



EBOOK

The Nuts and Bolts of
**Qt Industrial
Applications**



Introduction

If you read our last eBook, *The Perfect Framework for Industrial Applications*, you may already know that Qt is well suited for building industrial applications. But what's really in an industrial application built with Qt? We'll address that question in this eBook, where we take a look at the common requirements of industrial applications, and what's needed from the Qt toolbox to address those challenges.

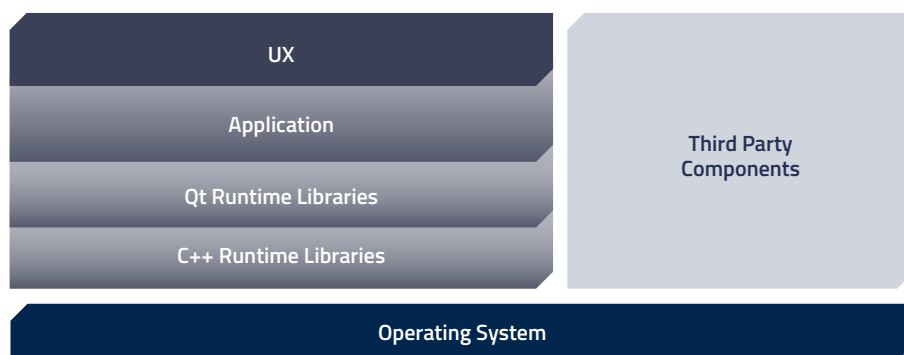
What do industrial applications need?

Although industrial applications can leverage many things from mainline application development, they often have more stringent or more specific requirements than your average web portal, mobile, or desktop app.

- ✔ **Reliability:** Industrial applications must be reliable; mistakes can be terrifically costly in terms of destroyed material, ecological consequences, or human endangerment. Qt is a long-lived and very stable platform with over two decades of testing and development behind it. It is built on top of C++, which does not hide underlying mechanics from engineers and can be statically analyzed for faults, making it the primary language of choice for industrial applications.
- ✔ **Functional safety:** Functional safety takes reliability to the next level and applications that have the potential of injuring people usually require a functional safety certification. Qt works very well within a safety-certified environment, allowing a mix of different safety levels with the Qt Safe Renderer.
- ✔ **Protocols and peripherals:** Industrial platforms are typically called upon to communicate with a wide variety of devices – like PLCs, CNCs, sensors, or printers. The Qt framework supports a large number of devices and protocols with tested drivers and plug-ins. And since C++ allows low-level access to hardware resources, Qt developers can readily integrate support for custom, unique, or obsolete hardware too.
- ✔ **Performance:** Industrial applications are often both resource constrained and performance critical. There is no better fit for this than Qt, which is able to deliver buttery-smooth and hardware-accelerated user interfaces even on low cost hardware. Due to C++'s excellent compiler performance, developers can also create code that directly communicates with devices or protocols demanding precise timing.
- ✔ **Security:** While all software should be secure from intrusion, industrial applications absolutely demand it. The C++ language that underpins Qt has the most tools for static analysis that can spot security flaws, allowing you to create highly secure applications.
- ✔ **Remote UI** – The need to monitor or control applications remotely – via an operators tablet, smartphone or desktop – is a very common requirement for an industrial application. Qt provides the tools necessary to easily accomplish this as an extension to an existing user interface design.

What's the structure of a Qt app?

With all of those requirements in mind, let's examine the high-level structure of a typical Qt industrial application and see how things look.



Software architecture for Qt-based industrial application.

UX – bridge between your app and the user. Written by your team, it uses the full power of the Qt framework, and contains the display logic for your interface.

Application – the main meat of your business logic, including services and glue code. It will also contain your company's secret sauce – libraries that control your custom hardware or encode your particular algorithms. For most industrial applications this block is composed of several distinct applications, not a monolithic executable.

Qt runtime libraries – core internals, support functions and libraries of Qt. This contains both the UI components (Qt Widgets, Qt Quick/QML, Qt 3D, printing, etc) as well as non-UI functionality (containers, strings, IPC, threading, parsers, etc).

