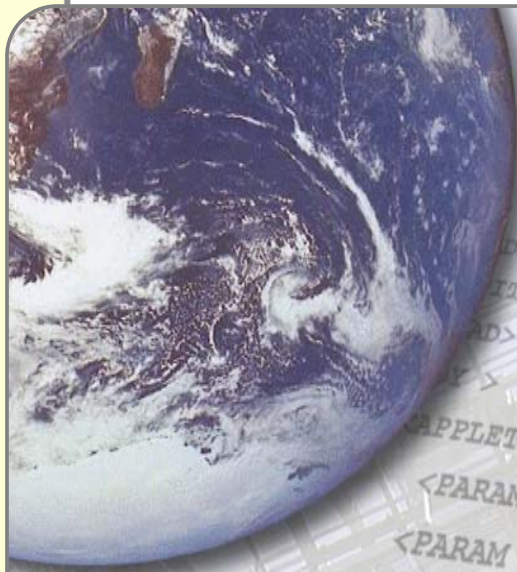


JavaPanel

LabVIEW-Frontpanels

auf

Java-Plattformen



HTML code snippets overlaid on the background:

```
<D>  
<TITLE>Technologievi...</TITLE>  
<AD>  
<Y>  
<APPLET code=JavaPanel.class width=750 height=500>  
<PARAM name="archive" value="JavaPanel.jar">  
<PARAM name="src" value="Technologie.vi">  
<PARAM name="ImagePort" value="1234">  
<PARAM name="EventPort" value="1235">  
<PARAM name="Timeout" value="60">  
</APPLET>  
</BODY>  
</HTML>
```

Two overlapping screenshots of a JavaPanel interface. The interface displays a complex industrial process flow diagram with various tanks, pipes, and control elements. Key data points visible include:

- Weight: 0,0 kg
- Percentage: 100,0 %
- Current: 0,0 U
- Amperage: 0,00 A
- Buttons: Test, Wechseln zu Link
- Labels: 15: Entleeren, Restzeit: 0 s
- Other labels: 15: Entleeren, Restzeit: 0 s, 40, 15: Entleeren, Restzeit: 0 s

Webtechnologien in verteilten Anwendungen

Motivation

Der Einsatz der Webtechnologien und der zugrunde liegenden Webarchitektur bietet eine Reihe von Vorteilen gegenüber der klassischen Client-/Server-Architektur.

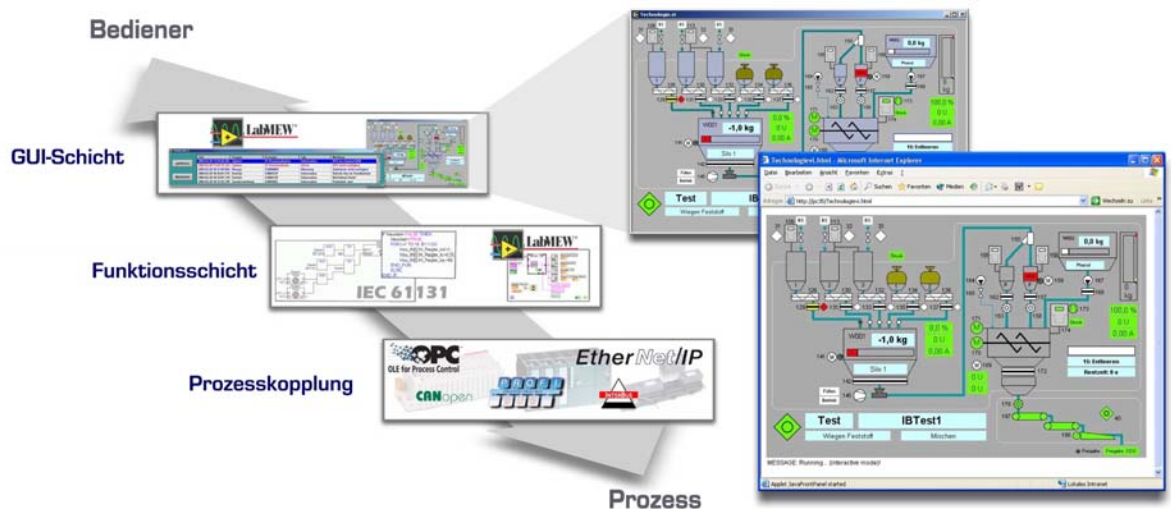
- ▶ Spezielle, meist plattformabhängige Clientprogramme werden durch einen Standard-Webbrowser ersetzt.
- ▶ Es gibt keine Verteilungsprobleme mehr, da keine anwendungsspezifische Software auf dem Client installiert werden muss.
- ▶ Die Komponenten der Applikation liegen nur noch auf einem oder wenigen Servern, wodurch sich die Wartung vereinfacht.
- ▶ Der Zugang zum Anwendungssystem vereinfacht sich, da ohne eigenes Zutun bereits weltweit die Infrastruktur vorhanden ist.

