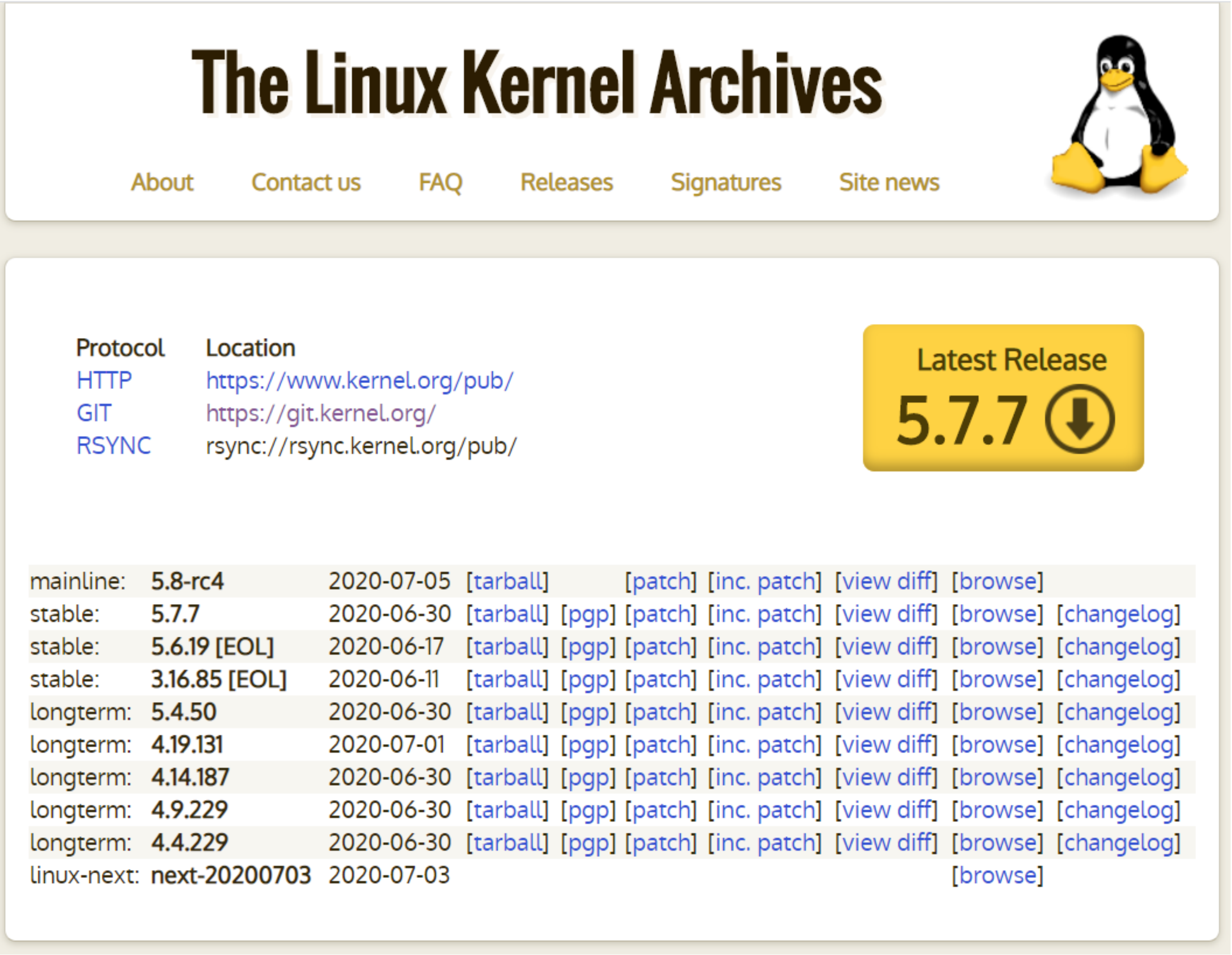
ネットワーク接続やクラウド連携が必須となった現代の電子機器の開発ではオープンソースソフトウエア（＝OSS）の活用は避けて通れません。しかし元々PC 向けの汎用 CPU で動作するように作られた OSS を高機能 SOC に移植するには SOC に内蔵された各種周辺機能を動かすためのデバイスドライバーや OS が持つ表示／音声調停メカニズムとの合わせ込みが必要になります。しかしこれらの開発について半導体サプライヤーとセットメーカーの役割分担の線引きが曖昧なので実際にはベンダーによって「OSS への対応レベル（＝品質）」には大きな差があります。今回はルネサスが考える「OSS を使った製品開発の成功戦略」をご紹介します。

