LTSI project update (2015 fall editoin)

How to choose the best kernel for your embedded system

Hisao Munakata

LTSI Project @ Linux Foundation, CE working group

October 5th 2015

Who am I ?

- From embedded SoC provider company Renesas
- Responsible for OSS software development and delivery for R-Car series SoC
- Working with W/W car OEM and 1st tear IVI customers

- Linux Foundation CE[1] working Gr. Steering committee member, LF/CEWG Architecture Gr. co-chair
- One of LF/CEWG LTSI[2] project initial proposer
- At my company, I had been encouraging my team developers to send a patches upstream

[1] CE = consumer electronics

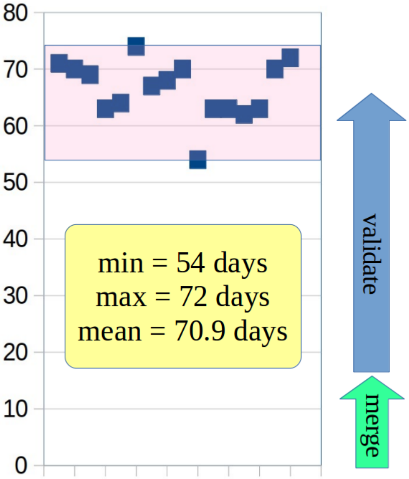
[2] LTSI =Long Term Support Initiative

What is LTS & LTSI kernel ?

community longterm kernel

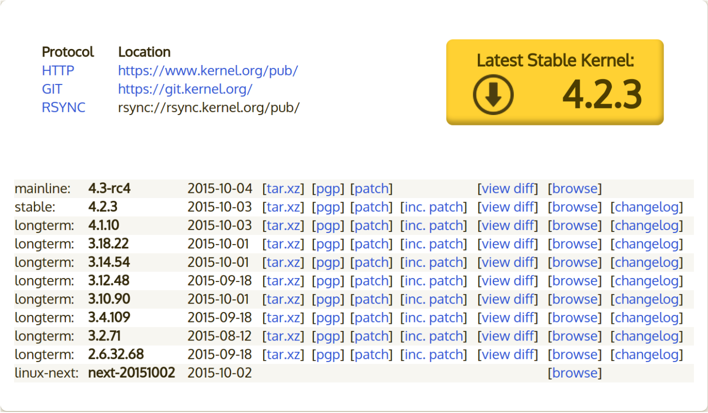
Schedule based periodic kernel release record

version	release date	duration
v3.7	2012-12-10	71 days
v3.8	2012-02-18	70 days
v3.9	2013-04-28	69 days
v3.10	2013-06-30	63 days
v3.11	2013-09-02	64 days
v3.12	2013-11-15	74 days
v3.13	2014-01-21	67 days
v3.14	2014-03-30	68 days
v3.15	2014-06-08	70 days
v3.16	2014-08-03	54 days
v3.17	2014-10-05	63 days
v3.18	2014-12-07	63 days
v3.19	2015-02-08	62 days
v4.0	2015-04-12	63 days
v4.1	2015-06-21	70 days
v4.2	2015-09-01	72 days



min = 54 days
max = 72 days
mean = 70.9 days

Upstream kernel @kernel.org



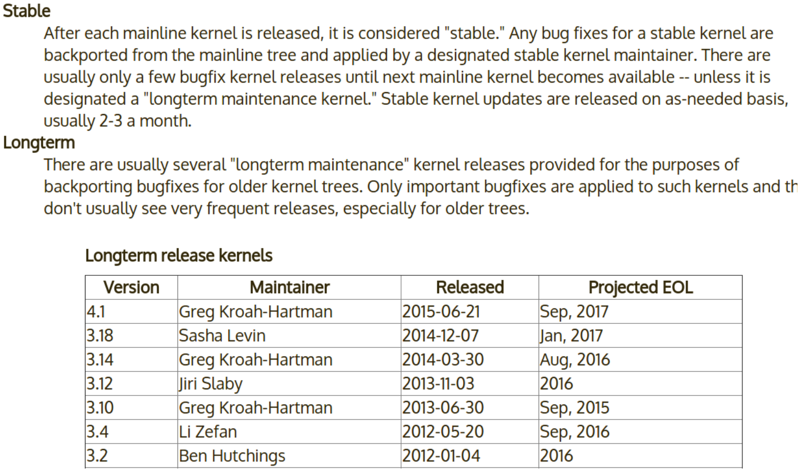
Protocol	Location
HTTP	https://www.kernel.org/pub/
GIT	https://git.kernel.org/
RSYNC	rsync://rsync.kernel.org/pub/

Latest Stable Kernel:
4.2.3

mainline:	**4.3-rc4**	2015-10-04	[tar.xz]	[pgp]	[patch]		[view diff]	[browse]	
stable:	**4.2.3**	2015-10-03	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	**4.1.10**	2015-10-03	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	**3.18.22**	2015-10-01	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	**3.14.54**	2015-10-01	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	**3.12.48**	2015-09-18	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	**3.10.90**	2015-10-01	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	**3.4.109**	2015-09-18	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	**3.2.71**	2015-08-12	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	**2.6.32.68**	2015-09-18	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
linux-next:	**next-20151002**	2015-10-02						[browse]	

You can find 1)latest released, 2)under development (=mainline, next), and several stable kernels

upstream kernel maintenance (Stable and Longterm)



Stable

After each mainline kernel is released, it is considered "stable." Any bug fixes for a stable kernel are backported from the mainline tree and applied by a designated stable kernel maintainer. There are usually only a few bugfix kernel releases until next mainline kernel becomes available -- unless it is designated a "longterm maintenance kernel." Stable kernel updates are released on as-needed basis, usually 2-3 a month.