C++ runtime libraries – runtime components and libraries provided by your C++ compiler, including things like STL or Boost.

Third party components – software libraries, databases, or protocol stacks – things your app relies on that your company didn't write, that aren't a part of the core Qt system.

Operating system – interface to your hardware. Depending on requirements, this may be a desktop OS, a Linux variant, or an RTOS like WindRiver VxWorks, QNX Neutrino, or GreenHills INTEGRITY.



Industrial-relevant Qt components

When it comes to code that is generically useful in creating an industrial application, Qt has a huge number of core features that make development easier and find their way into most applications. These include containers, Unicode strings, timers, state machines, databases, file management, and much more. Qt also provides several ways (Qt Widgets, Qt Quick, and Qt 3D) to construct a user interface. However, there are many things that Qt provides that are specifically created for the industrial developer. Let's look at some Qt features of either type that are most notable when developing an industrial application.

Qt for Industry Automation solutions

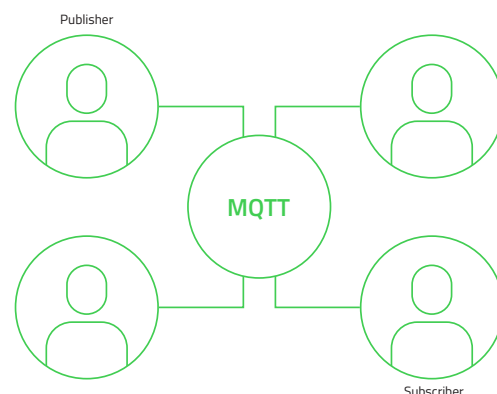
A bundle of technologies, Qt addresses many common industrial requirements. Developers can pick and choose the components they need from the vast Qt offering.

- **Protocols** – to support connecting to automation equipment and networking devices
- **Virtual keyboards** – to enable fully touch interfaces and enclosed, waterproof interfaces
- **Charts and tables** – to create great looking dashboards, alarm handling, and production job lists
- **Internationalization** – to allow multi-language and multi-region support
- **Remote UI** – to allow web-browser (and no-installation) remote access via tables and mobiles

In addition, Qt for Automation – a value added offering to complement the baseline offering – includes several technologies for a common problem – securely and reliably managing objects and devices across a widely distributed network:

QtMQTT

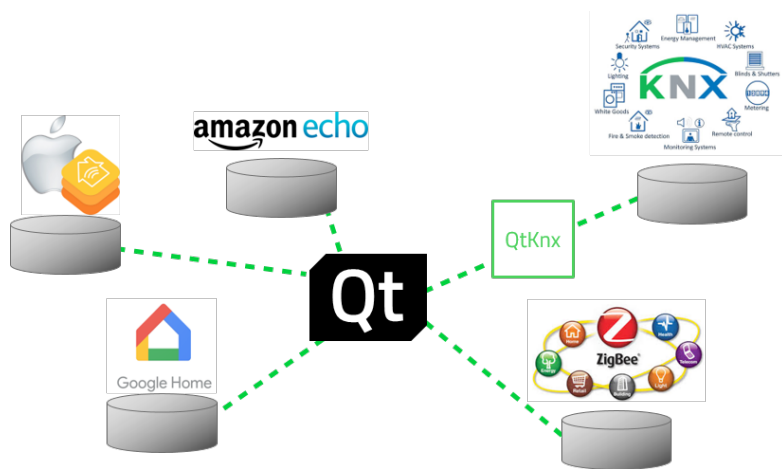
This Qt-specific library offering MQTT client services is based on the standardized publish-subscribe protocol for reliably sharing data between devices or components. Because MQTT is designed to guarantee state correctness, meet high security standards, and exchange minimal data, it's being used in many different distributed systems and IoT solutions.





