

► Systemaufbau für die Anforderungen an eine plattformunabhängige Web-HMI.

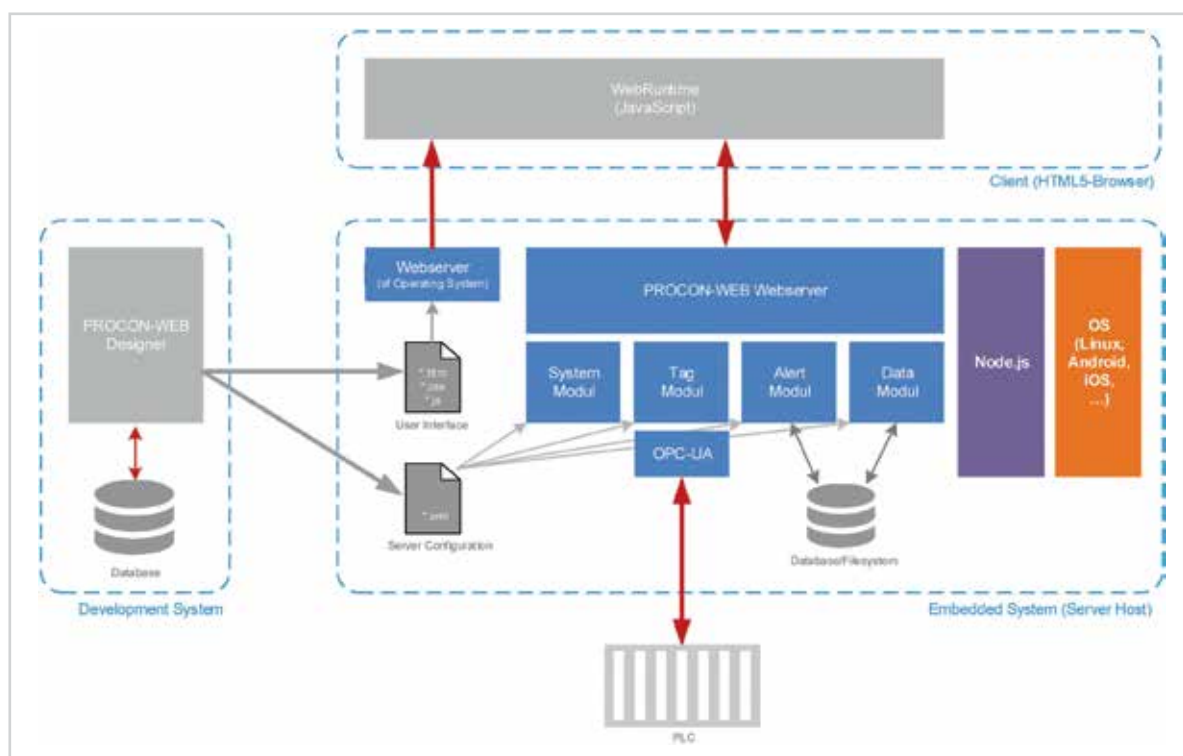


Bild: GTI-Control

# Unabhängig von der Plattform

**Web-HMI** Günstige Devices halten Einzug in die vernetzte Automatisierungswelt. Die Softwaremodule für diese Geräte sollten daher modular und portierbar implementiert werden, um auf verschiedenen Plattformen übertragbar zu sein. Dank Node.js und HTML5 können vollständig übertragbare HMIs für embedded Geräte jetzt realisiert werden.

**Frank Stegerwald\***



**Sariana Kunze,**  
Redakteurin  
sariana.kunze@vogel.de

Anschaulich vorgeführt wird die webbasierte HMI Procon-Web in diesem Video: <http://bit.ly/2fFzxD>

**A**ktuell unterliegt die Automatisierungs- und Softwarewelt einem enormen Wandel. Bisher war ein Standard-PC mit einem Microsoft Betriebssystem als Basis meist die Grundlage für Softwarelösungen in der Automatisierungstechnik. Diese Basis wurde innerhalb kurzer Zeit zum einen auf

\*Frank Stegerwald, Leiter Softwareentwicklung, GTI-Control

der Client-Seite durch mobile Geräte in Form von Smartphones und Tablets ergänzt und auf der Server-Seite gibt es inzwischen eine große Palette an sehr leistungsfähigen Devices zu geringen Kosten (z.B. Einplatinencomputer wie Raspberry Pi). Die Softwaremodule für diese Geräte sollten daher möglichst modular und portierbar implementiert werden, um die Möglichkeit zu schaffen, auf verschiedensten Plattformen wenn möglich ohne Portie-

rungsaufwand übertragen zu werden. Um Bedienoberflächen plattformunabhängig darzustellen, sollte die Visualisierung heute in HTML5 und JavaScript ausgeführt implementiert werden, um diese in jedem beliebigen Browser sowohl auf beliebigen PCs und Notebooks, aber auch auf Panels, Smartphones und Tablets darstellen zu können, unabhängig vom Betriebssystem. Durch die Beschränkung auf HTML5 und JavaScript werden alle moder-

