



Qt for MCUs

Ultimate Performance. Tiny Footprint.

最小の資源で究極のパフォーマンス

www.qt.io/qt-for-mcu

商品化における厳しい要求仕様の増加

現在、IoT機器、スマートコネクテッドデバイスの出荷数量は大きく伸びていますが、それらのデバイスの多くは商品化に当たり、以下の要求仕様の達成が必須となりつつあります。



リアルタイム処理



低消費電力

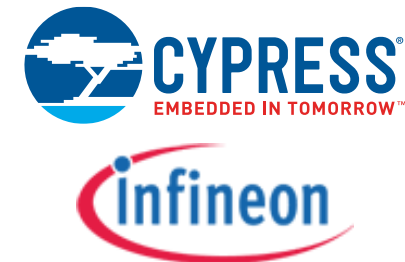


高速起動



部品コスト低減

32bit MCUs with 2Dグラフィックアクセラレータ



マイコン向けソフトウェア開発における課題

現行の課題

1 ビジュアルエディタのみによる制限的な画面開発
プログラミングによる拡張的なUI開発を実現するAPIの欠如

2 固定的なレイアウト
画面サイズ、縦横比の変更にはコードの書き直しが必要

3 単純、限定的なアーキテクチャ
Linux、Android等の高性能なOSへの移植が困難

4 少ない開発経験者数
開発経験のあるエンジニアの採用が困難

5 限定的なサポート体制
日本国内、日本語でのサポートが無いか限定的

6 見た目、操作性、パフォーマンス
現代的・スマートフォンのようなUXが実現できない

柔軟性の課題

拡張性の課題

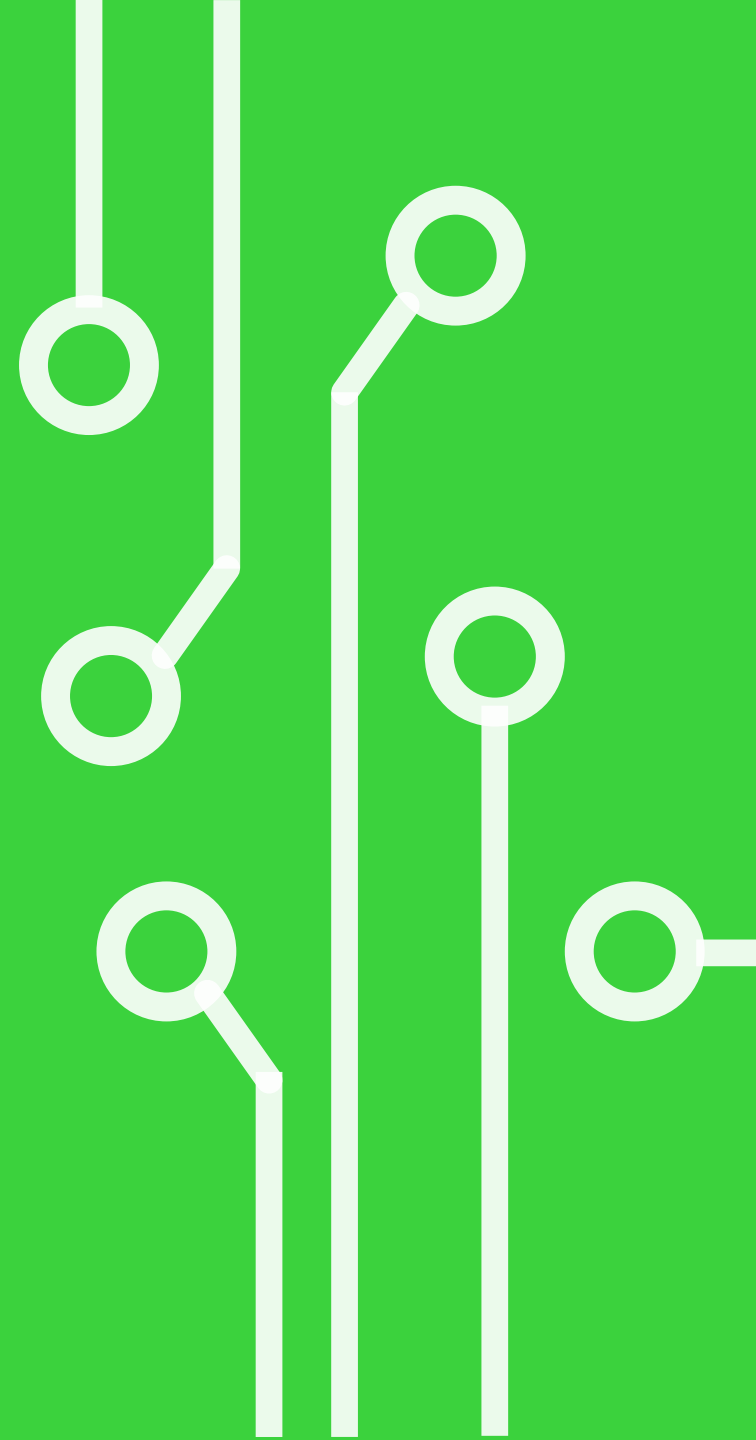
UX上の課題



for MCUs

Extend your use of Qt

Smaller footprint, Equal benefit, Broader use



Qt for MCUs –最小の資源で究極のパフォーマンス

Qt for MCUsは新しい Graphics Runtime (Qt Quick Ultralite) を使用。
QMLをC++に変換することにより少ないメモリ消費で高いパフォーマンスを実現。

