



EBOOK

산업용 어플리케이션을 위한 완벽한 프레임워크



소개

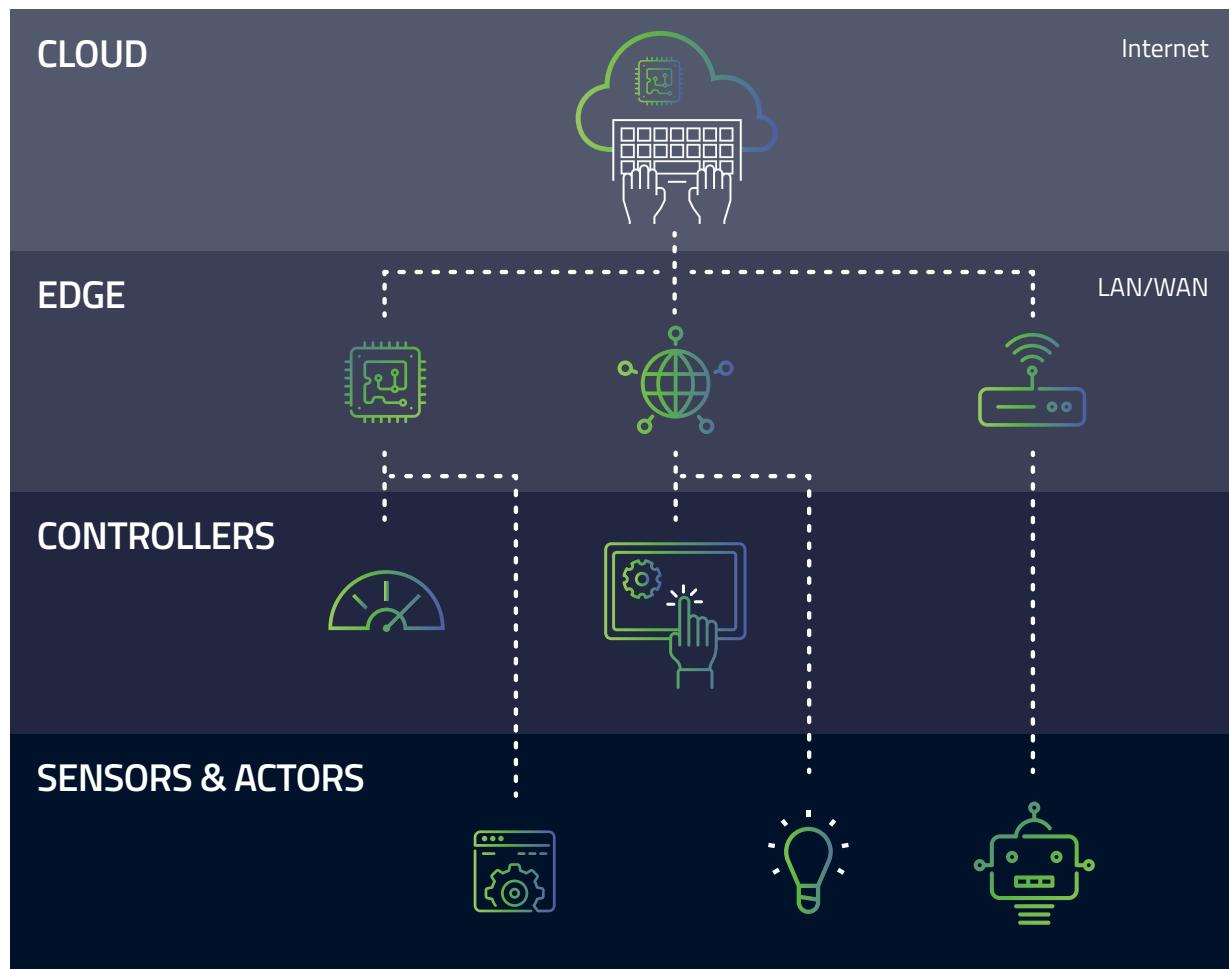
Qt는 산업용 어플리케이션 제작을 위한 강력한 프레임워크이며 이 분야에서 가장 신뢰할 수 있는 기업입니다. 최근 20년간, 제조업체들은 생산 가동 시간과 생산 신뢰도를 향상시켜줄 수 있는 복합적인 시스템을 만들기 위해서 오랜 기간 Qt를 사용해왔습니다. 그러나, 시장은 보다 간소화된 디자인 작업 흐름, 다양한 플랫폼과 디바이스에서 구동가능한 소프트웨어 배포, 원격 브라우저 접근성, 새로운 프로토콜을 통한 통신 등 많은 것을 요구하고 있습니다. Qt는 이러한 요구에 대응해 뛰어난 성능과 기능을 발휘하고 있으며, 프레임워크가 새롭게 릴리즈될 때마다 새로운 기술을 추가하는 방법으로, 시장이 원하는 바를 해결하고 있습니다.