Longterm

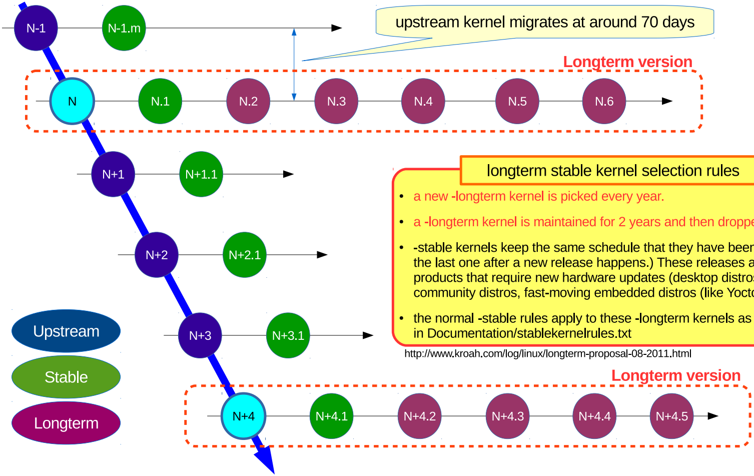
There are usually several "longterm maintenance" kernel releases provided for the purposes of backporting bugfixes for older kernel trees. Only important bugfixes are applied to such kernels and th don't usually see very frequent releases, especially for older trees.

Longterm release kernels

Version	Maintainer	Released	Projected EOL
4.1	Greg Kroah-Hartman	2015-06-21	Sep, 2017
3.18	Sasha Levin	2014-12-07	Jan, 2017
3.14	Greg Kroah-Hartman	2014-03-30	Aug, 2016
3.12	Jiri Slaby	2013-11-03	2016
3.10	Greg Kroah-Hartman	2013-06-30	Sep, 2015
3.4	Li Zefan	2012-05-20	Sep, 2016
3.2	Ben Hutchings	2012-01-04	2016

https://www.kernel.org/category/releases.html

Linux kernel life-cycle varies according to version



upstream kernel migrates at around 70 days

Longterm version

Longterm version

longterm stable kernel selection rules

- a new -longterm kernel is picked every year.
- a -longterm kernel is maintained for 2 years and then dropped.
- -stable kernels keep the same schedule that they have been (dropping the last one after a new release happens.) These releases are best for products that require new hardware updates (desktop distros, community distros, fast-moving embedded distros (like Yocto)).
- the normal -stable rules apply to these -longterm kernels as described in Documentation/stablekernelrules.txt

http://www.kroah.com/log/linux/longterm-proposal-08-2011.html

longterm kernel maintenance last longer than regular

Greg	Version	maintenance status
	v3.7	maintained till 3.7.10, then now EOL
	v3.8	maintained till 3.8.13, then now EOL
	v3.9	maintained till 3.9.11, then now EOL
√	v3.10	longterm stable (3.10.88), kept maintained
	v3.11	maintained till 3.11.10, then now EOL
	v3.12	longterm stable (3.12.47), kept maintained (by SUSE)
	v3.13	maintained till 3.13,11 then now EOL
√	v3.14	longterm stable (3.14.52), kept maintained
	v3.15	maintained till 3.15.10, then now EOL
	v3.16	maintained till 3.16.7, then now EOL
	v3.17	maintained till 3.17.8, then now EOL
	v3.18	longterm stable (3.18.21), kept maintained (by Debian)
	v3.19	maintained till 3.19.8, then now EOL
	v4.0	stable release (4.1.9), till 4.2 release
√	v4.1	next longterm stable version (4.1.7)

Stable release include MUST APPLY essential fixes

version	fixes
v3.7 -> v3.7.10	718
v3.8 -> v3.8.13	996
v3.9 -> v3.9.11	746
v3.10 -> v3.10.88	4,849
v3.11 -> v3.11.10	677
v3.12 -> v3.12.47	5,235
v3.13 -> v3.13.11	903
v3.14 -> v3.14.52	3,765
v3.15 -> v3.15.10.	703
v3.16 -> v3.16.7	871
v3.17 -> v3.17.8	884
v3.18 -> v3.18.21	2,114
v3.19 -> v3.19.8	873
v4.0 -> v4.0.9	757
v4.1 -> v4.1.7	697

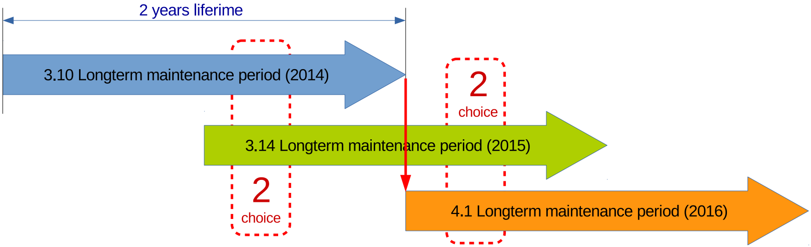
Rule : document/stable_kernel_rules.txt ->

- It must be obviously correct and tested.
- It cannot be bigger than 100 lines, with context.
- It must fix only one thing.
- It must fix a real bug that bothers people (not a, "This could be a problem..." type thing).
- It must fix a problem that causes a build error (but not for things marked CONFIG_BROKEN), an oops, a hang, data corruption, a real security issue, or some "oh, that's not good" issue. In short, something critical.
- Serious issues as reported by a user of a distribution kernel may also be considered if they fix a notable performance or interactivity issue. As these fixes are not as obvious and have a higher risk of a subtle regression they should only be submitted by a distribution kernel maintainer and include an addendum linking to a bugzilla entry if it exists and additional information on the user-visible impact.
- New device IDs and quirks are also accepted.
- No "theoretical race condition" issues, unless an explanation of how the race can be exploited is also provided.
- It cannot contain any "trivial" fixes in it (spelling changes, whitespace cleanups, etc).
- It must follow Documentation/SubmittingPatches rules.
- It or an equivalent fix must already exist in Linus' tree (upstream).