In den bestehenden Trend zu dezentralen Automatisierungslösungen mit Ethernet fügen sich die Webtechnologien hervorragend ein. Das Resultat dieser Entwicklungen ist eine durchgängig einheitliche und standardisierte Kommunikationsarchitektur basierend auf bereits erprobten, robusten und allgemein akzeptierten Standards.

- ▶▶ Der Einsatz der Webtechnologie verleiht den industriellen Kommunikationsstrukturen mehr Transparenz und überwindet derzeit vorhandene Kommunikationsbarrieren. Automatisierungsaufgaben können leicht auf andere Plattformen portiert und quasi weltweit verteilt werden.

Webfähige Applikationen mit LabVIEW und JavaPanel

Eine Applikation besteht aus mehreren einzelnen, untereinander abgegrenzten und in Schichten angeordneten Softwarekomponenten. Soll eine derart aufgebaute Applikation webfähig werden, tritt anstelle der GUI-Komponente der Webbrowser auf einem entfernten Rechner. Die Struktur der Applikation bleibt unbeeinflusst. Bei LabVIEW-Anwendungen (VIs - Virtuelle Instrumente) bildet meist das Frontpanel eines speziellen VIs in der obersten Applikationsschicht das GUI der gesamten Applikation. Die Funktionen dieses Frontpanels muss nun der Webbrowser übernehmen. So kann der Benutzer den Prozess über das Web beobachten und bedienen.



▶▶ LabVIEW-Webserver Der Grundbaustein

Einstiegspunkt für jede Webanwendung ist ein Webserver. Seit der Version 6i ist standardmäßig ein Webserver in LabVIEW integriert. Damit ist man bereits in der Lage, sich die in den Verzeichnissen des Webserver gespeicherten HTML-Seiten mittels Browser anzeigen zu lassen.

▶▶ Java - Schlüsseltechnologie im Internet

Java hat sich zur Schlüsseltechnologie im Internet entwickelt, weil es den Anforderungen in Bezug auf Sicherheit, Stabilität und Plattformunabhängigkeit voll genügt. Mit einem Webbrowser als Java-Laufzeitumgebung steht nahezu der gesamte Sprachumfang dieser modernen und objektorientierten Sprache in Form eines Applets zur Verfügung. Damit gelingt die Darstellung und Bedienung von LabVIEW-Frontpanels auf allen Java-Plattformen weltweit. Das Java-Applet übernimmt folgende Funktionen:

- ▶ Aufbau der Verbindung mit dem Web-Interface auf dem Applikationsserver,
- ▶ Darstellung des Frontpanels (GUI) im Webbrowser,
- ▶ Entgegennahme und Weiterleitung von Bedienhandlungen zum Web-Interface.

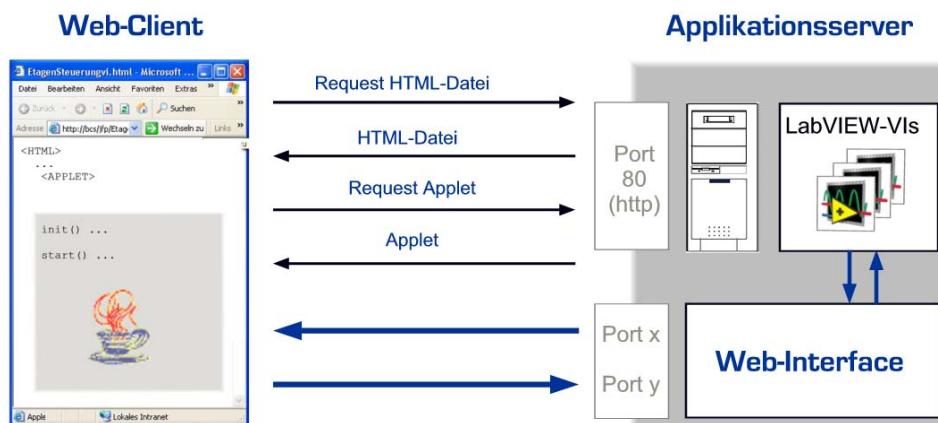
▶▶ ObjectVIEW - Basis für skalierbare Anwendungen