nen Browser unterstützt, da keine spezifischen Plugins notwendig sind. Die notwendigen Server-Komponenten für die zentrale Datenhaltung und Verbindung zu externen Komponenten wurden allerdings bisher plattformspezifisch realisiert und konnten nicht einfach auf eine neue Plattform portiert werden. Es waren entweder Anpassungen an den Komponenten oder gar eine Neuimplementierung der Serverkomponenten notwendig. So wurde z.B. eine komplett eigenständige Implementierung in IEC 61131 für Codesys V2 und V3 Plattformen eingebaut. Die Web-HMI Procon-Web von GTI-Control wurde als reine Web-Visualisierung konzipiert und stellt nun auch die Serverkomponenten in einer plattformunabhängigen Variante zur Verfügung. Eine plattformunabhängige Web-HMI muss den folgenden Anforderungen genügen: Projektierung aller Komponenten über ein einheitliches Engineeringtool ohne Programmierkenntnisse, Bedienoberflächen als HTML5-/JavaScript-Anwendung vollständig plattformunabhängig, portable und skalierbare Serverkomponenten für beliebige Betriebssysteme, Anbindung an den Prozess über OPC UA und alternativ proprietäre Interfaces, geringe Hardwareanforderungen an den Server-Host, Erweiterbarkeit des Systems um benutzerdefinierte Bedienelemente oder Serverfunktionalitäten.

Durch einen Designer wird das entsprechende Visualisierungsprojekt erstellt. Dieser Designer unterscheidet sich in seiner Bedienung und Handhabung nicht von den klassischen Projektierungswerkzeugen, der Anwender muss bei der Erstellung keinerlei Kenntnisse der Webtechnologien besitzen. Das Tool generiert im Hintergrund direkt HTML5 und JavaScript-Dateien, die die komplette Oberfläche mit allen Dynamisierungen beschreiben. Spezielle Erweiterungen erlauben die Konfiguration dynamischer Seiten mit Bildwechsel über Wischen oder das Nutzen von Animationseffekten z.B. beim Verschieben von Objekten. Diese Dateien werden dann vom Webserver zur Laufzeit dem Client zur Verfügung gestellt, damit diese im Browser des Clients dargestellt und ausgeführt werden können. Der Webserver ist gleichzeitig die gemeinsame Schnittstelle zwischen den Clients und den Funktionsservern. Die Funktionsserver versorgen die Client-Oberflächen mit den spezifischen Daten: Der Systemserver übernimmt Systemaufgaben, der Tagserver ist für die Kommunikation zum Prozess zuständig, der Alertserver löst

Alarmer und Meldungen basierend auf Prozessstatus aus und protokolliert Benutzereingaben, der Datenserver stellt Funktionen zum Zugriff auf externe Datenbanken zur Verfügung, bietet zyklische oder ereignisgesteuerte Prozesswertprotokollierung und stellt eine Rezeptur-, Format- und Auftragsverwaltung zur Verfügung.

#### **Node.js für beliebige Hardware-Systeme**

Um die Software für möglichst viele Hardware-Plattformen ohne Portierungsaufwände zur Verfügung stellen zu können, werden die Server-Komponenten für eine Software-Plattform implementiert, die die entsprechende Hardwareplattform abstrahiert. Hier kommt als Basis node.js zum Einsatz. Node.js als serverseitige Plattform ist für die wichtigsten Hardwareplattformen direkt verfügbar (Windows Client und Server-Betriebssysteme, Mac OS, Linux, SunOS, Android, ...) und im Quellcode verfügbar, so dass eine Anpassung an weitere Systeme mit geringem Aufwand möglich ist. Die Programmiersprache für Node.js-Anwendungen ist JavaScript, in der die Bedienoberflächen der Web-HMI codiert sind. Somit lässt sich das Entwicklungs-Knowhow von der Client-Seite auf die Serverkomponenten übertragen und die komplette Entwicklung der Runtime-Umgebung kann in der gleichen Umgebung mit den gleichen Tools und Sprach-Konstrukten erfolgen.

#### **OPC UA als universelle Prozessanbindung**

Für die Kommunikation zum Prozess wird eine Standardimplementierung der OPC UA Schnittstelle gestellt. Damit ist das System in der Lage, an alle modernen Steuerungen anzuknüpfen, ohne dass ein Zusatzaufwand an Programmierung anfällt. Das System ist für eine Großzahl an Systemkonstellationen „Plug&Play“-fähig verfügbar ohne Portierungsaufwand. Für Spezial-Ankopplungen an bestimmte Steuerungssysteme können eigenständige Module eingesetzt werden, die ein proprietäres Interface implementieren und die externen Systeme nahtlos in die Systemlandschaft integrieren.

Mit einer Web-HMI bieten sich neben der Implementierung der reinen Maschinenfunktionalität noch viel weitere Möglichkeiten. Da die Oberflächen auf jedem Gerät dargestellt werden können, ist man in der Lage, die Features der Maschine modular als Web-Apps auf unterschiedliche Nutzerkreise zugeschnitten zur Verfügung zu stellen. [kun]