製品開発における「ソフトウエア開発コストの急拡大」は多くのセットメーカーの頭痛の種です。商品企画毎にゼロからソフトウエアを作るという伝統的なやり方は破綻したと考えて良いでしょう。解決策としてアプリケーションソフトを同一世代の品種間や複数世代間で出来るだけ使い回していく「ソフトウエアファースト」の考え方が提唱されていますが、ここで重要となるのは長期間に亘って利用できる「サスティナブルなソフトウエア」という考え方です。これまでソフトウエアは特定のユースケース（＝絞り込まれた前提条件）を拠り所に開発／検証されてきたのですが SW を使い回していくには「将来発生するかもしれない未知の使い方」にも予め対応しておく必要があります。この検証手段のひとつに故意に間違ったデータをプログラムに喰わせてみる fuzz test があります。例えば表計算ファイルの拡張子を無理矢理 .mp4 に変更してメディアプレーヤに喰わせてみたりします。メディアプレーヤがフリーズせずに「正しい動画ファイルではありません」とエラー表示するのが期待値ですが、特定用途向けに開発された組み込み機器の中にはこの種の意地悪テストに耐えられないものもあります。

OSS は元来誰がどんな使い方をしても破綻しないようにプログラムされていて（いるべきで）、その開発途上でのオープンなピアレビュー履歴も閲覧可能です。このような特性から OSS には長期運用に耐える品質が期待できるでしょう。（対象製品のユースケースに対する動作検証は依然としてセットメーカーの責任である事は言うまでもありませんが）もうひとつ OSS が特定企業に帰属しない公知資産であることも長期運用観点でアドバンテージです。例えば Linux kernel は登場から既に四半世紀以上経過しましたが、現在でも創始者のリーナス氏が率いる開発コミュニティによって継続的で精力的な機能拡張や品質改善が行なわれています。企業に帰属するソフトウエアに類似のものはひとつもありません。



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

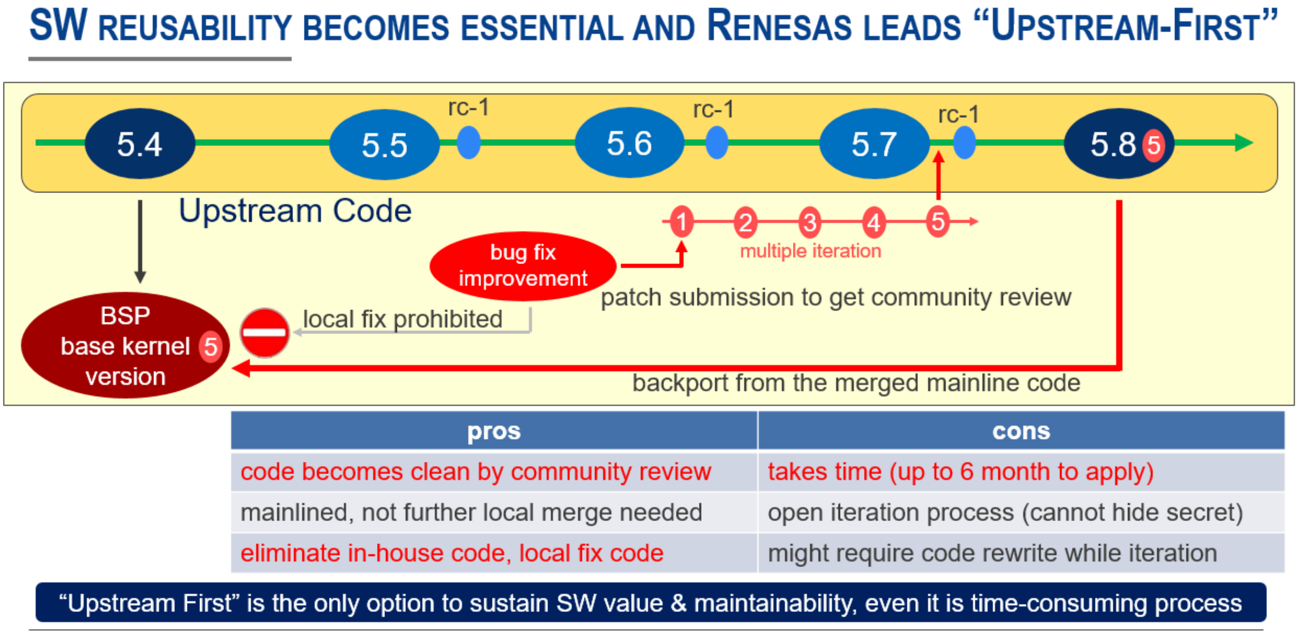
mainline:	5.8-rc4	2020-07-05	[tarball]		[patch]	[inc. patch]	[view diff]	[browse]	
stable:	5.7.7	2020-06-30	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
stable:	5.6.19 [EOL]	2020-06-17	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
stable:	3.16.85 [EOL]	2020-06-11	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.4.50	2020-06-30	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.19.131	2020-07-01	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.14.187	2020-06-30	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.9.229	2020-06-30	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.4.229	2020-06-30	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
linux-next:	next-20200703	2020-07-03						[browse]	