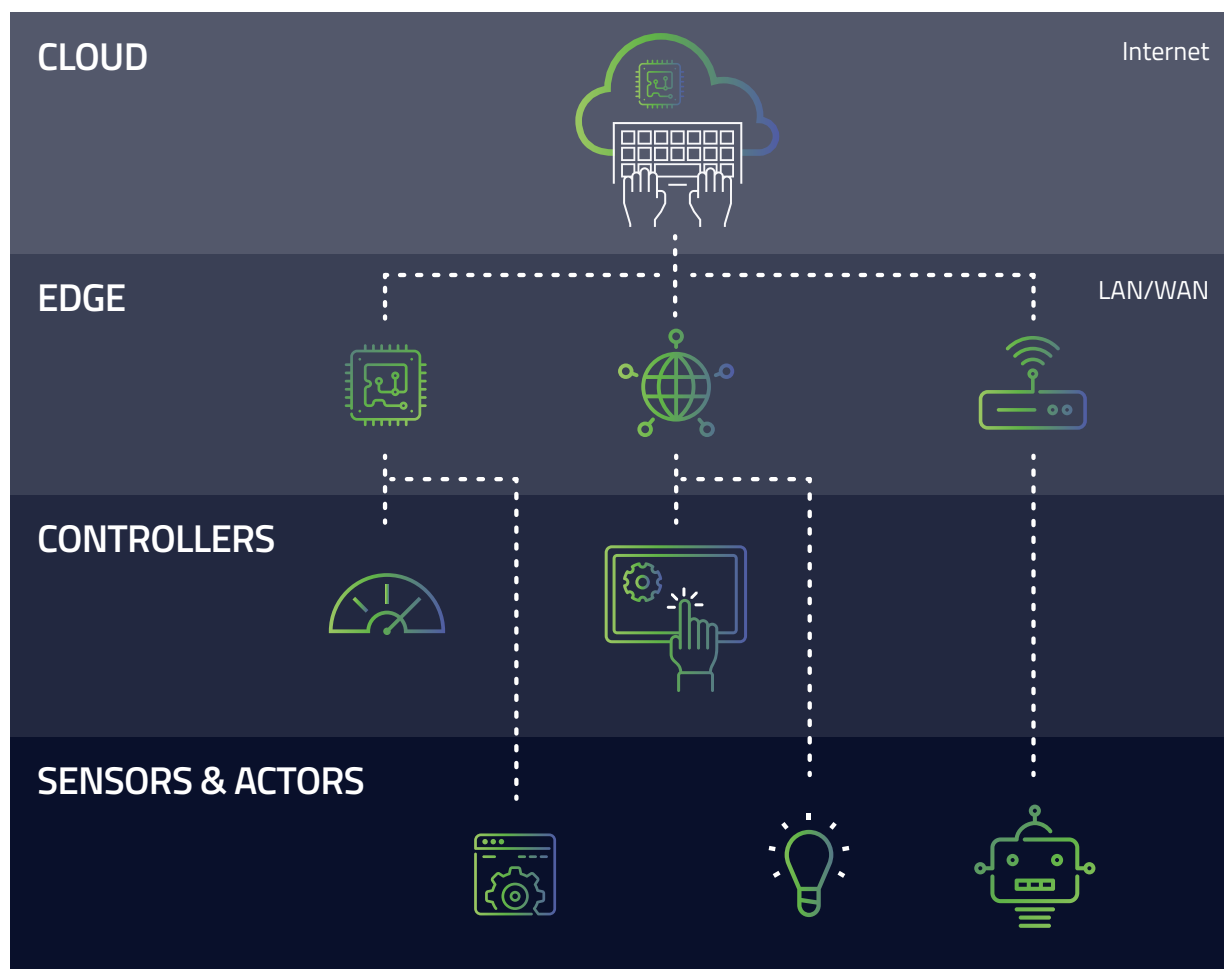
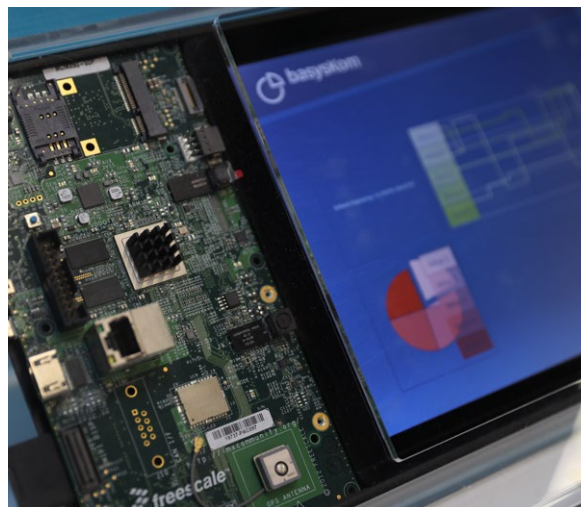
QtKNX