Longterm stable (LTS) kernel release cadence

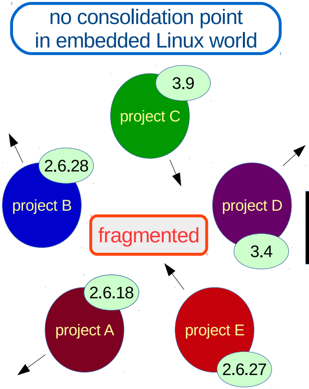
Target kernel selection rules

- Maintainer will **choose one LTS version per year**
- **Maintain it for 2 years** from its original release
- LTS-3.10 becomes EOL when LTS-4.1 is released
- Then, we have 2 LTS kernels versions like 3.14 and 4.1

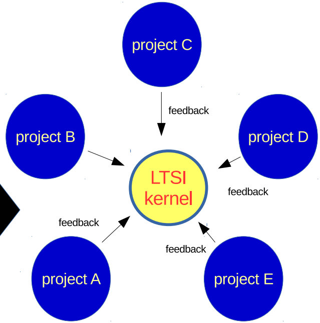

2 years liferime
3.10 Longterm maintenance period (2014)
2
choice
3.14 Longterm maintenance period (2015)
2
choice
4.1 Longterm maintenance period (2016)

LTSI kernel concept

Creating common ground for embedded industry



periodic LTS release
(one version per year)



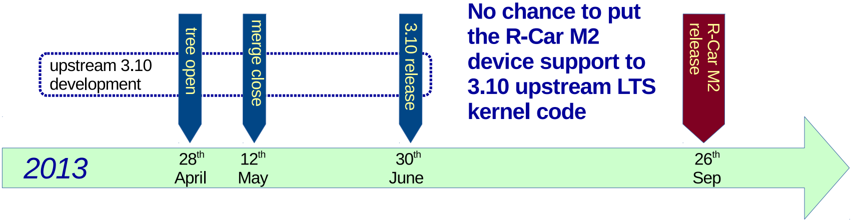
Before LTSI start, each embedded project picked random kernel version.

share experience within industries

- bug case information and workaround
- missing feature (error recovery)
- feature enhancement
- research and trial report

LTSI mechanism helps SoC vender a lot

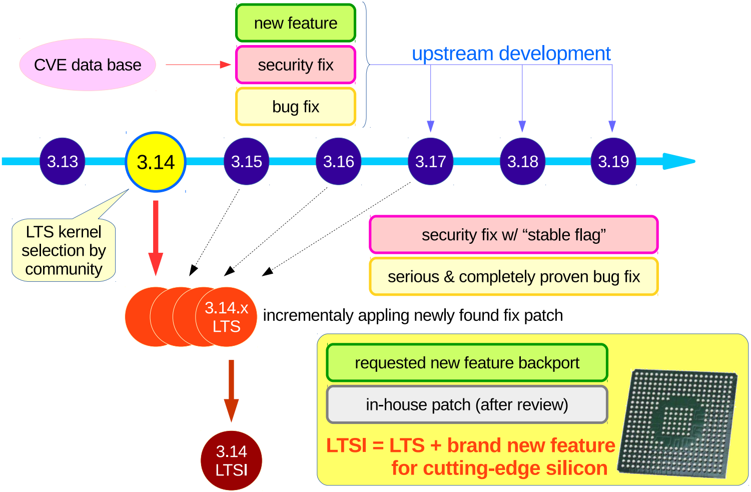
Renesas developed and delivered R-Car gen2 Linux BSP with LTSI-3.10 kernel. Here is the history of upstream 3.10 development and device release.



No chance to put the R-Car M2 device support to 3.10 upstream LTS kernel code

LTSI kernel was the only option to backport mainlined code

Seems some people confuse LTS and LTSI difference


new feature
CVE data base
security fix
bug fix
upstream development
3.13
3.14
3.15
3.16
3.17
3.18
3.19
LTS kernel selection by community
security fix w/ “stable flag”
serious & completely proven bug fix
3.14.x LTS
incrementaly appling newly found fix patch
requested new feature backport
in-house patch (after review)
3.14 LTSI

LTSI = LTS + brand new feature for cutting-edge silicon

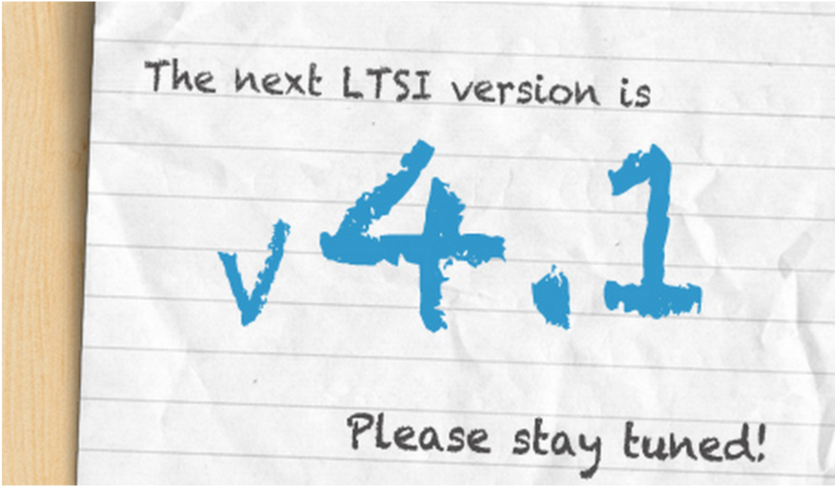
LTSI Status Update (3.14 and 4.1)