OSS の代表、Linux kernel コミュニティでは約 10 週間毎に最新バージョンをリリース

https://www.kernel.org/

OSS 本体は原則無償配布という事情もあって半導体サプライヤーの OSS への取り組み姿勢は各社まちまちなのですが、ルネサスは比較的早い時期から積極的に取り組んできました。その歴史は家庭にインターネットが導入され始めた 1995 年頃まで遡ります。当時私たちは開発コミュニティからダウンロードしたコードに自社 RISC マイコンが動作するように独自改造加えたものをお客様に配布していました。その後 CELF（CE Linux Forum）や Linux Foundation といった OSS の活用を目指した産業コンソーシアム活動や国際的な開発者会議などに参加する機会が増えて開発コミュニティの重鎮から直接 OSS との正しい付き合い方についてアドバイスを受ける事ができました。そこで学んだのは「ダウンロードしたものを独自改造するのではなく、コミュニティの一員となって OSS の継続的な開発サイクルに参加し、長期にわたって活用可能な SW 資産を構築する」という考え方です。これを実践するためルネサスはコミュニティで活躍中の現役バリバリのカーネルデベロッパー数名と契約し、後追いではなくコミュニティの開発サイクルの中でルネサスのデバイスや評価ボード向けのコードを開発／検証していく「Upstream First」と呼ばれる開発プロセスを導入しました。現在お客さまに提供している

Linux BSP （Board Support Package）は全てこの「Upstream First」に基づいて開発されているのでコミュニティが配布するどのバージョンにもルネサスのデバイスを動かす検証済みのコードが含まれています。