This multi-vendor protocol for smart homes and smart buildings is based on the KNX standard. QtKNX lets your application simply talk to KNX-enabled security systems, HVAC equipment, monitors and alarms, energy control, smart meters – even household appliances.



QtOpcUA

This protocol for data modelling and secure exchange of data in industrial applications is based on OPC UA. Because OPC UA uses data that is object-based while also offering built-in transport security based on certificates and established cryptographic algorithms, it provides a standardized and secure way for a system to share Qt and C++ objects, making it ideal for many industrial applications. (This component is a technology preview in Qt5.11.)



Qt for Device Creation

When looking to build an industrial application with Qt, a good starting point is Qt for Device Creation as it provides all the essential tools preconfigured for building embedded systems. Qt for Device Creation also supplies the Boot to Qt technology for extremely fast boot times – especially important for devices that can only have minimal down time. Qt for Device Creation also adds a number of components that are valuable for creating industrial applications

Qt SerialBus

The Qt Serial Bus API makes it a snap to talk to devices over industrial serial buses and protocols. Use Qt APIs to control and monitor your industrial peripherals through Qt ModBus and Qt CAN Bus, with a wide variety of supported adaptor types.

Qt WebGL

This API for Qt applications allows the application to quickly and efficiently stream a user interface over a network connection to a remote browser, making it trivial to add remote-access facilities to any application.

Qt WebAssembly

If you intend on providing a remote browser companion app to your industrial application, you may want to create it in Qt and use WebAssembly to deploy it. WebAssembly allows you to create an efficient browser application from source code other than HTML5/JavaScript and Qt for WebAssembly provides the means to do this for Qt applications. (This component is available as a technology preview in Qt5.11.)



Qt Creator

Qt Creator is an IDE (integrated development environment) used by programmers to write code. While developers can use whatever editors or tools they wish to create their software, Qt Creator understands Qt constructs, makes it easy to develop Qt interfaces, and is already configured to build Qt applications. Using it saves time in setup and makes on-going development easier.

