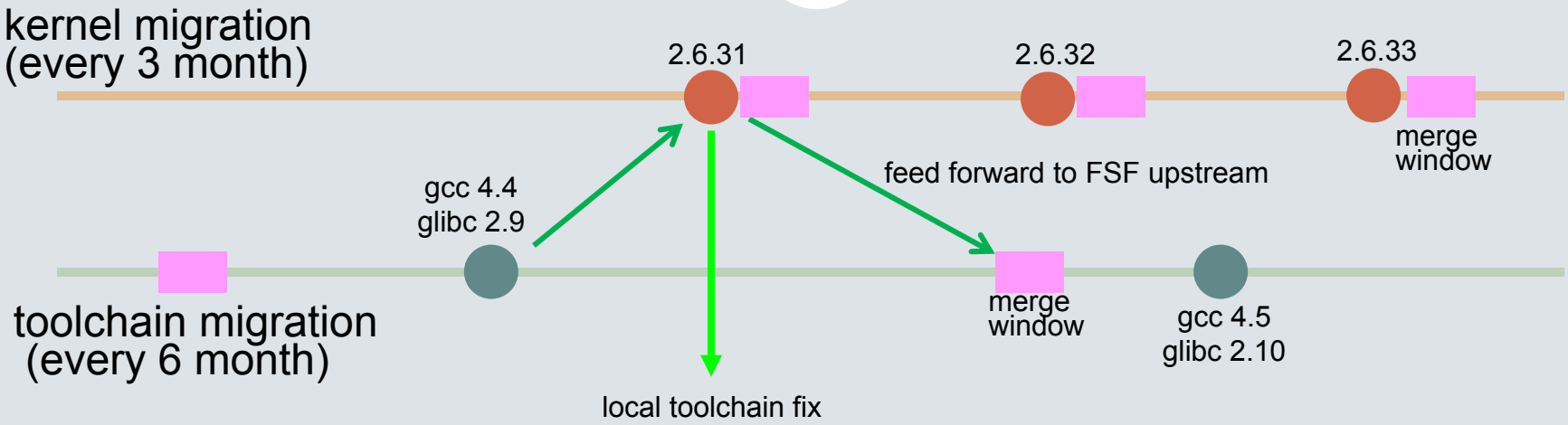
Complete Linux Software Package

Boot loader	Linux kernel using kexec_jump and uboot
Boot method	NFS, USB, NOR etc
Linux kernel	version 2.6.31 vanilla
Tool chain	gcc 4.3.4, glibc 2.9, binutil 2.19.51, kernel header 2.6.29
Userland (rootfs)	Debian based collection (apt-get management available)
Device Driver	SCI, Timer, DMA, Ether, USB, i2c , lcdc, video, audio etc
Multimedia Framework	OpenMAX/IL (Bellagio), Gstreamer support included
DirectFB	layered graphics and 2D hardware acceleration support
Renesas custom library	libshcodecs, libshjpu, libshveu····
Sample Application	camera capture, video streaming (http, rtsp), media player

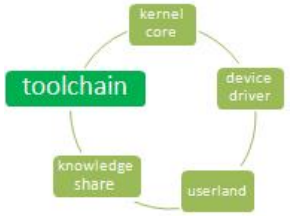




toolchain development



- Linux kernel have some relevancy with GNU toolchain
  - new kernel function depends on extended glibc (or uclibc) function
  - kernel source description is compliant with compiler syntax check
  - over all Linux system stability is often caused by toolchain
- So we need to keep GNU toolchain for SuperH enough maintained too
- Sometime we need to feed forward patch to FSF current development tree



SuperH Linux toolchain
(CodeSouecery G++ Lite)

CodeSourcery

CodeSourcery

SOURCERY G++

FEATURES | PLATFORMS | EDITIONS | SUPPORT | BUY

CodeSourcery > Sourcery G++

Resources

- Sourcery G++ Data Sheet
- Register for a Sourcery G++ Evaluation
- Buy Sourcery G++ Today!

Questions

- What target platforms does Sourcery G++ support?
- What's included in Sourcery G++?
- Which edition is best for me?
- What's new in the latest release?

Sourcery G++ 4.4-46 for SuperH GNU/Linux Datasheet

TRY NOW! BUY NOW!

Contents

- Target Systems
- Features
- Run-Time Libraries
- Support
- Editions
- Host System Requirements

Target Systems

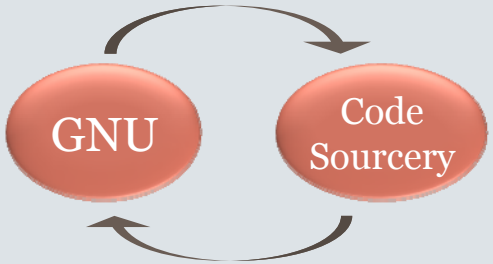
Systems running "full" Linux, i.e., Linux on CPUs with an MMU. Use Sourcery G++ to build both the Linux kernel and applications.

Features

Component	Version	Personal	Professional
Eclipse IDE	Galileo	✔	✔
GNU Binary Utilities	2.19.51-sg++	✔	✔
GNU C & C++ Compilers	4.4.1-sg++	✔	✔
GNU C Library	2.10-sg++	✔	✔
GNU Debugger	6.8.50-sg++	✔	✔
Graphical Installer		✔	✔
Library Reduction Utility		✔	✔
Sysroot Utilities		✔	✔
uClibc C Library	0.9.30-sg++	✔	✔

Version numbers shown for open-source components indicate the versions used as the basis for Sourcery G++ for SuperH GNU/Linux. CodeSourcery makes extensive enhancements to the base versions, adding support for more CPUs, improving code-generation, and addressing defects found through its validation process.

CodeSourcery will submit all their work (bug fix, functional improvement) to FSF upstream. So these improvement would be reflected to future FSF master.



http://www.codesourcery.com/sgpp/datasheet?target_arch=SuperH&target_os=GNU%2FLinux

demo

Conclusion

- “Openness of Linux” is not limited to source code exchange, we should leverage current freedom of CPU choice. I believe “variety” is a one of key factor for evolution of Linux.
- Linux is a moving target. If you want to success with open source, you can not stick on specific point. To catch up with innovation, joining its open development process is the best (and only possible) practice.
- SuperH is one of minor architecture, but Renesas try to keep it “standard” and “healthy” in Linux world.