LTSI 3.14 is the latest release @ January 9, 2015

LTSI 3.14 development history (result)

item	date
kernel 3.14 merge window open	2014.1.19
kernel 3.14 merge window close	2014.2.2
kernel 3.14 release	2014.3.30
Announce of 2014 LTS kernel version	2014.5.20 (LinuxCon JP)
LTSI-3.14 git tree created	2014.5.20
3.14 becomes LTS (=3.16 release)	2014.8.3
LTSI-3.14 merge window open	2014.8.23
patch collection period	84 days
LTSI-3.14-rc1 (=merge window close)	2014.11.14
validation period	56 days
LTSI-3.14 release	2015.1.9

Greg.K.H announced next LTS is 4.1 and LTSI follow it



The next LTSI version is

v4.1

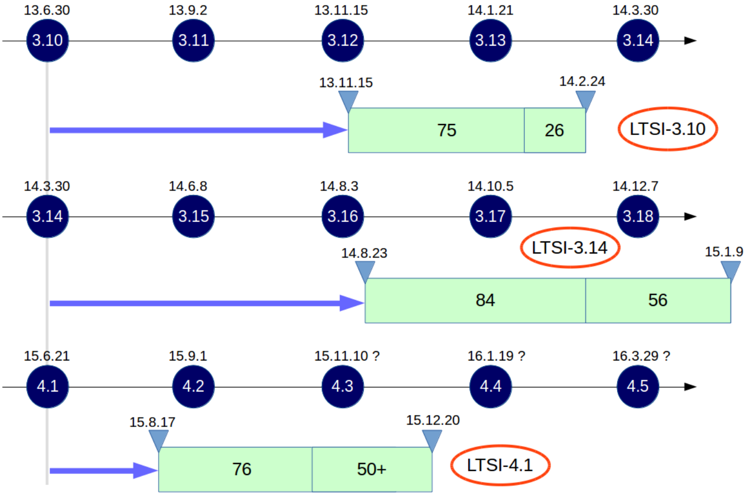
Please stay tuned!

LTSI 4.1 development schedule (merge close soon)

item	date
kernel 4.1 merge window open	2015.4.12
kernel 4.1 merge window close	2015.4.26
kernel 4.1 release	2015.6.21
Greg announced 4.1 is next LTS(I)	LinuxCon NA 2015
LTSI-4.1 merge window open	2015.8.17
patch collection period	76 days
LTSI-4.1 merge window close (target)	2015.10.31
validation period	50+ days
LTSI-4.1 release (target)	2015.12.20

Compare to the previous, we adopted relatively fresh kernel

Getting closer to the latest community kernel



Patch submission for LTSI-4.1 (as of today)

Altera SoCFPGA (4.2 -> 4.1.6 backport)

- Support for Arria10 platform
- Support of EDAC driver
- Support for Suspend-to-RAM

Toshiba (planing)

- flush file system related

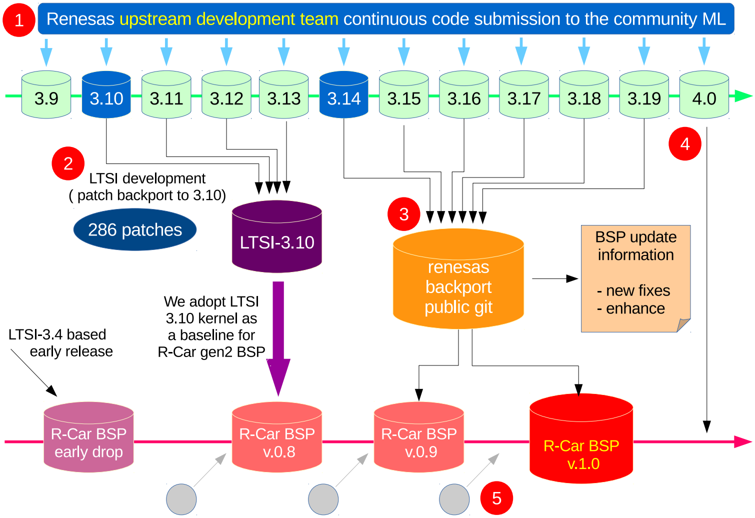
Renesas SoCs (4.2 -> 4.1.6 backport)

- ak4642
- cpg
- gpio-rcar
- i2c-rcar
- i2c-sh-mobile
- irq-renesas-intc-irqpin
- irq-renesas-irqc
- mach-shmobile
- phy-rcar-gen2
- phy-rcar-gen2-usb
- pinctrl/sh-pfc
- rcar-dmac
- rcar-du/snd/vin