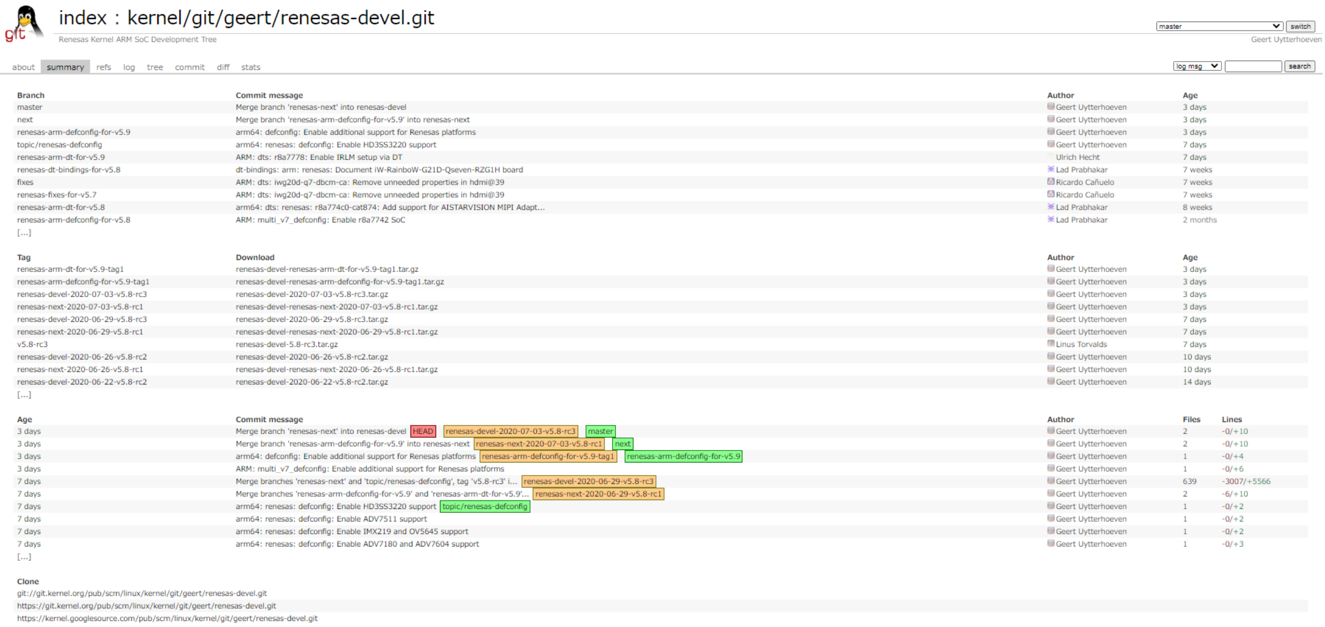
SW REUSABILITY BECOMES ESSENTIAL AND RENESAS LEADS “UPSTREAM-FIRST”

pros	cons
code becomes clean by community review	takes time (up to 6 month to apply)
mainlined, not further local merge needed	open iteration process (cannot hide secret)
eliminate in-house code, local fix code	might require code rewrite while iteration

“Upstream First” is the only option to sustain SW value & maintainability, even it is time-consuming process

Sustainable SW の基盤となる Upstream First の考え方を全面的に適用しています

さらに BSP を構築／配布するプラットフォームを開発する 「yocto project」や車載向けの Linux 適用拡大を目指した 「AGL」 「Genivi」 といった産業コンソーシアム活動においてもルネサスは重要な役割を担っています。またサスティナブルなソフトウエア実現の取り組みとして Linux の長期運用を目指している CIP（Civil Infrastructure Platform） や LTSI（Long Term Support Initiative） といった活動もリーディングしており、当社が配布する Linux BSP は全てコミュニティが指定する長期メンテナンス対象バージョンを採用していますが、これらの取り組みは「お客さまの SW 投資を保全」するために極めて重要かつ有効と考えています。ルネサスは今後も開発コミュニティや産業コンソーシアム活動と連携を通して「SW の長期運用」 という産業界の期待に更に応えていきます。



index : kernel/git/geert/renesas-devel.git

Renesas Kernel ARM SoC Development Tree

Geert Uytterhoeven

Branch	Commit message	Author	Age
master	Merge branch 'renesas-next' into renesas-devel	Geert Uytterhoeven	3 days
next	Merge branch 'renesas-arm-defconfig-for-v5.9' into renesas-next	Geert Uytterhoeven	3 days
renesas-arm-defconfig-for-v5.9	arm64: defconfig: Enable additional support for Renesas platforms	Geert Uytterhoeven	3 days
topic/renesas-defconfig	arm64: renesas: defconfig: Enable HD3SS3220 support	Geert Uytterhoeven	7 days
renesas-arm-dt-for-v5.9	ARM: dts: r8a7778: Enable IRLM setup via DT	Ulrich Hecht	7 days
renesas-dt-bindings-for-v5.8	dt-bindings: arm: renesas: Document iW-RainboW-G21D-Qseven-RZG1H board	Lad Prabhakar	7 weeks
fixes	ARM: dts: iwg20d-q7-dbcm-ca: Remove unneeded properties in hdmi@39	Ricardo Cañuelo	7 weeks
renesas-fixes-for-v5.7	ARM: dts: iwg20d-q7-dbcm-ca: Remove unneeded properties in hdmi@39	Ricardo Cañuelo	7 weeks
renesas-arm-dt-for-v5.8	arm64: dts: renesas: r8a774c0-cat874: Add support for AISTARVISION MIPI Adapt...	Lad Prabhakar	8 weeks
renesas-arm-defconfig-for-v5.8	ARM: multi_v7_defconfig: Enable r8a7742 SoC	Lad Prabhakar	2 months

[...]