이 eBook에서 우리는 시장의 최신 트렌드에 대해 살펴볼 것이며, Qt가 어떻게 최신 트렌드를 추구하는 회사를 도울 수 있는지에 대해서 설명할 것입니다. 또한, 회사의 목표 달성을 위해 Qt를 성공적으로 잘 활용한 회사의 실제 사례에 대해서도 몇 가지 살펴볼 예정입니다. 여러분이 이 eBook을 모두 읽었을 때 Qt가 시장에 주어진 도전 과제들을 어떻게 해결 했는지에 대해 관심이 생겼다면, "The Nuts and Bolts of Qt Industrial Applications (Qt 산업용 어플리케이션의 너트와 볼트)"라는 제목의 eBook 도 읽어보시기를 권해드립니다. 웹페이지 링크 <https://resources.qt.io/industry-solution-automation>에서 찾아보실 수 있습니다.

산업용 어플리케이션이란 무엇인가?

Qt 어플리케이션은 여러 분야의 산업 영역에서 공통적인 요소를 기반으로 사용되고 있기 때문에, 우리는 산업용 어플리케이션을 상대적으로 넓은 의미로 정의하고 있습니다. 본 문서에서 말하는 '산업용 어플리케이션'은 아래와 같은 경우입니다.

- 설비의 설치, 셋업, 최적화 또는 운영에 사용되는 경우.
- 임베디드 시스템, 데스크탑 어플리케이션, 웨어러블, 모바일/타블렛, 게이트웨이, 헤드리스 장비, 또는 이들의 조합으로 이루어진 장비에 배포되는 경우.
- 종단 장치로 사용되는 경우 또는 사물인터넷(IoT)에서 종단 장치를 관리할 때 사용되는 경우
- 사용상의 용이성과 기능적인 안정성을 바탕으로 높은 신뢰성과 원상 복구 기능이 요구되는 경우. IEC61508, IEC62304, or ISO26262와 같은 안전 표준을 따라야 하는 경우도 포함.



이 정의는 SCADA, 프로세스 컨트롤, PLC나 CNC 머신 같은 기존의 하드웨어를 사용한 원격 운영 등의 과거 방식도 포함하고 있습니다. 그러나, 음향 시스템, 사용자가 직접 조작하는 전기차 충전소, 산업용 게이트웨이, 재고 창고용 웨어러블 손목 밴드 등을 설정할 수 있는 태블릿 기기 중심의 companion app도 포함하고 있습니다.

산업용 인프라에서 오는 시장 대부분의 어플리케이션은 비슷한 소프트웨어 아키텍처를 가지고 있습니다. 이러한 사실은 Qt가 전기, 해양, 의약, 건물 자동화, 창고, 농업, 스마트 시티, 설비 관리, 전력 생산 및 배분, HVAC, 로보틱스, 제조 시스템과 같은 다양한 영역에 적용될 수 있도록 해주고 있습니다.

Qt가 이상적인 이유

Qt는 크로스-플랫폼 개발 프레임워크로서, 네이티브 어플리케이션이 가진 원래의 성능과 속도로 해당 어플리케이션의 상태를 유지함과 동시에, 밑바탕이 되는 코드베이스에서 큰 변경을 하지 않더라도 소프트웨어와 하드웨어 플랫폼 및 다양한 운영 체제에서 동작할 수 있습니다. 하지만 이것은 시작에 불과합니다. Qt가 산업용 어플리케이션 분야에 있어 이상적인 프레임워크라 불리는 이유에는 여러가지가 있습니다.