Qt for MCUs 構成

GUI
アプリケーション

QML UI
フロントエンド

C/C++ logic
バックエンド

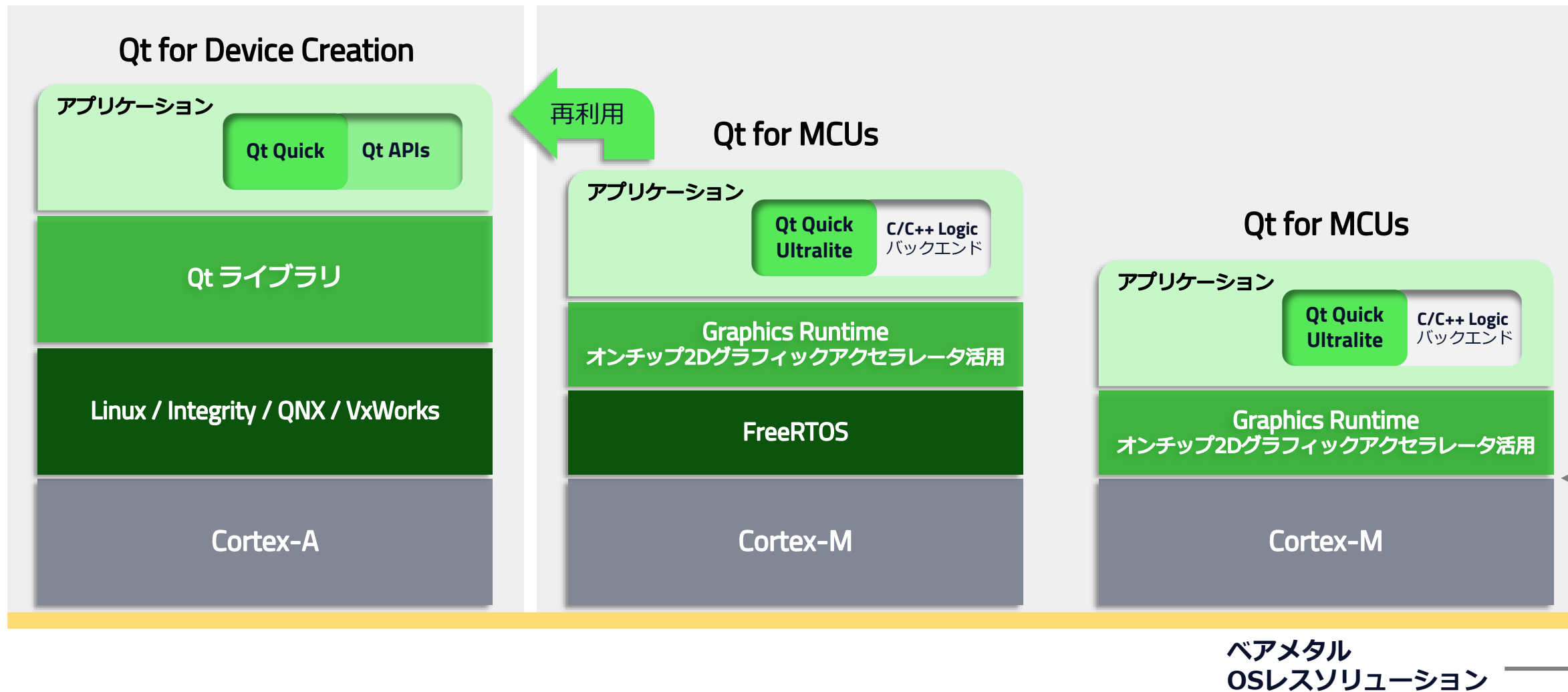
Qt Quick Ultralite
Graphics Runtime

オンチップ2Dグラフィックアクセラレータ活用




















スケーラビリティ実現のための新しい取り組み

Qt for MCUsで開発されたコードはQt for Device Creation向けに再利用可能



Qt for MCUs on STM32 MCUs

 STM32 MCUs 32-bit Arm® Cortex®-M			 STM32 Solutions	
 High Performance	 STM32H7 3224 CoreMark 480 MHz Cortex-M7 240 MHz Cortex-M4		Artificial Neural Networks 	
	 STM32F7 1082 CoreMark 216 MHz Cortex-M7	STM32F4 608 CoreMark 180 MHz Cortex-M4	 Qt interface 	
 Mainstream	STM32G0 142 CoreMark 64 MHz Cortex-M0+	STM32F2 398 CoreMark 120 MHz Cortex-M3	STM32G4 550 CoreMark 170 MHz Cortex-M4	STM32 Motor Control 
	STM32F0 106 CoreMark 48 MHz Cortex-M0	STM32F1 117 CoreMark 72 MHz Cortex-M3	STM32F3 245 CoreMark 72 MHz Cortex-M4	STM32 Connectivity 
 Ultra-low-power	 STM32L4+ 449 CoreMark 120 MHz Cortex-M4		STM32 USB Type-C 	
	STM32L0 75 CoreMark 32 MHz Cortex-M0+	STM32L1 93 CoreMark 32 MHz Cortex-M3	STM32L5 442 CoreMark 110 MHz Cortex-M33	STM32L4 273 CoreMark 80 MHz Cortex-M4
 Wireless	STM32WB 216 CoreMark 32 MHz Cortex-M0+ 64 MHz Cortex-M4		STM32 Community 	
