

HMI-ENTWICKLUNG FÜR INDUSTRIELLE ANWENDUNGEN

HMI-Entwicklerforum – 11/2019 – Stuttgart

AGENDA

- Über uns
- Einführung & Kontext
- Szenarien: Plattformunabhängigkeit
- Szenarien: Offenheit
- Szenarien: Vernetzung
- Zusammenfassung & Ausblick



ÜBER UNS



Über basysKom

Software Engineering Services

HMI und Anwendungsentwicklung

- Technologieorientiert

**Typischer Kunde: Maschinenbau, Messtechnik,
Automatisierung (DACH)**

Gegründet 2004

30 Mitarbeiter

Darmstadt und Nürnberg

Ein Unternehmen der UX Gruppe

Open-Source Aktivitäten

- Qt
- open62541
- Qt OPC UA

EINFÜHRUNG



Kontext

Was bedeutet „HMI-Entwicklung für Industrielle Anwendungen“ im Rahmen dieser Präsentation?

- Individual-Software nicht „Baukasten/Projektierung“

Einsatz von Projektierungswerkzeugen kann Sinn machen wenn...

- es ein (sehr) einfaches HMI ist (aka. Visu)
- die Anwendung gut in das Feature-Set von Projektierungswerkzeugen passt
- man in kurzer Zeit HMIs für individuelle Anlagen projektieren muss (System-Integration)
- man keine Software-Entwickler hat

Individual-Software macht Sinn...

- für komplexe Anwendungen
- hohe Stückzahlen mit dem gleichen HMI
- HMI-Varianten für eine Reihe von verwandten Maschinen
- man mehr Flexibilität/Kontrolle benötigt

Motivation

Aktuelle Themen

- Usability / Nutzbarkeit
- Gestiegene Ansprüche an das Aussehen
- Plattformunabhängigkeit (Embedded/Lokal, Apps, Remote-Displays, ...)
- Offenheit (Scripting durch den Kunden, Externe Anwendungen wie Matlab/Simulink, ...)
- Vernetzung (Cloud, Leitrechner, SCADA, Vernetzte HMI, ...)

Fragestellung

- Was bedeutet das für die eingesetzte Technologie?
- Was bedeutet das für meine System-/Software-Architektur?

Technologischer Kontext

Qt und Webtechnologie als die beiden dominanten Ansätze

- Äpfel und Birnen
- Beide Plattformunabhängig
- Beide FOSS

Weitere Präsentationen:

- **Qt Quick oder Webtechnologie?**
Zwei HMI-Stacks im Vergleich
- **Qt und das Web**
 - Aus der Perspektive von Qt

Showcase Industrial HMI

Messe Showcase (und Labor)

- Als Dienstleister trotz NDAs etwas zeigen können
- Aktuell von der Steuerung bis ins HMI → OPC UA
- HMI als reine „Viewer“
- Mehrere lokale HMIs (Qt und Web)
- Azure-Cloud-Anbindung
- Datenerfassung und Angular-HMI auf Azure



SZENARIEN: PLATTFORM- UNABHÄNGIGKEIT



Warum Plattformunabhängigkeit?

- Gerätevarianten (die große Version mit Windows10, die kleine Embedded-Linux auf i.MX6)
- Apps ohne zwei weitere vollständig unabhängige Code-Basen zu entwickeln?
- Gemeinsame Code-Basis für Embedded/Desktop/Mobile (zumindest in Teilen)
- Zukunftssicherheit

Embedded HMI + Apps

Wunsch

- Wir hätten gerne noch eine Android/iOS App zusätzlich zum lokalen HMI (oder gar statt dem lokalen HMI)
- Mobiles Bedienteil, Rüst-Tablet, etc... (lokale Nutzung)

Frage

- Macht die App 1:1 das Gleiche wie das lokale HMI?
Wie groß ist die Schnittmenge?
- Soll das Styling gleich sein?
- Einsatz im Umfeld der Maschine oder weltweit?

Lösungsmöglichkeit Qt

- Lokales HMI mit Qt Quick (zb. Embedded-Linux)
- 1:1 das selbe HMI für iOS/Android Tablets
- Kommunikation per OPC UA

Zu beachten

- Smartphones erfordern mehr Arbeit (Formfaktor!)
- Wichtig in der Architektur: das HMI darf nur ein „Viewer“ sein

Showcase: Embedded Linux + Android

Embedded HMI + Apps

Lösungsmöglichkeit Web

- Embedded HMI per
 - Single Page Application mit Angular
 - Standard-Browser (oder für QWebEngine für Embedded)
- Apps über Paketierung per Cordova
- Kommunikation per REST/WebSocket API mit der Anwendung
- Entkopplung von Business-Logik und HMI der Default

Zu beachten

- Setzt für den Embedded-Fall oft stärkere Hardware voraus
- Ggfls. müssen interne Protokolle (z.B. OPC UA) auf Web-orientierte Protokolle umgesetzt werden
- Cordova: Zugriff auf Features der mobilen Plattformen oft schwierig/inkonsistent
- Smartphones erfordern auch hier mehr Arbeit (Formfaktor!)