- 사용자 인터페이스 프레임워크를 사용하는 유저 대부분이 사용하는 가장 큰 개발자 커뮤니티 중 하나이며, 이는 결국, 관련 예제, 소프트웨어 라이브러리, 사용법에 대해 교육받은 엔지니어, 이를 지원하는 회사가 아주 많다는 것을 의미합니다.
- 최신 CPU, 보드, 그래픽, 주변장치, 프로토콜에 적극적으로 기여하고 지원하고 있습니다.
- 위젯 기반, 선언형, 3D 또는 이것들을 같이 사용하는 등의 멀티 디스플레이 기술 옵션을 지원함으로서, 개발자가 하나의 툴만으로도 복합적인 UX를 만들어낼 수 있게 해줍니다.
- Qt의 강력한 통합 개발 환경(IDE)은 개발자들이 어플리케이션을 쉽게 생산, 분석, 디버깅, 최적화할 수 있도록 해줍니다.
- 기타 자동화 분야 (제조업, 건물 자동화, 산업용 IoT)의 어플리케이션을 위해 여러 프로토콜을 지원합니다.
- 임베디드, 데스크탑, 모바일 장비를 함께 사용하는 경우 외에도, 여러 종류의 보드와 운영 시스템을 지원하는 내장된 크로스-플랫폼입니다.

“크로스-플랫폼 개발이라면 HTML5같은 툴도 적용가능하지 않을까?” 라고 생각할 수 있습니다. Qt가 산업용 어플리케이션 분야에서 HTML5를 월등히 앞서는 솔루션인 이유는 여러 기술적인 배경이 있습니다. Qt 프레임워크는 수십 년간의 탄탄한 개발 경험을 통해 만들어 졌고 그 결과 신뢰성이 뛰어나고 안정적일 수 밖에 없습니다. C++이 가진 원래의 성능을 활용하여, Qt는 부드러운 유저 인터페이스와 끊김 없는 머신 제어를 가능하게 합니다. 그리고 Qt 어플리케이션은 웹 앱보다는 훨씬 더 안전하게 만들 수 있습니다. 이 특징에 대해서도 “The Nuts and Bolts of Qt Industrial Applications (Qt 산업용 어플리케이션의 너트와 볼트)”라는 제목의 eBook에서 보다 상세히 확인하시기 바랍니다. 웹페이지 링크 <https://resources.qt.io/industry-solution-automation>에서 찾아보실 수 있습니다.

Qt는 어떻게 시장의 선도자가 되었는가?

산업용 시장에 영향을 주는 요소들이 전 세계의 소프트웨어에도 동일하게 영향을 주고 있습니다:

- ✓ 커지고 있는 어플리케이션의 복잡성
- ✓ UI의 화려함, 애니메이션, 사용 편리성에 대한 새로운 요구
- ✓ 세계화 및 개인화
- ✓ 여러 다양한 플랫폼을 넘나드는 고성능
- ✓ 기능적인 면에서의 안전성 및 높은 신뢰도
- ✓ 일반적인 플랫폼의 요구를 만족시킬 수 있는 요구사항

Qt는 여러 시장의 다양한 고객들을 보유하고 있기 때문에, 20년이 넘는 기간 동안 세계 곳곳의 다양한 영역에서 나온 개선 사항에 대한 요구를 바탕으로, 그간의 모범 사례와 교훈을 통해 많은 것을 얻었고, 이를 프레임워크에 반영하여 개발해왔습니다.



실제 사례: Parker Hannifin

Parker Hannifin은 모션 및 컨트롤 기술과 시스템의 선두적인 제조 회사로서, 농업, 건설, 삼림, 재료 처리, 교통 분야의 다양한 시장에서 쓰이는 모바일 기계용의 정밀엔진이어링 솔루션을 제공합니다. Parker Hannifin은 자사의 전자 컨트롤 시스템과 계기용 디스플레이를 트럭, 크레인, 버스, 트랙터에 장착하여 판매하는 중장비 OEM 회사들에 직접 제품을 판매 합니다. Parker Hannifin은 OEM 고객들이 장비를 차량 내 디스플레이 장비에 통합시킬 때 소프트웨어의 복잡성으로 인해서 어려움을 겪는다는 것을 알게 되었습니다. 무엇보다, OEM 고객들은 디스플레이를 개별적인 여러 시스템을 위해 대량으로 한번에 설치해야 했기 때문에, 이러한 방식은 사용자가 운영하기에도 어려웠을 뿐 아니라 제조 비용 또한 비싸질 수밖에 없었습니다.