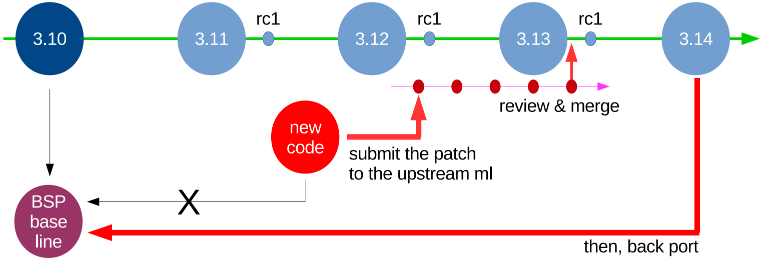
BSP development w/LTSI kernel

Renesas R-Car generation2 SoC case

R-Car gen2 Linux BSP whole development process

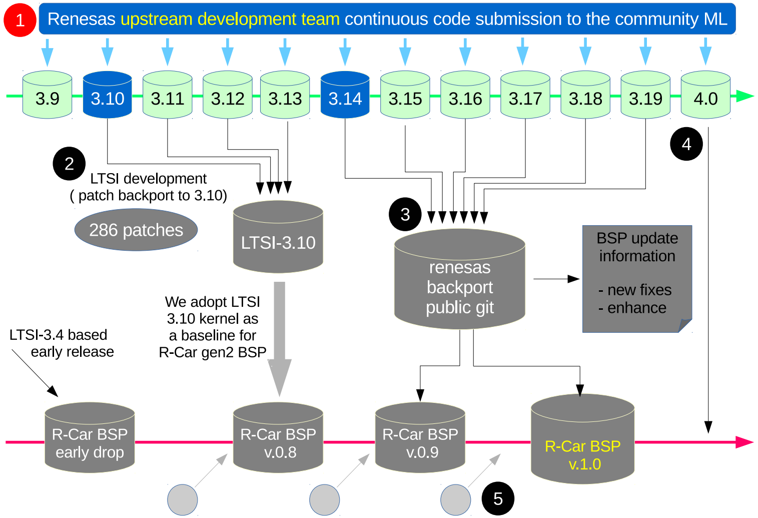

1
Renesas upstream development team continuous code submission to the community ML
3.9
3.10
3.11
3.12
3.13
3.14
3.15
3.16
3.17
3.18
3.19
4.0
2
LTSI development
(patch backport to 3.10)
286 patches
LTSI-3.10
3
renesas backport public git
BSP update information
- new fixes
- enhance
4
LTSI-3.4 based early release
We adopt LTSI 3.10 kernel as a baseline for R-Car gen2 BSP
R-Car BSP early drop
R-Car BSP v.0.8
R-Car BSP v.0.9
R-Car BSP v.1.0
5

Initially, Renesas adopts upstream first strategy

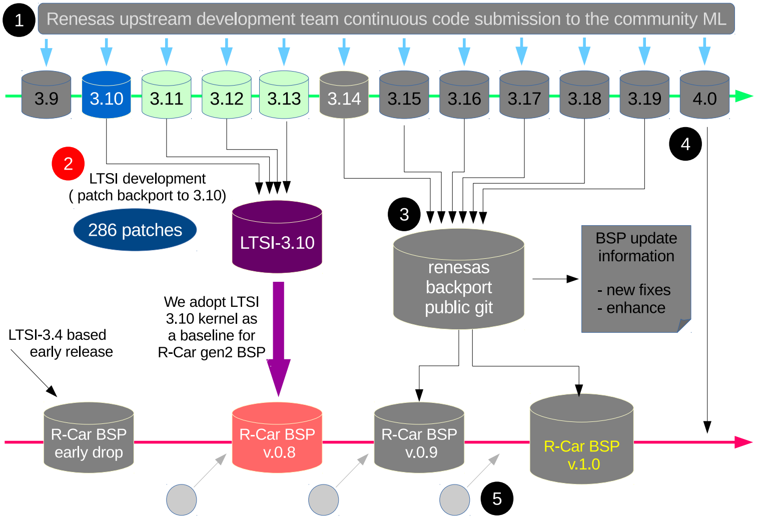


pros	cons
- clean code (reviewed by the community) - coordinated with existing code - merged to the upstream code - no need to keep in-house code	- take time (roughly 6 month) - might need iterative approach (bit b bit) - might need code adjustment

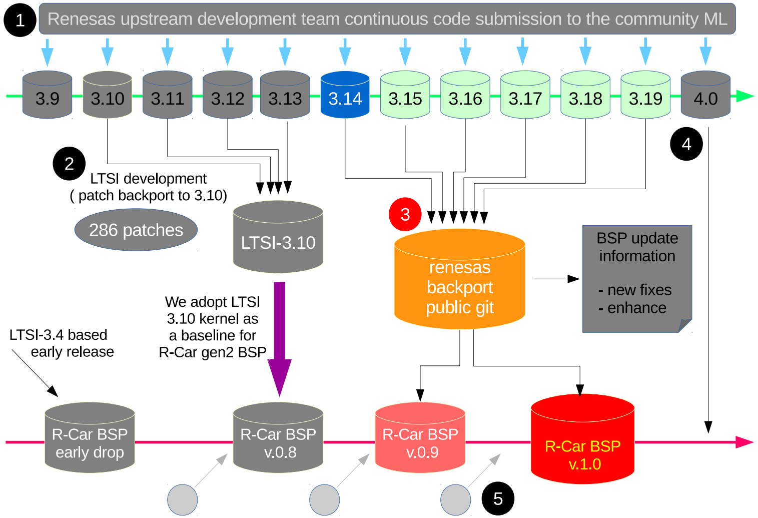
(1) Upstream development (contentious effort)



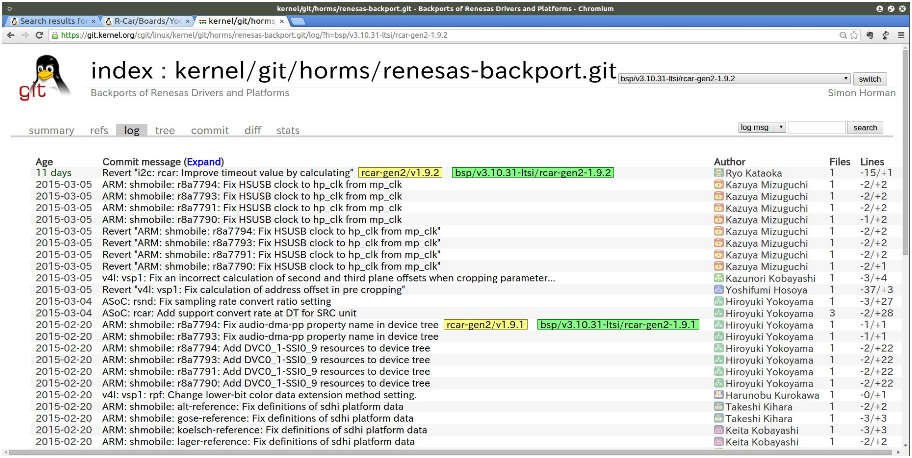
(2) Pick a LTSI-3.10 kernel as a baseline