	STM32WL 161 CoreMark 48 MHz Cortex-M4		STM32 Education 	

Qt for MCUs on NXP i.MX RT MCUs



Qt

Qt

Qt

Qt

Key Features										
Product	CPU	Package	Memory	Graphics Acceleration	Display Interfaces	Camera Interfaces	Audio	USB with PHY	Ethernet	CAN
i.MX RT1170	Cortex-M7 @1 GHz + Cortex-M4 @400 MHz	289 BGA	2MB SRAM	2D GPU, PxP	Parallel, MIPI	Parallel, MIPI	4x I2S, SPDIF, DMIC	2	2x Gbps, 1x10/100	3x CANFD
i.MX RT1064	Cortex-M7 @600 MHz	196 BGA	4MB Flash 1MB SRAM	PxP	Parallel	Parallel	3x I2S, SPDIF	2	2x 10/100	2x FlexCAN, 1x CANFD
i.MX RT1060	Cortex-M7 @600 MHz	196 BGA	1MB SRAM	PxP	Parallel	Parallel	3x I2S, SPDIF	2	2x 10/100	2x FlexCAN, 1x CANFD
i.MX RT1050	Cortex-M7 @600 MHz	196 BGA	512 kB SRAM	PxP	Parallel	Parallel	3x I2S, SPDIF	2	1x 10/100	2x FlexCAN
i.MX RT1020	Cortex-M7 @500 MHz	100 LQFP 144 LQFP	256 kB SRAM	-	-	-	3x I2S, SPDIF	1	1x 10/100	2x FlexCAN
i.MX RT1015	Cortex-M7 @500 MHz	100 LQFP	128 kB SRAM	-	-	-	3x I2S, SPDIF	1	-	-
i.MX RT1010	Cortex-M7 @500 MHz	80 LQFP	128 kB SRAM	-	-	-	2x I2S, SPDIF	1	-	-
i.MX RT600	Cortex-M33 @300 MHz + Cadence® Tensilica® HiFi 4 @600 MHz	176 VFBGA	4.5 MB SRAM	-	-	-	DMIC	1	-	-