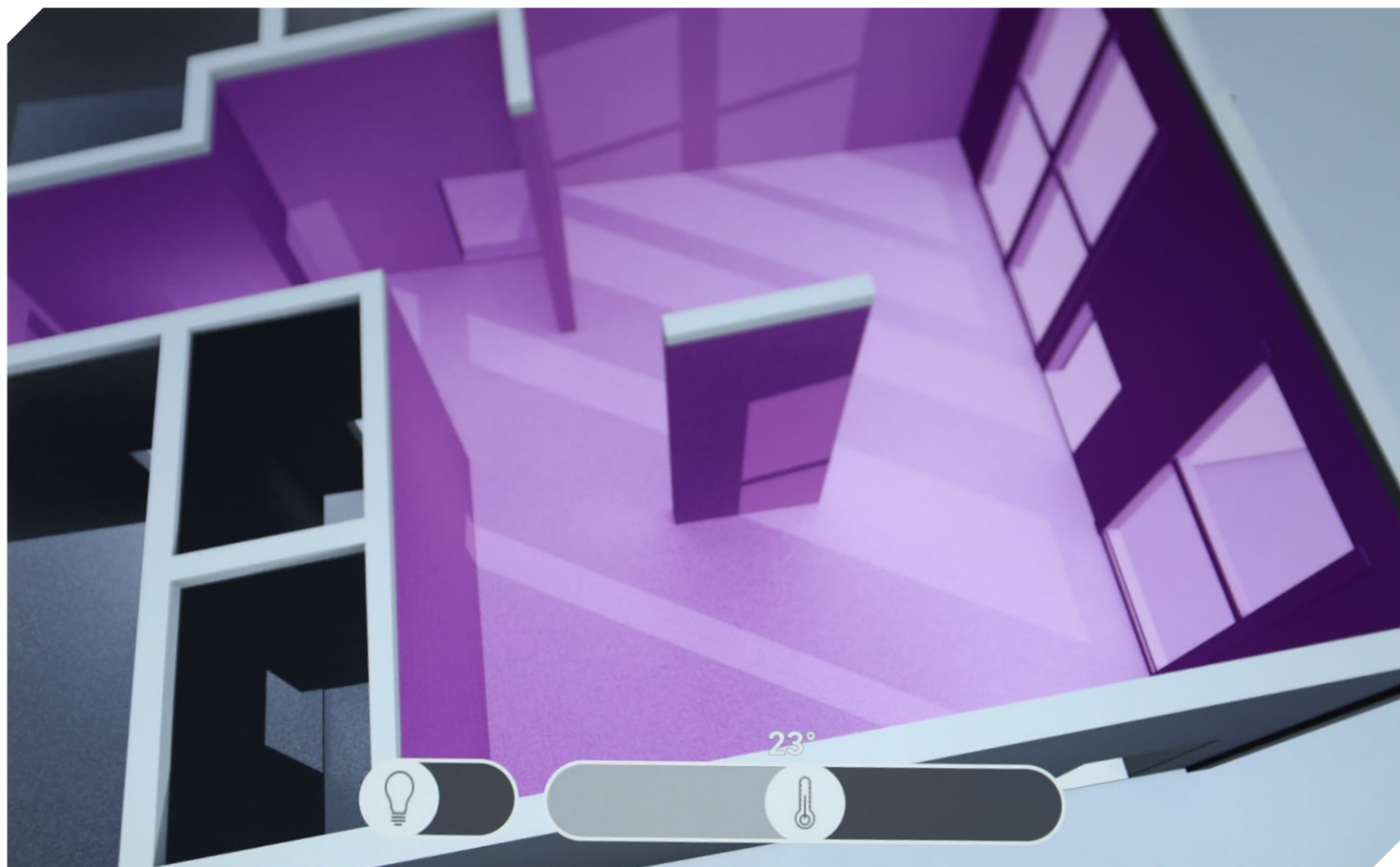
Qt Quick Designer

An essential part of Qt is building user interfaces, which is what Qt Quick Designer is all about. This modern UI technology allows designers to rapidly and easily create a UX in QML, making it very simple to put together an attrac-

tive interface. And because QML is a declarative language, designers need only describe what they want rather than how to achieve it, making it possible to create complex user interactions without understanding how to program.

Qt 3D Studio

The Qt 3D Studio provides tooling that can easily create 3D user interfaces with designer-friendly concepts, methods, and workflow. It also includes a Qt integration and runtime component that allows immediate execution of those interfaces without requiring developer intervention.





Where to go for more

Interested in getting started with Qt?
Maybe you want to learn more specifics
of subjects discussed in this eBook?
Try the following resources.

The Qt Company – detailed info and downloads: <http://qt.io/>

Qt Industrial page – webinars, demos, and datasheets: <https://www.qt.io/qt-in-automation/>

Qt Online docs – reference for components: <http://doc.qt.io/>

Qt community portal – discussion and Q&A: <http://www.qtcentre.org/>

Qt mailing lists – Qt related news: <http://lists.qt-project.org/>



The Qt Company develops and delivers the Qt development framework under commercial and open source licenses. We enable the reuse of software code across all operating systems, platforms and screen types, from desktops and embedded systems to wearables and mobile devices. Qt is used by approximately one million developers worldwide and is the platform of choice for in-vehicle digital cockpits, automation systems, medical devices, Digital TV/STB and other business critical applications in 70+ industries. With more than 250 employees worldwide, the company is headquartered in Espoo, Finland and is listed on Nasdaq Helsinki Stock Exchange. To learn more visit <http://qt.io>