Lokales HMI + Browser

Wunsch

- Wir hätten gerne noch ein Webinterface zusätzlich zum lokalen HMI
- „Man soll nichts installieren müssen“
- Nutzung aus dem Umfeld der Maschine
- Variante: Komplette „Headless“

Lösungsmöglichkeit Qt

- Umsetzung einmal in Qt Quick, einmal in Web-Technologie? Hmm...

Umsetzung in Qt Quick + x

- WebGL-Streaming
- VNC + WebSocketProxy + NoVNC?
- WebAssembly

Wichtig: fühlt sich nicht wie eine Web-Anwendung an!

Showcase: WebGL-Streaming

Lokales HMI + Browser

Wunsch

- Wir hätten gerne noch ein Webinterface zusätzlich zum lokalen HMI
- „Man soll nichts installieren müssen“
- Nutzung aus dem Umfeld der Maschine
- Variante: Komplette „Headless“

Lösungsmöglichkeit Web

- Heimspiel für die Web-Lösung
- Rahmenbedingungen vgl. vorheriges Szenario

Showcase

- REST/WebSocket-Gateway + Embedded-WebServer + Angular HMI

SZENARIEN: OFFENHEIT



Integration mit Drittanwendungen/Kundenanwendungen

Wunsch

- Wir würden gerne ein Matlab/Simulink (z.B) anbinden
- Kunden sollen selbst eine Integration schreiben dürfen

Fragestellungen

- Plugins? Achtung SDK!
- Scripting-API im Prozess?
- Extern?

Externe Schnittstellen

- Welches Serialisierungsformat?
- JSON/XML/Protobuf, ...
- Mit oder ohne Schema?
- REST/WebSocket-API (Heimspiel)
- RPC-Dienste (D-BUS, GRPC, Thrift, ...)
- OPC UA?

Showcase: Zugriff per UaExpert

SZENARIEN: VERNETZUNG



Vernetzung von HMIs

Wunsch

- Es gibt mehrere HMIs (Instanzen des gleichen oder verwandte HMIs) die alle auf der gleichen Datenbasis arbeiten sollen (Aufträge, Werkzeuge, Messdaten, ...)

Klassisch als Client/Server-System?

Pub/Sub mit oder ohne Broker?

Showcase

- OPC UA Server der Daten und Anwendungslogik enthält
- Clients müssen explizit Änderungen überwachen (lassen)

Die „Cloud“

Was bedeutet Cloud?

- Entwicklung einer eigenen „Cloud-Anwendung“?
- Auch hier wieder Baukasten vs. Individual-Software
- Integration in existierende Infrastruktur? Welche?

Welche Kunden haben überhaupt Interesse an einer Cloud-Anwendung?

- Welchen Mehrwert kann man bieten?

Daumenregel

- Je kleiner der typische Käufer, umso weniger existierende Infrastruktur, umso mehr kann eine „Cloud-Anwendung“ eine Option sein.