Qt for MCUs on Renesas MCUs

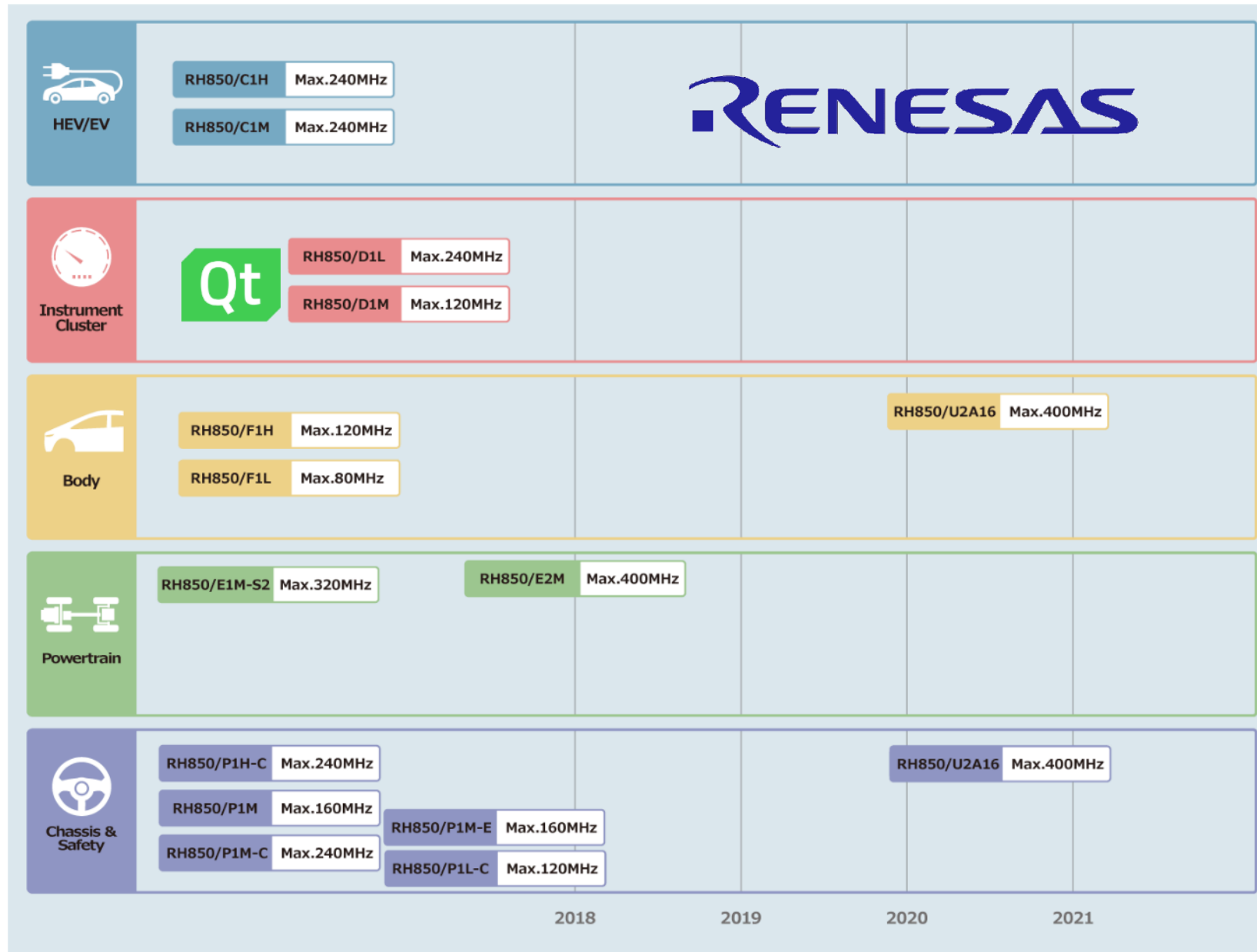
Renesas RA Product Series



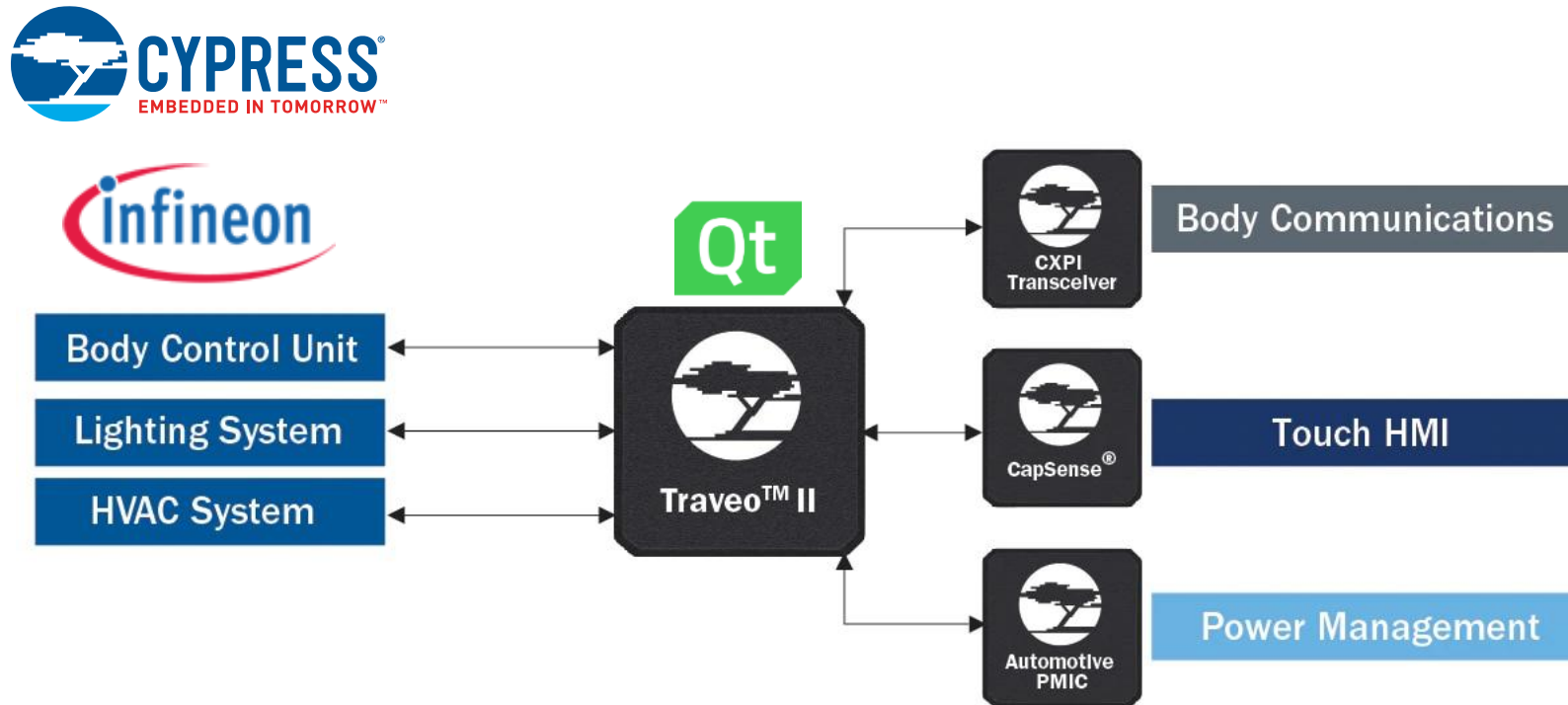
The four Renesas RA Family MCU series are based on 32-bit Arm® Cortex®-M cores. All four Renesas RA Series have been designed on common DNA, making these products feature- and pin-compatible. This allows easy scalability and code reuse from one device to another.

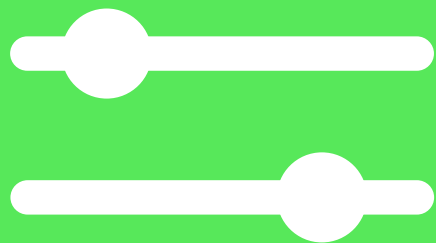
		Performance Range	Feature	Series Memory Ranges	ASSP Extensions
Qt	RA8	More than 200MHz 1.8-3.6V	Highest Performance, HMI, Connectivity, Security, Analog	Highest Memory Integration: 2MB Flash, 1MB SRAM	HMI Analog
Qt	RA6	Up to 200MHz 2.7-3.6V	Advanced Performance, Connectivity, Security	High Memory Integration: up to 2MB Flash, 640kB SRAM	Motor/Inverter Control Wireless HMI
	RA4	Up to 100MHz 1.6V-5.5V	Excellent Power, High Performance Mix Paired with Security	Medium Memory Integration: up to 1MB Flash, 128kB SRAM	Wireless Sensor
	RA2	Up to 60MHz 1.6V-5.5V	Low Power	Medium memory integration: 512kB Flash, 64kB SRAM	Rich Analog Wireless