Tag	Download	Author	Age
renesas-arm-dt-for-v5.9-tag1	renesas-devel-renesas-arm-dt-for-v5.9-tag1.tar.gz	Geert Uytterhoeven	3 days
renesas-arm-defconfig-for-v5.9-tag1	renesas-devel-renesas-arm-defconfig-for-v5.9-tag1.tar.gz	Geert Uytterhoeven	3 days
renesas-devel-2020-07-03-v5.8-rc3	renesas-devel-2020-07-03-v5.8-rc3.tar.gz	Geert Uytterhoeven	3 days
renesas-next-2020-07-03-v5.8-rc1	renesas-devel-renesas-next-2020-07-03-v5.8-rc1.tar.gz	Geert Uytterhoeven	3 days
renesas-devel-2020-06-29-v5.8-rc3	renesas-devel-2020-06-29-v5.8-rc3.tar.gz	Geert Uytterhoeven	7 days
renesas-next-2020-06-29-v5.8-rc1	renesas-devel-renesas-next-2020-06-29-v5.8-rc1.tar.gz	Geert Uytterhoeven	7 days
v5.8-rc3	renesas-devel-5.8-rc3.tar.gz	Linus Torvalds	7 days
renesas-devel-2020-06-26-v5.8-rc2	renesas-devel-2020-06-26-v5.8-rc2.tar.gz	Geert Uytterhoeven	10 days
renesas-next-2020-06-26-v5.8-rc1	renesas-devel-renesas-next-2020-06-26-v5.8-rc1.tar.gz	Geert Uytterhoeven	10 days
renesas-devel-2020-06-22-v5.8-rc2	renesas-devel-2020-06-22-v5.8-rc2.tar.gz	Geert Uytterhoeven	14 days

[...]

Age	Commit message	Author	Files	Lines
3 days	Merge branch 'renesas-next' into renesas-devel HEAD renesas-devel-2020-07-03-v5.8-rc3 master	Geert Uytterhoeven	2	-0/+10
3 days	Merge branch 'renesas-arm-defconfig-for-v5.9' into renesas-next renesas-next-2020-07-03-v5.8-rc1 next	Geert Uytterhoeven	2	-0/+10
3 days	arm64: defconfig: Enable additional support for Renesas platforms renesas-arm-defconfig-for-v5.9-tag1 renesas-arm-defconfig-for-v5.9	Geert Uytterhoeven	1	-0/+4
3 days	ARM: multi_v7_defconfig: Enable additional support for Renesas platforms	Geert Uytterhoeven	1	-0/+6
7 days	Merge branches 'renesas-next' and 'topic/renesas-defconfig', tag 'v5.8-rc3' i... renesas-devel-2020-06-29-v5.8-rc3	Geert Uytterhoeven	639	-3007/+5566
7 days	Merge branches 'renesas-arm-defconfig-for-v5.9' and 'renesas-arm-dt-for-v5.9'... renesas-next-2020-06-29-v5.8-rc1	Geert Uytterhoeven	2	-6/+10
7 days	arm64: renesas: defconfig: Enable HD3SS3220 support topic/renesas-defconfig	Geert Uytterhoeven	1	-0/+2
7 days	arm64: renesas: defconfig: Enable ADV7511 support	Geert Uytterhoeven	1	-0/+2
7 days	arm64: renesas: defconfig: Enable IMX219 and OV5645 support	Geert Uytterhoeven	1	-0/+2
7 days	arm64: renesas: defconfig: Enable ADV7180 and ADV7604 support	Geert Uytterhoeven	1	-0/+3