(3) continuous backport after LTSI merge closed


1
Renesas upstream development team continuous code submission to the community ML
3.9
3.10
3.11
3.12
3.13
3.14
3.15
3.16
3.17
3.18
3.19
4.0
2
LTSI development
(patch backport to 3.10)
286 patches
LTSI-3.10
3
renesas backport public git
BSP update information
- new fixes
- enhance
4
We adopt LTSI 3.10 kernel as a baseline for R-Car gen2 BSP
LTSI-3.4 based early release
R-Car BSP early drop
R-Car BSP v.0.8
R-Car BSP v.0.9
R-Car BSP v.1.0
5

(3) continuous backport after LTSI merge closed


kernel/git/horms/renesas-backport.git - Backports of Renesas Drivers and Platforms - Chromium
Search results fo ×
R-Car/Boards/Yo ×
kernel/git/horms ×
https://git.kernel.org/cgit/linux/kernel/git/horms/renesas-backport.git/log/?h=bsp/v3.10.31-ltsi/rcar-gen2-1.9.2
git
bsp/v3.10.31-ltsi/rcar-gen2-1.9.2
switch
summary
refs
log
tree
commit
diff
stats
log msg
search

index : kernel/git/horms/renesas-backport.git

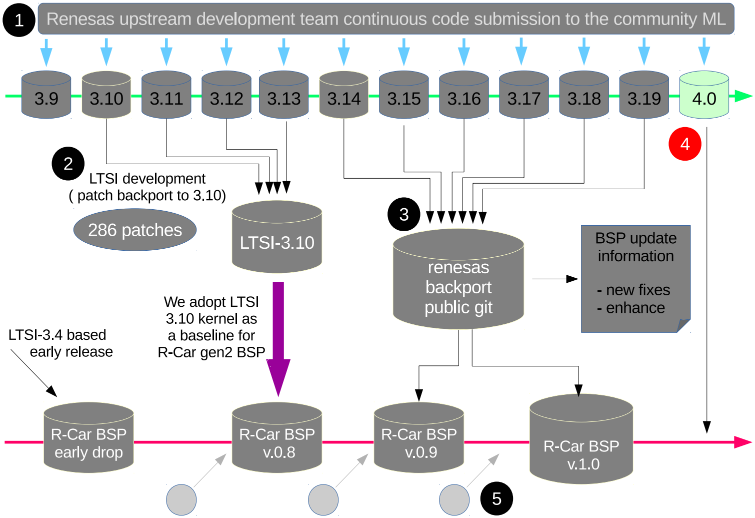
Backports of Renesas Drivers and Platforms

Simon Horman

Age	Commit message (Expand)	Author	Files	Lines
11 days	Revert "i2c: rcar: Improve timeout value by calculating" rcar-gen2/v1.9.2 bsp/v3.10.31-ltsi/rcar-gen2-1.9.2	Ryo Kataoka	1	-15/+1
2015-03-05	ARM: shmobile: r8a7794: Fix HSUSB clock to hp_clk from mp_clk	Kazuya Mizuguchi	1	-2/+2
2015-03-05	ARM: shmobile: r8a7793: Fix HSUSB clock to hp_clk from mp_clk	Kazuya Mizuguchi	1	-2/+2
2015-03-05	ARM: shmobile: r8a7791: Fix HSUSB clock to hp_clk from mp_clk	Kazuya Mizuguchi	1	-2/+2
2015-03-05	ARM: shmobile: r8a7790: Fix HSUSB clock to hp_clk from mp_clk	Kazuya Mizuguchi	1	-1/+2
2015-03-05	Revert "ARM: shmobile: r8a7794: Fix HSUSB clock to hp_clk from mp_clk"	Kazuya Mizuguchi	1	-2/+2
2015-03-05	Revert "ARM: shmobile: r8a7793: Fix HSUSB clock to hp_clk from mp_clk"	Kazuya Mizuguchi	1	-2/+2
2015-03-05	Revert "ARM: shmobile: r8a7791: Fix HSUSB clock to hp_clk from mp_clk"	Kazuya Mizuguchi	1	-2/+2
2015-03-05	Revert "ARM: shmobile: r8a7790: Fix HSUSB clock to hp_clk from mp_clk"	Kazuya Mizuguchi	1	-2/+1
2015-03-05	v4l: vsp1: Fix an incorrect calculation of second and third plane offsets when cropping parameter...	Kazunori Kobayashi	1	-3/+4
2015-03-05	Revert "v4l: vsp1: Fix calculation of address offset in pre cropping"	Yoshifumi Hosoya	1	-37/+3
2015-03-04	ASoC: rsnd: Fix sampling rate convert ratio setting	Hiroyuki Yokoyama	1	-3/+27
2015-03-04	ASoC: rcar: Add support convert rate at DT for SRC unit	Hiroyuki Yokoyama	3	-2/+28
2015-02-20	ARM: shmobile: r8a7794: Fix audio-dma-pp property name in device tree rcar-gen2/v1.9.1 bsp/v3.10.31-ltsi/rcar-gen2-1.9.1	Hiroyuki Yokoyama	1	-1/+1
2015-02-20	ARM: shmobile: r8a7793: Fix audio-dma-pp property name in device tree	Hiroyuki Yokoyama	1	-1/+1
2015-02-20	ARM: shmobile: r8a7794: Add DVC0_1-SSI0_9 resources to device tree	Hiroyuki Yokoyama	1	-2/+22
2015-02-20	ARM: shmobile: r8a7793: Add DVC0_1-SSI0_9 resources to device tree	Hiroyuki Yokoyama	1	-2/+22
2015-02-20	ARM: shmobile: r8a7791: Add DVC0_1-SSI0_9 resources to device tree	Hiroyuki Yokoyama	1	-2/+22
2015-02-20	ARM: shmobile: r8a7790: Add DVC0_1-SSI0_9 resources to device tree	Hiroyuki Yokoyama	1	-2/+22
2015-02-20	v4l: vsp1: rpf: Change lower-bit color data extension method setting.	Harunobu Kurokawa	1	-0/+1
2015-02-20	ARM: shmobile: alt-reference: Fix definitions of sdhi platform data	Takeshi Kihara	1	-2/+2
2015-02-20	ARM: shmobile: gose-reference: Fix definitions of sdhi platform data	Takeshi Kihara	1	-3/+3
2015-02-20	ARM: shmobile: koelsch-reference: Fix definitions of sdhi platform data	Keita Kobayashi	1	-3/+3
2015-02-20	ARM: shmobile: lager-reference: Fix definitions of sdhi platform data	Keita Kobayashi	1	-2/+2

```
https://git.kernel.org/cgit/linux/kernel/git/horms/renesas-backport.git/log/?h=bsp/
v3.10.31-ltsi/rcar-gen2-1.9.2
```