Parker Hannifin은 고객이 보다 사용자 친화적인 인터페이스를 만들고, 그들의 모바일 기계를 시장에 더 빨리 출시 할 수 있도록 해주기 위해서, 고객들로 하여금 이미 내장되어 테스트가 완료된 컴포넌트로 구성된 다양한 조합을 사용해, 고객 어플리케이션을 간단하고 빠르게 조합할 수 있도록 도와 주는 툴을 생산하기로 결정했습니다. 이 툴은 일체화된 유저 인터페이스를 사용하여 싱글 터치 스크린 화면을 통해 트럭이나 트랙터 내의 장비와 인터페이스 할 수 있는 소프트웨어를 생산해낼 수 있어야만 합니다. 또한 사용자가 작업하기에 간단하고 쉬워야 한다는 조건도 만족시켜야 했습니다. Parker Hannifin은 결국 Qt를 사용하기로 결정했는데, 그 이유는 Qt를 사용하면 다양한 커스터마이징이 지원되는 세련된 사용자 인터페이스를 만들 수 있었고, OEM 회사가 사용하기에 쉬운 high-level의 선언형 스크립트 언어(QML)를 지

“QML 프로그래밍은 정말 놀라웠습니다. QML은 사용하기 쉽고 뛰어난 성능을 가지고 있습니다. 처음에도 그렇게 생각되었고 프로젝트를 진행하면서 QML이 우리 소프트웨어의 확장과 기반을 마련해준다는 것을 알게 되었습니다”

Tommi Forsman - 수석연구원, Parker Hannifin

원했으며, J1939 bus 같은 underlying 하드웨어에 쉽게 접근할 수 있기 때문이었습니다.

Parker Hannifin은 Boot to Qt, Qt Creator, Qt 컨설팅 서비스의 조합을 통해 그들의 제품인 Parker Application Designer를 만들었습니다. Parker Hannifin은 Qt IDE에 이미 내재되어 있는 성능과 기술을 잘 활용하여 Qt Creator를 그들의 툴에 직접 통합시켰습니다. 또한 서로 다른 장비 옵션에 개별 맞춤된 어플리케이션 라이브러리와 고객만의 디자인적인 취향을 위해서 스타일 시트와 테마를 일부 제공하는 기술을 추가했습니다. 어플리케이션 기반의 플랫폼을 제공함으로서 하나의 계기용 디스플레이가 각각의 기계 기능을 제어할 수 있게 되었으며 생산에 들어가는 비용을 줄이고 생산 과정은 덜 복잡하게 해주었습니다. 또한 Parker Hannifin 입장에서는 모든 소프트웨어 업데이트, 디버깅, 개발을 위해서 모듈을 하나만 사용하면 된다는 장점도 있습니다. 최종적인 결론은, OEM 회사들이 어려운 프로그래밍을 찾지 않고도, 복잡한 계기용 디스플레이들을 드래그-앤-드롭 방식과 high-level의 QML 스크립트만으로 간단하게 모두 한데 모을 수 있게 되었다는 것입니다.



실제 사례: DMG Mori

DMG Mori는 'Global One Company'로서 머신 툴 분야의 세계적인 선두 기업입니다. 생산 제품으로는 자동화 완전 기술 솔루션과 ULTRASONIC, LASERTEC and ADDITIVE MANUFACTURING과 같은 첨단 기술 분야, 그리고 밀링머신이 있습니다. DMG MORI는 의학 분야 뿐 아니라 우주공학, 자동차, 사출 금형 분야의 첨단 산업에서도 자사의 기술을 발휘하고 있습니다. 'Industrial Service'는 장비를 사용하는 기간 동안의 교육, 수리, 유지보수, 부품 서비스와 같은 고객 중심의 서비스를 제공합니다. 앱 기반의 제어와 운영 소프트웨어인 CELOS 그리고 독보적인 DMG MORI 기술 사이클과 Powertool은 DMG MORI가 Industry 4.0을 이룩할 수 있도록 해주었습니다. 1만개의 DMG MORI 기계 제품은 CELOS와 함께 시장에서 독보적인 제품위치를 차지하고 있습니다.