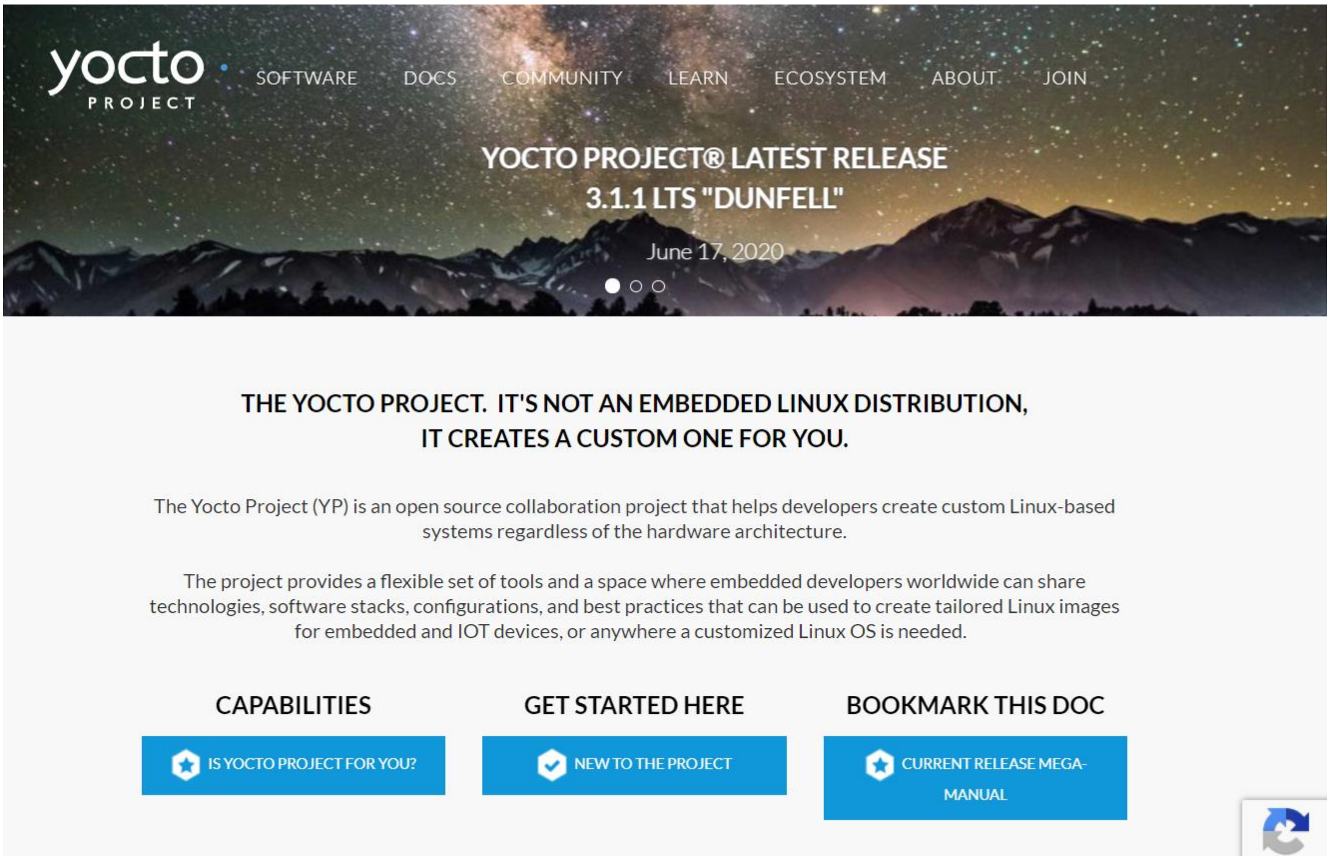
[...]

Clone

git://git.kernel.org/pub/scm/linux/kernel/git/geert/renesas-devel.git
https://git.kernel.org/pub/scm/linux/kernel/git/geert/renesas-devel.git
https://kernel.googlesource.com/pub/scm/linux/kernel/git/geert/renesas-devel.git

Renesas R-Car 向けの Linux kernel の公開リポジトリ（最新版 kernel 対応を含め逐次更新している）

https://git.kernel.org/pub/scm/linux/kernel/git/geert/renesas-devel.git/



YOCTO PROJECT® LATEST RELEASE
3.1.1 LTS "DUNFELL"

June 17, 2020

THE YOCTO PROJECT. IT'S NOT AN EMBEDDED LINUX DISTRIBUTION, IT CREATES A CUSTOM ONE FOR YOU.

The Yocto Project (YP) is an open source collaboration project that helps developers create custom Linux-based systems regardless of the hardware architecture.

