

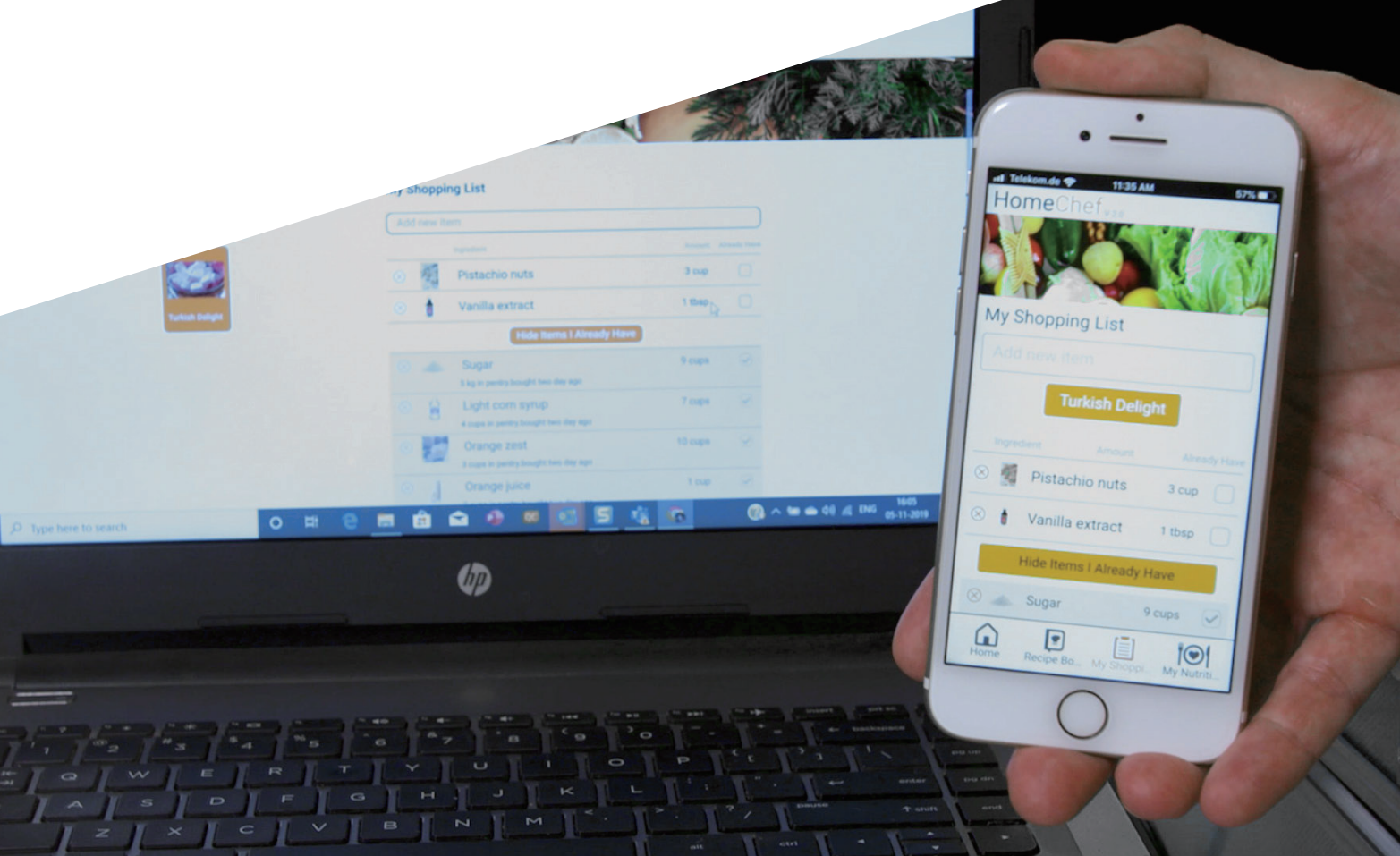
基于统一的 Qt Quick 架构

将 Qt Quick 应用移植到 Qt for MCUs

本白皮书分享了我们利用 Qt for MCUs 将现有的 Qt Quick 应用程序移植到微控制器 (MCU) 的经验。

我们选择将“HomeChef”应用程序移植部署到桌面、移动和嵌入式系统中。我们使用了统一的架构实现不同的部署，这些部署具有不同的应用场景和操作系统。所有的部署，包括 MCU 移植，都基于相同的代码库和工具链，从而打造统一的品牌体验。