DMG Mori는 완전한 디지털화와 paper-free 프로세스를 도입하는 것 외에도 작업자, 기술자, 매니저들을 위해서 복잡한 작업 흐름을 간소화할 필요가 있었습니다.

이러한 목표를 달성하기 위해서, DMG Mori는 그들의 모든 장비에 적용하기 위해서 Qt를 사용하여 CELOS라는 이름의 앱 기반 사용자 인터페이스를 만들었습니다. 그들은 디스플

레이를 사용하기 쉽게 만들었고, 또한 매력적인 사용자 인터페이스를 사용하여 사용성을 향상시켰습니다. 심지어 그들의 인터페이스는 이미 고도로 숙련된 작업자들마저도 만족시키는 수준이었습니다. C++의 속도와 안정성을 바탕으로 속도 지연없이 실시간으로 데이터를 처리하고 다룰 수 있게 되었으며, 이러한 점은 정밀한 밀링 장비를 제어하는데 있어 매우 중요한 부분입니다. Qt의 크로스-플랫폼 성능을 이용해서, 그들은 하나의 개발팀 만으로도 PC버전 뿐 아니라 임베디드 버전의 제품을 만드는데 성공했습니다.

소프트웨어 트렌드: 산업계에서 무슨 일이 일어나고 있는가?

일부 사람들은 산업용 어플리케이션으로 개발된 소프트웨어는 지난 몇 년간 전혀 변화하지 않았다고 생각합니다. 물론 이 분야가 고객이 원하는 유행에 쉽게 편승하지 않는 건 사실이지만, 새로운 요구사항들 예를 들면, 새로운 프로토콜에 대한 지원, 차트 및 시각화에 대한 요구, 글로벌한 지역별 맞춤 언어 버전에 대한 요구, 터치스크린과 가상 키보드, 직관적인 대시보드, 원격 브라우저 접근에 대한 고객들의 요청은 급증하고 있습니다. 이러한 기능의 도입을 주도하는 것은 산업용 어플리케이션의 디자인적인 측면에 영향을 주는 다양한 시장트렌드와 이들이 디자인되고 만들어지는 방식입니다.

모바일의 영향

애플의 아이폰과 구글의 안드로이드 스마트폰이 제공하는 주머니 속의 편리함은 쉬운 사용성, 기능성, 신속함에 대한 고객의 기대에 엄청난 영향을 미쳤습니다. 사람들이 모바일을 통해 매일 하는 작업들은 사용자 인터페이스의 바람직한 디자인에 대한 사람들의 세계관을 바꾸어 놓았고, 이는 산업용 어플리케이션에도 예외는 아닙니다.

즉, Qt는 터치스크린, 제스처 방식과 햅틱 컨트롤을 사용한 어플리케이션의 현대화, 앱 스토어 방식의 기능 지원, focus-directing 애니메이션의 도입을 주도했습니다. 개발자들은 Qt를 통해, 마치 일반 사용자들이 사용하는 스마트폰과 같은 인터페이스 스타일의 뛰어난 반응성과 미려한 디자인을 가진 HMI를 만들 수 있게 되었습니다.

장비 통합과 원격 접속

사람들은 산업용 장비도 스마트폰 같은 인터페이스를 가지기를 원하지만, 많은 제조사들은 자신들의 장비와 연동하는 스마트폰이나 태블릿용 어플리케이션을 만들고 있습니다. 또 다른 제조사들은 그들의 장비가 '반응형' 웹 인터페이스를 지원하여 데스크탑이나 모바일에서 별도의 앱을 설치할 필요없이 쉽게 사용될 수 있도록 하기도 합니다. 이러한 보편화된 기술은 사람들로 하여금 자신의 개인 장비를 이용해서 대부분의 산업용 시스템과 커뮤니케이션하는 것을 기대하게 만들며, 결국 더욱 모던하고 접근성이 뛰어난 그리고 개인화를 가능하게 하는 산업용 장비를 제공하는데 도움을 주고 있습니다.