The project provides a flexible set of tools and a space where embedded developers worldwide can share technologies, software stacks, configurations, and best practices that can be used to create tailored Linux images for embedded and IOT devices, or anywhere a customized Linux OS is needed.

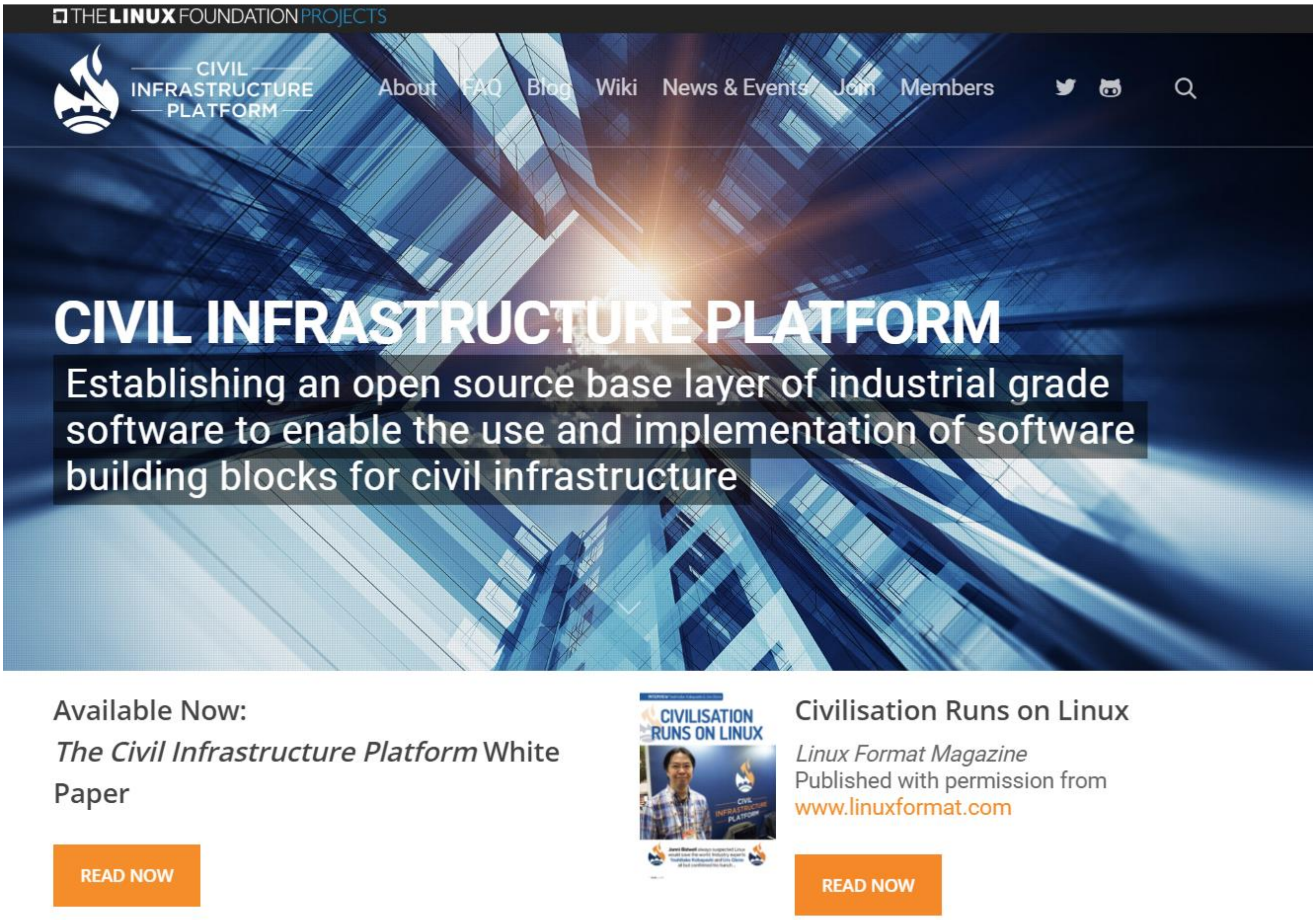
CAPABILITIES

GET STARTED HERE

BOOKMARK THIS DOC

Renesas は Linux BSP 構築ツールを開発する yocto project の Advisory Board メンバー

https://www.yoctoproject.org/



CIVIL INFRASTRUCTURE PLATFORM

Establishing an open source base layer of industrial grade software to enable the use and implementation of software building blocks for civil infrastructure