我们将深入介绍使用 Qt for MCUs 1.0 将 Qt Quick 应用程序缩小到 480kb 并将其移植到 STM 32F769 的整个过程。我们将解释如何在极短的时间内解决这些挑战，同时仍然能达到极高的性能，这通常在更强大的硬件上才能实现。



目录

HomeChef 生态系统：统一的 Qt Quick 架构	3
多场景部署的开发挑战	3
Qt 技术论证	3
利用 Design Studio 和 WebAssembly 开发 HomeChef 的经验	4
向 Qt UltraLite 扩展	5
我们为什么要在 MCU 上开发人机交互界面（HMI）？	5
HomeChef 自上而下的开发方法	5
差异分析 (Qt for MCUs 还缺少哪些功能)	6
迁移到 Qt Quick UltraLite (QUL) 的挑战	8
总体工作和时间安排	11
结论 / 经验	12

HomeChef 生态系统： 统一的 Qt Quick 架构

多场景部署的开发挑战

当您的用户想在多种设备上使用自己最喜欢的软件时，他们希望得到一致的体验。与此同时，随着您将软件移植到越来越多的平台上，您创建一个有凝聚力的、统一的产品体系结构的挑战也会越来越大。此外，增量和迭代开发过程的转变将为产品开发增加了额外的工作。

在这种分散的产品交付环境中，集成已经成为一个关键问题。为了解决这个问题，我们相信可以通过统一架构 (UAM) 之类的方法利用离散用户界面和跨平台环境中的可重用组件来支持跨平台的产品交付。

随着技术要求的提高，满足新型、创新的设备设计的解决方案变得更加复杂和昂贵。虽然易于使用的 MCU 在这些全新的复杂实现中表现不佳，但高端工业微处理器 (MPU) 具有更高的软、硬件复杂性，这可能导致后续高昂的成本。

Qt 技术论证

为了介绍如何利用 Qt 应对这些挑战，我们开发了一个消费类电子产品示例“HomeChef”，作为一个将多平台部署成生态系统的实际用例。我们的想法是，优化烹饪（更快、更便宜、少浪费、更健康）将是未来可持续发展的一个关键方面。HomeChef 让人们随时了解厨房里有什么，以及需要在超市里买什么。我们相信类似的场景将适用于其他行业和用例。HomeChef 有三种不同的应用和设备：

桌面应用帮助准备本周菜肴：从食谱书籍、收藏夹或自动推荐中选择食谱，并将配料添加到购物清单中。HomeChef 还可以收集营养信息，并显示健康生活的统计数据。我们没有创建本地桌面应用，而是决定采用 WebAssembly 技术进行部署，这样用户就可以在不安装任何软件的情况下运行该应用，这使得 HomeChef 成为展示用 Qt 开发的软件即服务 (SaaS) 的窗口。

第二个部署是移动应用，它具有与桌面版相同的功能，但针对更小外形尺寸进行了优化，成为 iOS 和安卓系统的原生应用。

最后，我们开发了嵌入式烹饪设备，它可以在用户做饭时提供分步骤烹饪指导。我们用的是 Garz&Fricke 的 Santaro，一个基于 i.MX6Dual 的人机界面 (HMI) 面板，它是以 Yocto 为构建系统运行嵌入式 Linux 的。其中的技术亮点是添加了 WebGL 技术用于远程观看摄像机视频流以及集成 WebEngine 用于观看 Youtube 视频。

我们相信可以通过统一架构 (UAM) 之类的方法利用离散用户界面 (UI) 和跨平台环境中的可重用组件来支持跨平台的产品交付。

所有部署都通过云端的 MQTT¹ 代理进行通信 (或出于演示目的在本地运行)。



图 1: 统一的 Qt Quick 架构

利用 Design Studio 和 WebAssembly 开发 HomeChef 的经验

经过初步培训, 我们第一次使用 Qt Design Studio² 就发现在开发过程中运用它很方便。一开始, 我们使用 Qt Design Studio 进行原型设计, 但很快它就成为了必不可少的开发工具。

- 1 [MQTT](#) 是一个利用发布 / 订阅模式的机器对机器 (M2M) 协议。其目的是提供一个通信开销最小的信道。
- 2 Qt Design Studio 是一个用户界面 (UI) 设计和开发环境, 用于创建动画 UI, 并在桌面、安卓和嵌入式 Linux 设备上进行预览。

向 Qt UltraLite 扩展

我们为什么要在 MCU 上开发人机交互界面 (HMI) ?