Qt for MCUs on Renesas MCUs



Qt for MCUs on Infineon (Cypress) Automotive MCUs





UX

Qt Quick Controlsによる
スマートフォンのUX実現



Reuse

ARMアーキテクチャ間で
ソースコードを再利用



Speed Up

QMLとQtツール群による
迅速かつ効率的な開発

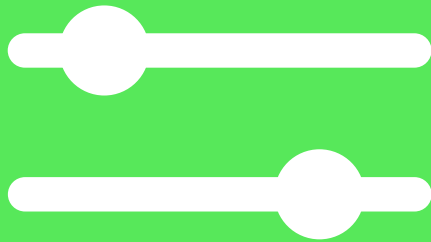


Community

開発コミュニティから得
られる恩恵

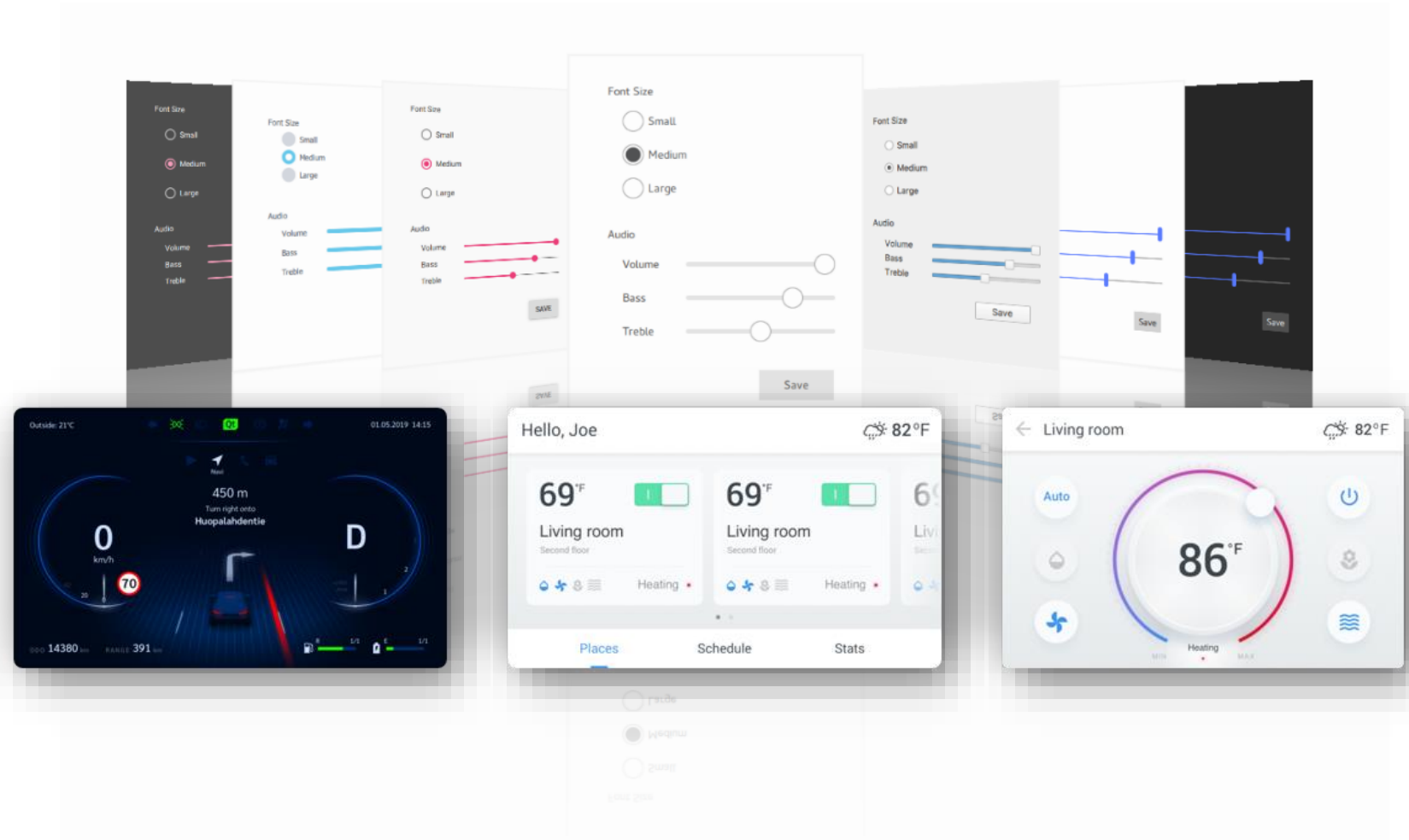
Qt Quick Controls 2.0 によるスマートフォンのようなUXの実現