Qt는 크로스-플랫폼을 코드를 이용할 수 있는 기술로 유명합니다. 모바일 companion app을 개발하기 위해서 동일한 코드 베이스로 임베디드 시스템용의 기본적인 인터페이스를 만드는 것이 어렵지 않습니다. 이것은 공통 라이브러리 한 세트를 만드는 데 들어가는 시간을 절약시켜 줄 뿐만 아니라 디자인적인 요소의 공유도 가능하게 합니다. 만약, 여러분이 앱 없이 구동되는 장비-호스트 방식의 웹사이트 방식을 선택한다면, Qt WebGL이나 Qt-compatible WebAssembly를 사용하여 원격 브라우저를 지원하는 인터페이스를 여러분의 산업용 앱에 쉽게 넣을 수 있습니다.

장비와 산업용 어플리케이션

하드웨어 제어를 위해서 companion app이나 리모트-컨트롤 웹 접근을 생성하는 것은 언제 필요한가요? 제조사들이 왜 그들의 산업용 어플리케이션을 개인 장비와 연동하는지에 대해 몇 가지 이유를 설명드리겠습니다.

- 접근이 불편한 디스플레이 - 온도조절장치, HVAC 컨트롤, 알람 시스템
- 복잡한 설치 및 설정 - 콘서트 훌 오디오 시스템, 실험실 자동화, CNC 기계
- 과거 방식의 또는 현존하지 않는 디스플레이 - 게이트웨이, 펌프, 노변 센서
- 복잡한 진단 보고에 대한 요구 - 자동차 시스템, 파워 플랜트, 생산 라인
- 모바일로 보고자 하는 요구 - 의학 장비, 보안 시스템, 위험물 처리 로봇

⌚ 시장 출시 타이밍을 맞추는 어려움

가전제품과 모바일 앱의 업데이트는 몇 주 또는 며칠에 한번 진행되는 릴리즈 사이클로 그 주기가 빠릅니다. 이는 고객들이 중요한 버그 수정이나 기능 추가를 위해서 6개월이나 1년씩 기다리기 어렵게 만듭니다. 이러한 시간적인 압박은 개발 사이클을 단축시킬 뿐만 아니라 OTA(over-the-air) 소프트웨어에 대한 필요성을 대두시키기도 합니다.

Qt는 프로토타입을 신속하게 제작하는데 있어서 가장 적합한 솔루션이라 할 수 있는데, 이는 Qt가 보유한 Qt Quick과 QML 덕분입니다. 물론 개발 사이클의 단축과 동시에 품질도 그대로 유지할 수 있는 특효약은 존재하지 않지만, Qt는 개발자들이 빠른 시간 내에 코딩을 진행하여 쉽게 개발, 테스트, 디버깅할 수 있도록 해주는 강력하고 빠른 프레임워크를 가지고 있습니다.

개발 사이클의 시간 단축에 있어 또 다른 눈에 띠는 부분은 디자이너와 개발자 사이의 작업 흐름을 개선시켜 주는 점입니다. Qt Design Studio와 Qt 3D Studio를 통해, 사용자 인터페이스 디자이너들은 사용한 강력한 비주얼 툴을 사용하여 HMI를 만들 수 있으며, 이러한 내부 결과물은 개발자들이 곧바로 가져다 사용할 수 있습니다. 이는 디자이너와 개발자 사이에서 서로 상이하고 호환되지 않는 툴로 인해 여러 번 반복되는 힘든 과정을 줄여줄 수 있습니다.

☑ 상용 라이선스 / 오픈 소스 라이선스

개발자들에게 있어 새로운 기능이 필요할 때마다 매번 그 기능을 만들어 내는 것은 비합리적인 일이며, 이는 새로운 어플리케이션의 개발 비용에도 영향을 주고 또한 누군가가 이미 수정했을 버그를 만들어낼 수도 있기 때문에 굉장히 비효율적이라 할 수 있습니다. 그렇기 때문에 많은 프로젝트들이 타사의 라이브러리, 컴포넌트, 상용 라이선스 어플리케이션의 COTS(commercial off-the-shelf)나 오픈 소스를 사용합니다.

거의 모든 오픈 소스 라이브러리는 언어의 종류와는 상관없이, C/C++ API를 제공하는데 이러한 사실은 Qt로 하여금 이 API들과 쉽게 인터페이스할 수 있도록 해줍니다. 또한 오픈 소스와 저작권이 있는 소스 코드가 함께 사용될 때의 복잡함을 관리하는데 도움을 주기 위해서, Qt 프레임워크는 다양한 소프트웨어 라이센스와 함께 사용이 가능하며 여러 일반적인 라이센스 호환성 툴과 이미 통합작업이 완료되어 있습니다.