作为一家嵌入式 HMI 开发公司, Verolt 一直对在恩智浦等嵌入式平台上开发应用感兴趣。但是, 这些应用程序几乎都需要基于 Linux 的处理器和 GPU 以及其他硬件支持。

我们一直考虑将基于微控制器的 HMI 产品添加到我们的产品组合中。然而, 到目前为止, 供应商非常分散, 提供的产品也不完整。当 Qt 公司发布 Qt for MCUs 时, 一个为微控制器优化的图形框架和工具包, 对我们来说投资于这项技术是一个自然而然的选择。

我们将继续介绍我们的方法, 对用 Qt for MCUs 开发人机交互界面的分析, 我们发现的差距, 以及为缩小这些差距所采取的措施。

HomeChef 自上而下的开发方法

当 HomeChef 应用程序在 Linux 上的设计和开发接近完成, 我们决定在微控制器上对 HomeChef 采用自上而下的开发方法。

分而治之: 统一体系结构的优点在于您可轻松地将项目分解为更小、更易管理的部分。我们分析了 HomeChef 的功能及其与微控制器的兼容性, 就可行性问题数次集思广益, 并定期召开带验证的实时编程会议, 帮助团队克服未知因素, 以最小的设计更改实现功能完成的目标。

借助这种自上而下的方法, 我们可以做到

- 关注与客户最相关的功能
- 立即测试完成的新功能。集成测试和用户测试在整个项目中频繁发生, 这使得结果更加可预测
- 定期向客户发送包含每个新实施功能的持续构建

统一体系结构的优点在于
您可轻松地将项目分解为
更小、更易管理的部分。

差异分析 (Qt for MCUs 还缺少哪些功能)

微处理器通常运行允许多处理和多线程的操作系统。微控制器如果不运行这样的操作系统将影响应用程序的设计。下表比较了 Qt for Embedded Linux 和 Qt for MCUs 之间的功能差异。

嵌入式 Linux 上的 Qt 模块	Qt for MCUs 上的可用性
<u>Qt Network</u>	不可用
<u>Qt Quick Dialogs</u>	不可用
<u>Qt Quick Controls</u>	有限可用
<u>Qt Quick Layouts</u>	不可用
其它模块 (比如: <u>Qt MQTT</u> 、 <u>Qt Serialport</u> 等等 .)	不可用

开发者需要注意，Qt for MCUs 不支持 Qt C++ 模块。但是，它允许您使用预先存在的代码。



图 2: 在微控制器上部署 HomeChef

我们决定移植前端，因为后端需要花费更多精力集成 MQTT 和 WIFI 模块。

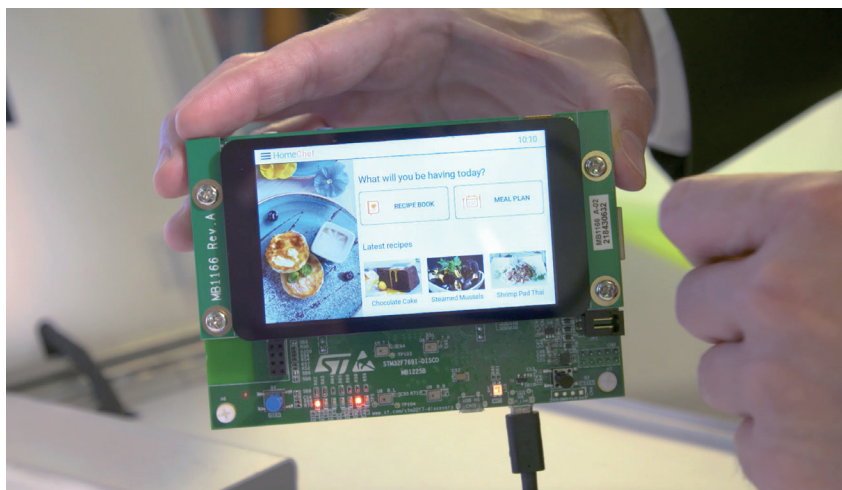
Qt for MCUs 提供了一个 Qt Quick 子集，称为 Qt Quick UltraLite³ (QUL)，它解决了微控制器中的资源限制。与 Qt Quick 相比，QUL 占用的空间更少，其代价是功能有限。为了获得预期的结果，我们必须简化某些 QML⁴ 设计元素。下面是此开发阶段的经验。