多彩なUI controlsライブラリを活用した効率的な開発



UX

Qt Quick Controlsによる
スマートフォンのようなUX実現



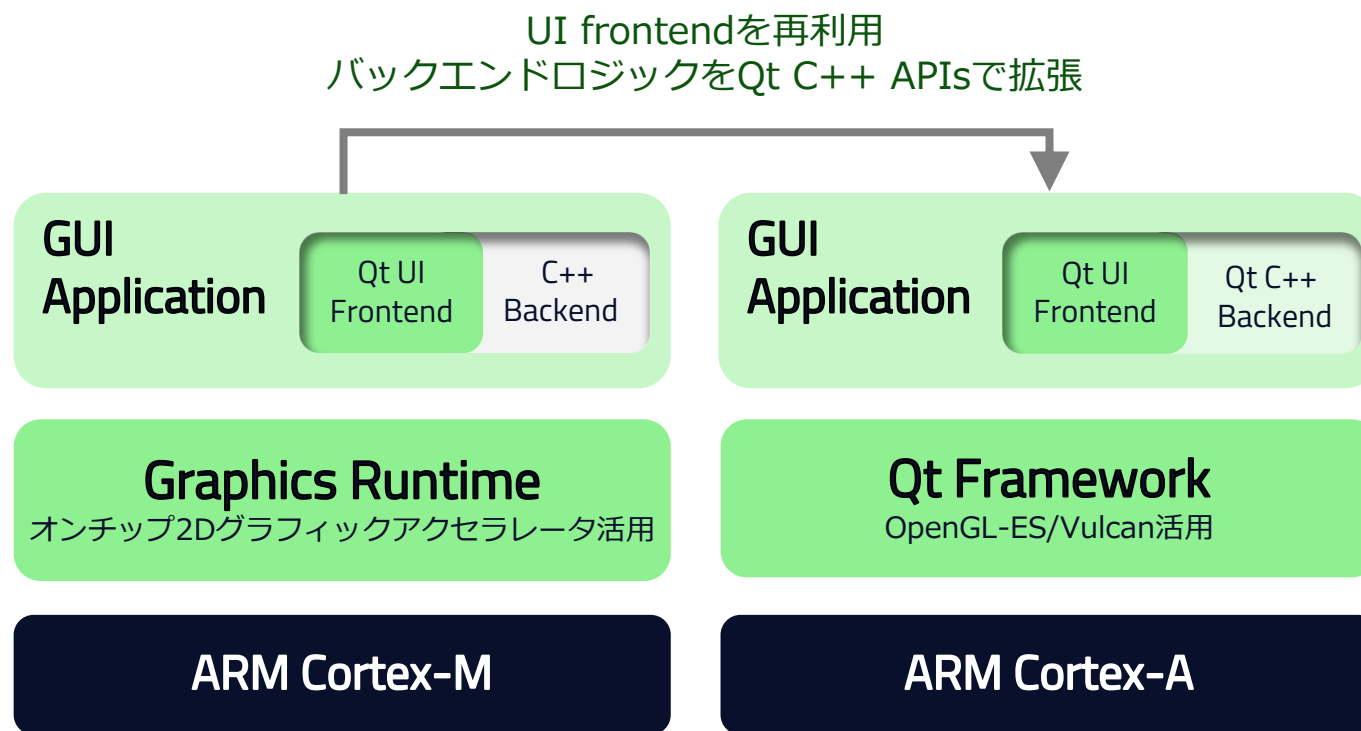


Reuse

ARMアーキテクチャ間
でソースコードを再利用

UIグラフィックスをアプリケーションプロセッサ向けに再利用

一度のコードで、どこへでも展開





Speed Up

QMLとQtツール群による
迅速かつ効率的な開発

QMLとQtツール群の活用による開発プロセスの加速

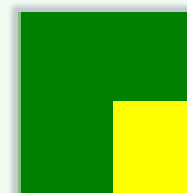
UIフロントエンドには簡単で直感的なQML、ロジックバックエンドにはC++、
機能的なツール群活用による開発プロセス全体を効率化

QML活用

QMLはJSONに似た構文で、UIフロントエンドの効率的な開発をサポート
宣言型言語であるQMLはバイナリーへコンパイル可能

```
Rectangle {  
  x: 50  
  y: 50  
  width: 100  
  height: 100  
  color: "green"  
}
```

```
Rectangle {  
  x: 100  
  y: 100  
  width: 50  
  height: 50  
  color: "yellow"  
}
```