☑ 클라우드 컴퓨팅

클라우드 전송 기능은 산업용 어플리케이션에서 논의되는 끊임없는 주제이며 고객들은 기본적으로 두 가지 이유에서 클라우드 기능을 요구합니다. 첫 번째는, 장비에 배포된 모든 하드웨어에 동시에 버그 수정과 기능 업데이트를 할 수 있기 때문이며, 이는 클라우드 서버를 업데이트하면 고객이 방대한 양의 IT 데이터를 직접 건드리거나 관리할 필요없이 현장에 설치되어 있는 모든 제품에 대해서 업데이트를 자동으로 진행할 수 있기 때문입니다. 두 번째 이유는, 클라우드에 데이터를 두면 모든 고객 사이트에 설치된 제품에 해당 데이터가 손쉽게 공유될 수 있으며 동시에, 동기화와 복제 문제를 획기적으로 단순화시킬 수 있기 때문입니다.

Qt는 REST, SOAP/WSDL API와 같은 대부분의 클라우드 서비스와 데이터 유형에 접근 가능하도록 구현되어 있습니다. Qt내에서 직접 사용할 수 없는 것은 대부분 C/C++ API를 통해 쉽게 연동할 수 있습니다.

☑ 데이터 시각화

많은 산업용 시스템은 방대한 양의 데이터를 만들어내므로 이러한 데이터를 분석하는 것은 어렵고, 지루하고, 특별한 교육을 필요로 하기도 합니다. 결과적으로, 많은 시스템들은 문제를 발견하고, 패턴을 파악하고, 비효율적인 부분을 대신 찾아줄 수 있는 방법을 제공해주는 데이터 시각화에 점점 의존하고 있으며, 특히 미리 예측 가능한 유지보수 작업일수록 더욱 그렇습니다.

Qt는 기본적으로 막대 차트, 분포 다이어그램, 표면 메쉬를 활용하여 데이터를 3D로 시각화해주는 전용 라이브러리를 제공합니다. 다양한 샘플들을 통해 독창적인 방법으로 데이터 시각화 라이브러리를 사용하는 방법을 보여줍니다.

클라우드 기반의 산업은 모두를 위한 것입니다.

클라우드 기반의 산업용 앱이 주는 이익은 단순히 고객만을 위한 것은 아닙니다. 고객 데이터에 지속적으로 접근할 수 있다는 측면은 제조사에게도 여러 가지 면에서 중요합니다.

- 제품의 개선 및 고객 요구의 예측을 위해, 어플리케이션이 어떻게 사용되고 있는지에 대한 파악
- 익명으로 수집되는 데이터 또는 수집된 데이터의 분석에 기반한 새로운 제품
- 예측 가능한 유지보수 및 보고와 같은 데이터 기반의 서비스 - 이러한 서비스를 점차 규모를 확장하거나 새로운 수입원으로 사용할 수 있는 가능성

☑ 자유로운 운영체제 플랫폼의 선택

데스크탑에는 Windows, Mac, Linux 세 가지가 있습니다. 모바일에는 iOS와 안드로이드가 대표적입니다. 임베디드 하드웨어는 해변의 모래 알갱이 만큼이나 많이 있습니다. 어떤 식으로 보든, 여러분 스스로를 하나의 하드웨어나 OS 플랫폼에 한정시키는 것은 오늘날에는 스스로 고립을 자초하는 무모한 행동입니다. 플랫폼에 독립적이면서도 마이그레이션에 변화가 거의 필요 없는 유연한 소프트웨어를 만드는 것은 위험 요소를 최대한 줄이고 최대한의 이익을 가져갈 수 있는 길입니다.

Qt는 크로스-플랫폼 코드 생산에 있어서 매우 뛰어납니다. 이것은 Qt가 가장 유명한 이유 중 하나입니다. Qt는 모든 데스크탑과 모바일 변이형을 동등하게 지원하며 이는 어떠한 새로운 SOC(system-on-chip)나 GPU(Graphics processing unit)에서도 사용 가능한 첫번째 그래픽 프레임워크입니다.