Design Studio

我们利用 Design Studio 开发了前端。由于开发时还不支持 QUL，某些依赖于 Design Studio 组件和 Qt Quick 的组件需要返工。

Design Studio 组件	基于 Qt for MCU 进行重构	重构原因
Circular progress bar	QML 需要使用基于时间轴的动画来更新，而不是基于时间轴的动画	Design Studio 的 <u>时间轴</u> 组件在 Qt for MCUs 上不可用
Charts	完全重新设计了 QML 中的组件	Qt for MCUs 上没有 <u>QChart</u> 模块
烹饪步骤	轻微修改	不支持 QML 类型

在微控制器上运行的 HomeChef



- 3 优化图形运行时，以在资源受限的设备上实现高性能、低内存消耗。
- 4 Qt Modeling Language 是一种用于设计以 UI 为中心的应用程序的声明性标记语言。

迁移到 Qt Quick UltraLite (QUL) 的挑战

在迁移过程中，我们发现了 QUL 和 Qt Quick 之间的差异。下面是我们需要为 HomeChef 返工的元素列表。

Qt Quick	QUL 1.0	限制	解决方案
QTimer	有限可用	函数 Triggeronstart 不可用	开始时调用显式
Rectangle border	不可用		重叠矩形来显示边框
List View	有限可用	水平滚动不可用。	用可闪烁属性实现水平滚动的。
Text Input	不可用		自定义类型
Text Edit	不可用	-	自定义类型
Property Animation	有限可用	属性动画到锚不可用	利用计时器和转换属性来旋转对象。在 Photoshop 中预绘动画
Java script functions	不可用		
Popup	不可用		有 Z-order 的自定义类型
Font	有限可用	字体：加粗字体会显著增加内存占用。字体：像素大小应该是一个固定的值	应避免动态调整字体大小。
Loader	不可用		在页面加载时使用可见性属性
Scroll View	不可用		垂直滚动的可滑动属性
Drawer	不可用		自定义类型的动画和可视性
Stack view	不可用		拥有可视性的自定义类型
Model	有限可用	不能在其它文件中申明	模型和委托应该在同一个文件中声明
Combobox	不可用		自定义类型
Application window	不可用		无可替代选项，但未来的版本中有计划
ItemDelegate	不可用		无可替代选项，但未来的版本中有计划
Page	不可用		无可替代选项，但未来的版本中有计划

QUL 性能和微控制器的局限性

个别动画表现出性能下降的迹象。为克服这个性能问题，我们进行了多次试验，通过减少计算并为多个微动画使用单个计时器，最终确定了解决方案。

有一个这样的用例，一个由 Design Studio 元素开发的描述揉捏过程的圆形进度条。在本用例中，我们使用了 Design Studio 元素 "Arc"，它由另一个 Qt Design Studio 元素 "Timeline" 生成动画。因为这些元素是 Qt Design Studio 的一部分，所以必须重新设计循环进度条。

- 进度条的动画需要更多的内存和能耗，所以我们必须采用基于计时器的旋转方法。我们发现这种方法极大地减少了内存占用，并提供了平滑的动画。
- 最后，我们的应用程序总大小为 480KB，并且能够提供几乎与嵌入式 UI 相同的外观。