Qtツール群活用

既存のQt Creator、Qt Design Studioによる開発が可能
Qt Design Studioの活用によりデザイン仕様を直接QMLとして出力可能



デザイナー

UI flow & navigation
Wireframes
Visual assets

デザインプロセス



SW開発プロセス

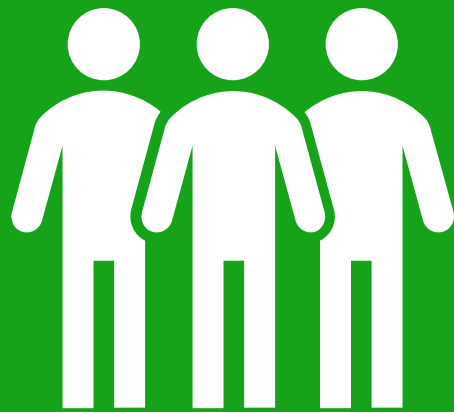


Qtツール群を
活用した協業



SWエンジニア

Custom UI components
Data bindings
Application logic



Community

開発コミュニティから
得られる恩恵

既存の社内Qt経験者による開発が可能

社内Qt経験者や外部Qtユーザーコミュニティによる知識、人材的なサポート



>1M
Qt ユーザー数

多数のQtユーザー（開発者）の存在

Qtユーザーコミュニティの恩恵

Thanks to its roots in the Open-Source community, Qt constantly evolves through contributions from helpful developers around the world.

"What was amazing was that there was already a body of work done by the Qt Community. Had that open source community not been there we would have taken a much longer time to deliver."

マイコン向けソフトウェア開発における課題

現行の課題

1 ビジュアルエディタのみによる制限的な画面開発
プログラミングによる拡張的なUI開発を実現するAPIの欠如

2 固定的なレイアウト
画面サイズ、縦横比の変更にはコードの書き直しが必要

3 単純、限定的なアーキテクチャ
Linux、Android等の高性能なOSへの移植が困難

4 少ない開発経験者数
開発経験のあるエンジニアの採用が困難

5 限定的なサポート体制
日本国内、日本語でのサポートが無いか限定的

6 見た目、操作性、パフォーマンス
現代的・スマートフォンのようなUXが実現できない

Qt for MCUs

1 ビジュアルエディタ、C++、QML API
Qt Design Studio、QML APIによる拡張的なUI開発を実現

2 スケーラブルUI
相対的レイアウト開発により画面サイズ、縦横比の変更に対応

3 クロスプラットフォームアーキテクチャ
QtライブラリによりLinux、Android等の高性能なOSへ移植可能

4 世界100万人以上のユーザーベース
開発経験のあるエンジニアの採用が容易

5 日本国内、日本語によるサポート
経験豊富な日本人スタッフによるローカルサポート

6 最新のUI Controls
豊富なQt Quick ControlsによるスマートフォンのようなUXを実現

Qt for MCUs ロードマップ

2019年12月

Release 1.0

最高クラスのグラフィックスパフォーマンス実現

- ルネサス社、ST社、NXP社 MCUs 対応

多言語対応

- 多言語切り替えに対応

ローエンドユースケース対応

- Bare metal サポート

2020年4月

Release 1.1

マルチタスクアプリケーション対応

- FreeRTOS サポート(テクニカルプレビュー)

高度なHMIアプリケーションへの対応

- 複数のメモリー上へのプログラム、アセットの分割搭載を実現

アプリケーションサイズの縮小

- PNGの圧縮への対応(ランタイム上で復元可能)

MCUsへの対応拡充

- NXP i.MX RT1170 and STM32H7

2020年6月

Release 1.2

マルチタスクアプリケーション対応

- FreeRTOS サポート(正式サポート)

フォントエンジン

- リソース制約のあるデバイス向けにフォントエンジンサポート(テクニカルプレビュー)

HMIツールサポート

- Qt Design Studio

プラットフォームの抽象化対応

- プラットフォームの抽象化によりカスタムHWへのポーティングが比較的簡単に

MCUsへの対応拡充

- NXP i.MX RT1060, Cypress Traveo II and Renesas RA6

Qt for MCUs ロードマップ

今後のリリース予定

更なるMCUs、RTOS、コンパイラーへの対応

- uITRON 他、ご要望によりさらに多くのプラットフォームへの対応

3rd party開発環境への対応

- 3rd party SDK、IDEへの対応

多彩なユースケースへの対応

- Video and Audio, 機能安全



Thank you

The future is written with Qt

www.qt.io