BOM(Bill of materials) 비용

아무리 특별한 훈련을 받았다 하더라도, 누구라도 비용에 대한 걱정을 안할 수는 없습니다. 글로벌 시장은 생산 비용이 낮은 국가들이 생산하는 제품의 가격을 기준의 높은 마진을 추구하는 나라들에게도 계속적으로 확인시켜 주며 세계를 좁혀나가고 있습니다. 이것은 아무리 제품에 차별화된 특징적인 장점이 있다고 하더라도 세계시장에서 경쟁하기 위해서라면 가격을 낮춰야 한다는 압박은 피할 수 없다는 것을 의미합니다.

Qt를 사용한다고 해서 마법처럼 원가를 절감할 수 있는 것은 아닙니다. 혹시, 가능할까요? 사실, Qt는 최적화된 성능의 코드 생산 및 최소화된 데이터 사이즈의 측면에 있어 다른 프레임워크에 비해 월등히 뛰어납니다. 이는 Qt를 사용하면 다른 낭비적 요소가 많은 프레임워크를 사용하여 만든 어플리케이션에 비해, CPU 전력을 덜 소모하고, RAM 칩의 개수를 줄이고, 더 작은 플래쉬 디스크를 사용할 수 있다는 것을 의미합니다.

개발자 편의

개발자들은 이전보다 더 많은 결과물을 내야하며 물론 버그는 더 적어야 합니다. 효율성이 가장 중요한 덕목입니다.

Qt 코드는 그것 자체만으로 모든 것이 설명되지는 않습니다. 그러나, Qt는 굉장히 강력하며 개발자가 간단한 선언만으로도 프로그래밍을 가능하게 해 시간과 노력을 절약할 수 있게 해줍니다.

산업용 시장이 수 천개의 여러 어플리케이션 영역으로 세분화되어 있기는 하지만, 원격 접근성, 통신, 일반 사용자 수준의 인터페이스 등에 대한 사람들의 요구로 인해 자동화에 대한 전반적인 관심은 최근 몇 년간 급격히 커졌습니다. Qt와 같은 크로스-플랫폼 어플리케이션 프레임워크는 상대적으로 배우고 처리하기 쉬운 언어로서, 이러한 시장의 트렌드와 과제를 명확하게 다루고 있습니다. Qt가 다루는 산업의 과제에 대해 더 자세하게 알고 싶으시면, 우리가 발행한 eBook 'The Nuts and Bolts of Qt Industrial Applications (Qt 산업용 어플리케이션의 너트와 볼트)'를 읽어보시기 바랍니다. 웹페이지 링크 <https://resources.qt.io/industry-solution-automation>에서 찾아보실 수 있습니다.

Bolts of Qt Industrial Applications".

개발자가 부족한가요?

Qt 경험이 있는 개발자를 찾는 것은 걱정할 필요가 없습니다. 70개 이상의 산업 영역에서 백만명 이상의 개발자들이 현재 Qt를 사용하고 있습니다.



더 큐티 컴퍼니(The Qt Company)는 Qt 개발 프레임워크(Qt Framework)를 상용 라이선스와 오픈소스 라이선스로 개발 및 배포하고 있습니다. Qt 개발 프레임워크를 이용하면 데스크탑과 임베디드 그리고 모바일용 앱에 이르기 까지 다양한 운영체제와 플랫폼에 걸쳐 동작할 수 있는 애플리케이션을 제작할 수 있습니다. 전세계에서 백만명이 넘는 Qt 개발자들이 차량용 인포테인먼트 시스템(M), 자동화 시스템(Automation), 의료(Medical), 가전 제품 및 군사용 소프트웨어에 이르기 까지 70 여종류의 다양한 산업군에서 핵심적인 소프트웨어 개발에 Qt 개발 프레임워크를 사용하고 있습니다. The Qt Company 는 핀란드에 본사를 두고 있으며, 대한민국을 비롯해 미국, 독일, 일본, 중국, 러시아 등에서 지사와 연구소를 운영하고 있습니다. Qt 상용라이선스 문의는 Qt 한국지사 전화번호 **031-712-0045** 혹은 이메일 **korea@qt.io** 로 문의주시기 바랍니다. Qt와 관련한 보다 자세한 안내는 웹페이지 www.qt.io 에서 확인 하실 수 있습니다.