图 3 通过重用相同的定时器优化屏幕中的动画元素（温度、风扇速度和进度条）



总体工作和时间安排

将任何软件迁移到微控制器的关键是以更低的开发成本和更快的上市时间交付具有可行软件架构的嵌入式系统。

就 HomeChef 而言，统一架构的选择和 Qt 在多个平台上的优势帮助开发者以最少的学习曲线在微控制器上实现了快速部署。由于缺乏嵌入式软件资源和开发时间，需要重用软件组件。基于我们在 HomeChef 中最复杂的四个屏幕上的工作，以下是将 Qt Quick 移植到 QUL 所花费的时间和精力。我们的评估是基于一位拥有基本 QML 知识的中等水平的开发者。

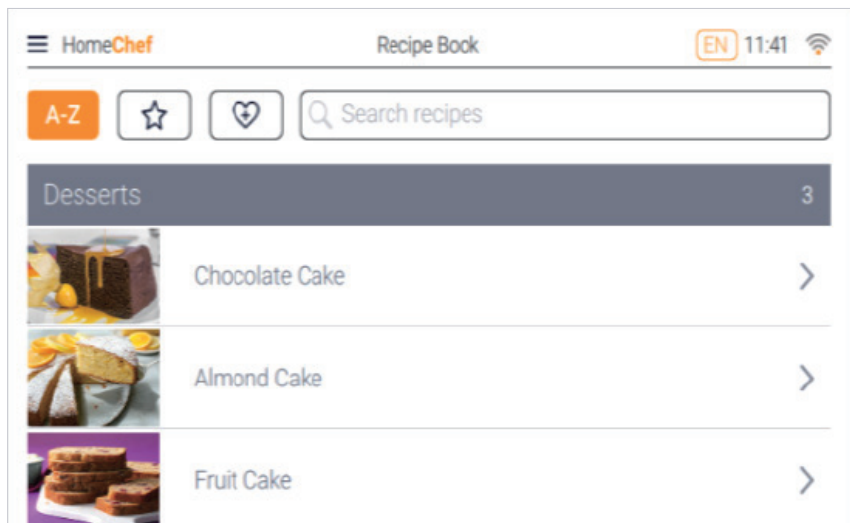
Screen



复杂度：中

- 界面必须进行中等级别的修改，包括水平滚动列表和其他 GUI 元素。
- 所需时间：一天

Screen



复杂性：高

- 嵌套可折叠列表：该组件必须进行 GUI 更改才能在微控制器上运行。
- 为了适应小屏幕尺寸，更改了用户界面，放大了字体和鼠标区域。
- 所需时间：三天

Screen



复杂性：中

- 动画循环进度条
- 将 Qt Design Studio 元素替换为基于计时器的动画
- 图像叠加已用于显示数据和剪辑不需要的图像数据
- 所需时间：一天

Screen



复杂性：中

- 循环进度条
- 旋转动画
- 相同的计时器用于多个微动画，如风扇或温度。
- 所需时间：两天

Qt 公司凭借 Qt for MCUs 填补了低成本、高产量制造业领域的重要空白。

结论 / 经验

通过这个演示，我们已证明可以在所有部署场景中使用相同的设计和代码库。因此，需要的工程技能更少，设计和开发之间的协作更顺畅，从原型到产品开发的过渡更平稳，产品迭代可同步，组件可跨 UI 风格、版本和平台重复使用。

Qt 公司凭借 Qt for MCUs 填补了低成本、高产量制造业领域的重要空白。正如我们所展示的，1.0 版本已经提供了一个全面的产品组合。虽然如果有改进的 QML 类型和 SCXML⁵、完整的 Design Studio 集成或图像处理（例如 2.5D 动画）之类的东西会更好，但是对于我们来说，它们并没有成为前进的阻碍。

总之，HomeChef 生态系统的概念证明表明我们可以通过 Qt for MCUs 减少客户的开发成本和材料成本。

如果没有遗留代码的限制，使用 Qt 就可以实现包括微处理器在内的统一架构，如果处理得当，还可以保证开发团队未来的创新。作为该领域的服务提供商，我们没有其他更好的选择。

作者



Lars König

软件服务总监

德国乌尔姆

+49 151 612 575 36

Lars.Koenig@verolt.com

www.verolt.com



Ravi Dattatraya

工程部总监

印度班加罗尔

+ 91 90359 87074

Ravi.Dattatraya@verolt.com

www.verolt.com



5 Qt SCXML 模块提供了从 SCXML 文件创建状态机的功能