(4) super-long term security fix adoption



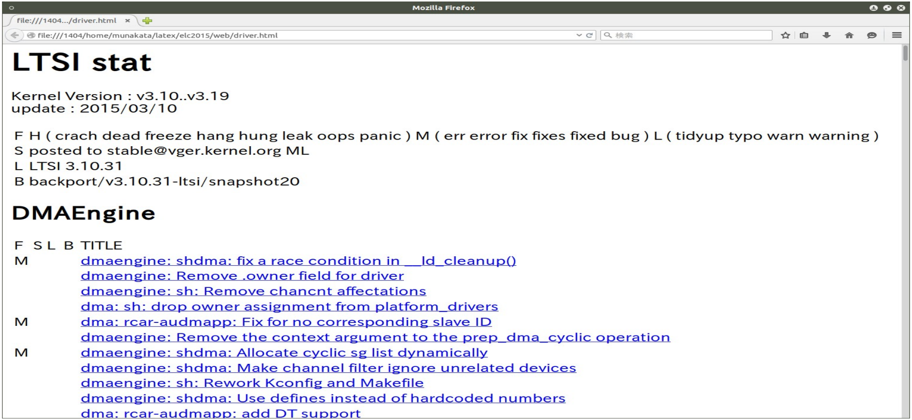
BSP maintenance : new bug-fix patch tracking

We continue check if new fixes is available

- automated upstream patch scan from git
- crawling scope is own code or modified code
- F : patch severity parsing
  - H : crash, dead, freeze, hang, hung, leak, oops, panic
  - M : err, error, fix, fixes, fixed, bug
  - L : tidyup, typo, warn, warning
- S :check if this patch is cc'd to stable@vger.kernel.org
- L :check if this patch is already a part of LTSI kernel
- B :check if this patch is send to renesas-backport git

BSP maintenance : new bug-fix patch tracking

actual scan result example (comparing upstream 3.10..3.19)


Mozilla Firefox
file:///1404.../driver.html
file:///1404/home/munakata/latex/elc2015/web/driver.html
検索

LTSI stat

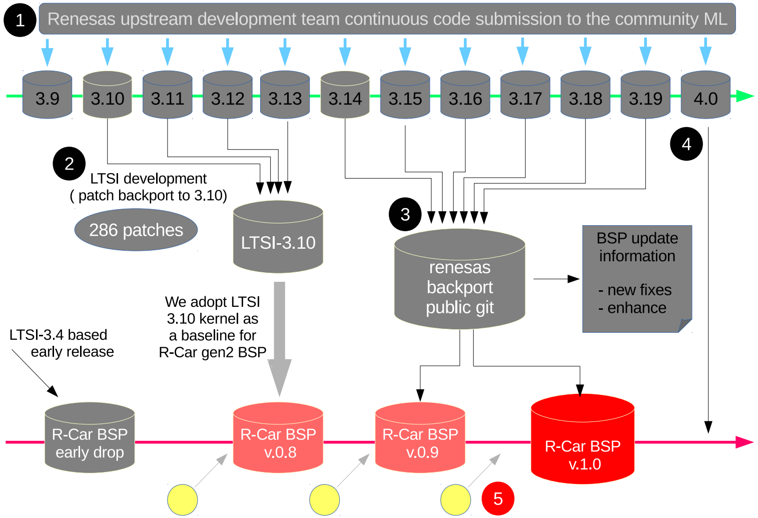
Kernel Version : v3.10..v3.19
update : 2015/03/10

F H (crach dead freeze hang hung leak oops panic) M (err error fix fixes fixed bug) L (tidyup typo warn warning)
S posted to stable@vger.kernel.org ML
L LTSI 3.10.31
B backport/v3.10.31-ltsi/snapshot20

DMAEngine

F	S	L	B	TITLE
M				dmaengine: shdma: fix a race condition in __ld_cleanup()
				dmaengine: Remove .owner field for driver
				dmaengine: sh: Remove chancnt affectations
				dma: sh: drop owner assignment from platform_drivers
M				dma: rcar-audmapp: Fix for no corresponding slave ID
				dmaengine: Remove the context argument to the prep_dma_cyclic operation
M				dmaengine: shdma: Allocate cyclic sg list dynamically
				dmaengine: shdma: Make channel filter ignore unrelated devices
				dmaengine: sh: Rework Kconfig and Makefile
				dmaengine: shdma: Use defines instead of hardcoded numbers
				dma: rcar-audmapp: add DT support