Available Now:
The Civil Infrastructure Platform White Paper

Civilisation Runs on Linux

Linux Format Magazine
Published with permission from
www.linuxformat.com

社会インフラ向けに Linux の長期運用を目指す CIP 、Renesas もコアメンバーとして活動を支援中

https://www.cip-project.org/

The utilization of open source software (=OSS) is inevitable in the development of modern electronic devices that require a network connection and cloud cooperation. However, to port an OSS created to operate on a general-purpose PC CPU to a high-performance SOC, it is necessary to develop device drivers for on-chip IP functions and a display/voice arbitration mechanism integration. However, the division of roles of such development between semiconductor suppliers and set manufacturers is ambiguous. Reality is that a significant difference exists in the level of OSS compliance (= quality), depending on the silicon vendor. I introduce Renesas's "success strategy for product development using OSS" in this article.

"Rapidly increasing software development costs" becomes a headache for many set makers. The traditional software development method that develops SW from scratch for each product plan has already failed. As a way to remedy the situation, the "Software first" scheme has proposed that reuse the SW as much for the product variant in the same generation and across the multiple generations. To realize that, "sustainable software" will be an essential requirement. So far, software development and verification rely on specific use cases (= narrowed down preconditions). However, to create reusable SW, you need to design also for unknown usage that may occur in the future in advance. The fuzz test is the new SW validation method for such requirements, which intentionally feeds the wrong data to the target application. For example, try changing the spreadsheet file's extension to .mp4 and feed it to the media player. The expected behavior is that the media player outputs an error message saying, "It is not the correct video file" without freezing. However, embedded devices specially developed for specific purposes might not withstand this kind of mean test.

OSS is (should be) programmed so that it would not fail no matter who used it, and its peer review history in a public mailing list is also open to anyone. You can expect that OSS owns enough quality that can withstand long-term operation due to such characteristics. (The operation verification for use cases of target products is still the responsibility of the set manufacturer, though.) From the long-term operation stands point, the fact that OSS is a public asset, not belong to any company, can be another advantage. For example, the first Linux kernel has created more than a quarter of a century ago. Even now, the development community led by its founder, Mr. Linus Torvalds, continues to make active enhancements and quality improvements. There is no similar software belonging to a company.

Because OSS itself is distributed free of charge in principle, semiconductor suppliers have different attitudes towards OSS. And Renesas actively supported OSS since relatively early times. Its history goes back to around 1995 when the Internet came into the home. Back in those days, we downloaded the code from the development community repository and modify it to work with our 32bit RISC microcomputer, then distributed it to the customer. After that, we had opportunities to participate in industrial consortium activities aiming to utilize OSS such as CELF (CE Linux Forum) and Linux Foundation. Also, we attended some international developer conferences. Then we got the wisdom from the development community leaders regarding the right way to work with OSS. The idea was "building long-term sustainable SW assets by participating in the OSS continuous development cycle as a member of the community, rather than modifying the downloaded ones locally." To realize this, we contracted with several active kernel developers. We asked them to develop/verify the Renesas devices and board support code inside the community development cycle rather than in-house activity. With this practice, we could realize the "Upstream First" process. As all Linux BSP that we currently offer our customers is developed based on "Upstream First," so you can find verified Renesas devices support code in any version distributed by the community.

Renesas also plays an essential role in the industrial consortium such as the "yocto project," which develops a platform for building/distributing Linux BSP (Board Support Package), and "AGL" and "Genivi" which expand the Linux adoption in the automotive. We also strive for the long-term operation of Linux, such as CIP (Civil Infrastructure Platform) and LTSI (Long Term Support Initiative) to realize sustainable software. And we adopt the community's long-term maintenance target version to all our Linux BSP. We believe that these efforts are critical and beneficial ways for "protecting customers' SW investment." To satisfy further industry expectations of the "long-term operation of SW," Renesas commit the future by the collaboration with OSS development community and industrial consortium.