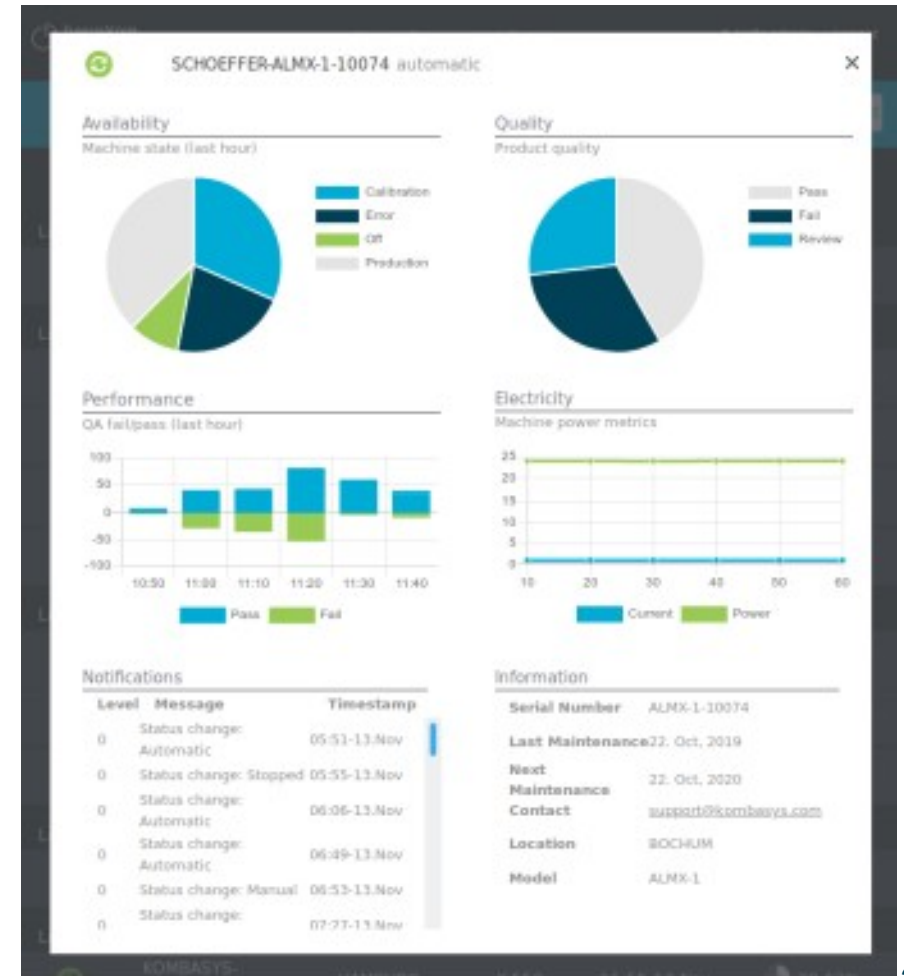
Fallstrick

- Was ist mit den „Cloud-Anwendungen“ der „Anderen“?

Die „Cloud“ in unserem Showcase

Monitoring-Anwendung

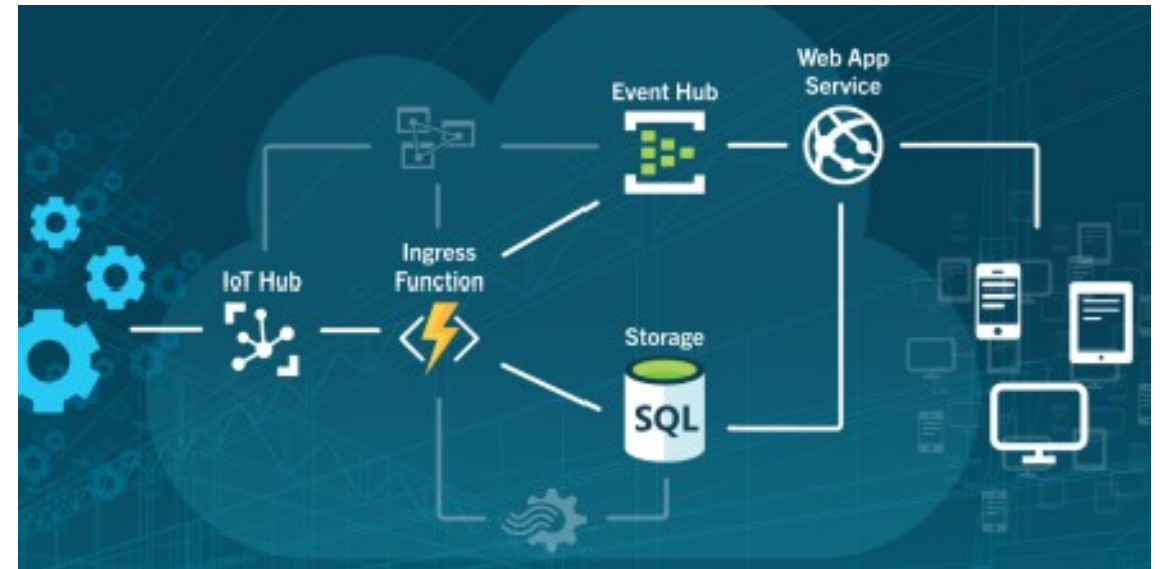
- Anzeige von Kenndaten
- Logs und Fehlerzustände
- Push von Events in die Web-Clients



Die „Cloud“ in unserem Showcase

Umsetzung auf Azure

- Anbindung/Provisionierung per iotHub (Azure-IOT-SDK)
- Telemetriedaten über das MQTT-Interface des iotHub (Standard MQTT-Client/Azure-IOT-SDK)
- Ingress-Verarbeitung per Azure-Function
- Persistenz per SQL
- REST-API in einem Web App Service (node.js)
- HMI in Angular



Nächster Schritt: OPC UA direkt in die Cloud?

ZUSAMMEN- FASSUNG & AUSBLICK



Zusammenfassung & Ausblick

Plattformunabhängigkeit, Offenheit und Vernetzung als Herausforderung in der Umsetzung von HMIs

Weitere Vorträge

- Qt Quick oder Webtechnologie – HMI-Stacks im Vergleich
- Qt und das Web – ein Überblick
- OPC UA Anwendungsentwicklung mit open62541

Besuchen Sie uns in der Ausstellung

- Die Showcase Industrial HMI kann ausprobiert werden
- Wir freuen uns auf interessante Gespräche rund um ihre HMI-Themen

DANKE SCHÖN!

FRAGEN?

спасибо 谢谢
GRACIAS 谢谢
THANK YOU
ありがとうございました MERCI
DANKE धन्यवाद
شُكْرًا OBRIGADO