(5) local in-house patch adoption and elimination



(5) local in-house patch adoption and elimination

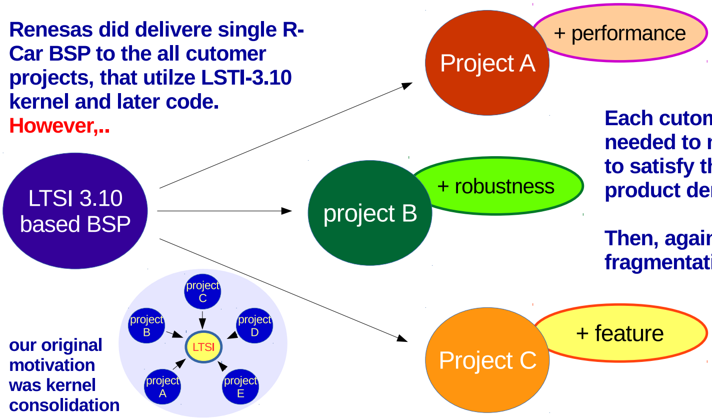
we try to eliminate in-house code from our BSP, however

- Due to time constraint (=up to six months needed for upstream-first attempt), we still need to manage some in-house patch.
- We start up-porting challenge (in-house code to the upstream flow) to eliminate (at least reduce) in-house code.
- It requires an extra code polish to comply with latest mainline kernel patch adoption criteria. But we believe this is the valuable challenges.

Lessons learned form the real life

We noticed each customer modified kernel a lot

Renesas did delivere single R-Car BSP to the all cutomer projects, that utilze LSTI-3.10 kernel and later code.
However,..



LTSI 3.10 based BSP

Project A

+ performance

project B

+ robustness

Project C

+ feature

project C

project B

project D

LTSI

project A

project E

Each cutomer needed to modify to satisfy their product demands.

Then, again it caused fragmentation

our original motivation was kernel consolidation

LTSI-3.10 kernel was not ready to develop product?

upstream kernel is not designed for production?

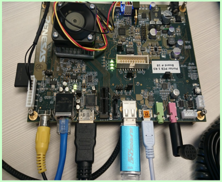
- Driver code must be written portable and unified
- And it can not be tie with specific hw implementation
  - assumption of specific hw feature
  - use-case specific error recovery algorithm
  - allocation of dedicated acceleration engine like DMA
- Security fix done by the community is not good enough?

Maybe we may need distribution like enterprise world

The production kernel needs landing procedure

Distribution = verified collection of various Linux programs (=packages)
per-build binary distribution = Debian, Ubuntu, Cent, Fedora
source code distribution = Gentoo, Open Embedded (yocto),...

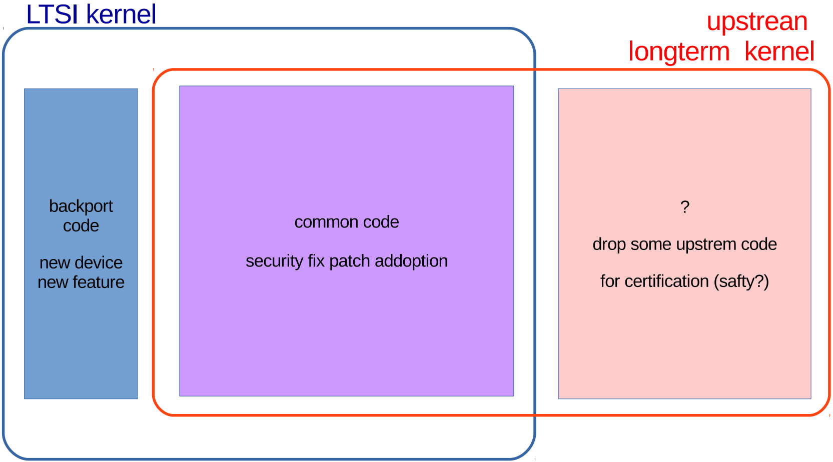
SDK (Software Development Kit) = subset of distribution designed for specific application = Android, Tizen IVI, MPD, Drone Code



BSP (Board Support Package) = subset of SDK, designed for specific target hardware. Embedded SoC vendor develops BSP for their reference platform. Product producer modify BSP to fit with product hw

LTSI enhance

What should be added / dropped for future LTSI?



Conclusion and Resources

Conclusion

Conclusion

- We have LTSI-3.10 and 3.14 for now and start patch collection for next LTSI-4.1 release.
- Version gap between LTS(I) and upstream becomes narrow at 4.1 development. This is a good thing, but...
- We ad opted LTSI-3.10 kernel for the production BSP, and noticed many customers needed to enhance LTSI kernel for various purpose. We want to discuss how can we improve the usability of LTSI kernel.

Resources

Resources = ltsi.linuxfoundation.org

- LTSI process document (new) = http://ltsi.linuxfoundation.org/participate-in-ltsi/ltsi-development-guide
- ML
  - ML subscription = https://lists.linuxfoundation.org/mailman/listinfo/ltsi-dev
  - ML archives = http://lists.linuxfoundation.org/pipermail/ltsi-dev/
  - ML patchwork = https://patchwork.kernel.org/project/ltsi-dev/list/
- git(each patch) = http://git.linuxfoundation.org/?p=ltsi-kernel.git;a=summary
- download (tar ball) = http://ltsi.linuxfoundation.org/downloads/releases
- twitter = @LinuxLTSI
- document archives = http://ltsi.linuxfoundation.org/resources