Mit dem Web-Interface auf dem Applikationsserver ist der entfernte Zugriff auf jedes VI über das Web möglich. Die Umsetzung erfolgte mit ObjectVIEW, die Basis für skalierbare Anwendungen. ObjectVIEW basiert vollständig auf LabVIEW, so dass die gewohnte Laufzeitstabilität sichergestellt ist. Das serverseitige Web-Interface realisiert folgende Aufgaben:

- ▶ Feststellung eingehender Verbindungen,
- ▶ Zugriffskontrolle,
- ▶ Handling der Verbindungen zwischen Client und VI,
- ▶ Publikation von VIs für den entfernten Zugriff.

Funktionsprinzip

Für jedes veröffentlichte VI existiert eine HTML-Datei. Durch Eingabe der URL im Browser wird diese über den Webserver in den Browser geladen. Das Applettag signalisiert dem Browser, dass weiterhin ein Applet zu laden ist. Das Applet übernimmt die VI-spezifischen Parameter aus der HTML-Seite und baut die Verbindung zum Web-Interface auf. Es können nun das jeweils aktuelle Frontpanel empfangen und Bedienhandlungen gesendet werden.



Nutzung der Internet-/Intranetressourcen

▶▶ Mit Standardtechnologien kostengünstig zu skalierbaren Anwendungen in heterogenen Umgebungen

JavaPanel ermöglicht die Beobachtung und Bedienung von LabVIEW-Frontpanels im Standard-Webbrowser sowie auf allen Plattformen mit Java-Unterstützung. Diese Möglichkeit ist die Basis für skalierbare Anwendungen in heterogenen Umgebungen bei minimalem Kostenaufwand. Aktuelle Zustände von Systemen sind weltweit erfass- und beeinflussbar.

▶▶ Transparenz - Keine zusätzliche Programmierung für den Client

Die Clientseite bleibt bei der Erstellung einer Webanwendung mit JavaPanel transparent. Es ist kein Programmieraufwand für den Client nötig. Ohne eine Zeile HTML-Code zu schreiben, findet der Benutzer das User-Interface der Applikation 1:1 im Webbrowser wieder und kann die Bedienung in gewohnter Weise vornehmen.

▶▶ Grafische Benutzeroberfläche

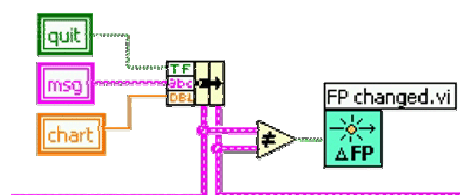
Die grafische Benutzeroberfläche ist die Grundlage, um Applikation schnell und einfach im Web verfügbar zu machen. Der aktuelle Zustand von JavaPanel ist für den Nutzer jederzeit ersichtlich.

▶▶ Auto-Modus - der schnellste Weg zur Webanwendung

Mit nur wenigen Mausklicks kann im Auto-Modus eine vorhandene Applikation im Web verfügbar gemacht werden. Mit einer einstellbaren Updaterate wird der Client zyklisch aktualisiert. Ein Eingriff in den Programmcode der Applikation entfällt vollständig.

▶▶ Event-Modus - sparsam im Verbrauch von Rechner- und Netzwerkressourcen

Im Event-Modus wird der Bedarf an Rechner- und Netzwerkressourcen auf ein Minimum reduziert. Der Client wird nur dann aktualisiert, wenn sich auch tatsächlich Änderungen auf dem Frontpanel der Anwendung ergeben haben. So wird sichergestellt, dass die vorhandenen Ressourcen primär der eigentlichen Applikation zu Verfügung stehen. Erreicht wird dies durch das Einfügen eines speziellen VI's in den Programmcode der Anwendung. Der Aufruf muss immer dann erfolgen, wenn sich Werte von Controls bzw. Indikatoren auf dem Frontpanel ändern. Dieser Aufruf löst das Ereignis für JavaPanel aus und der Client wird aktualisiert.

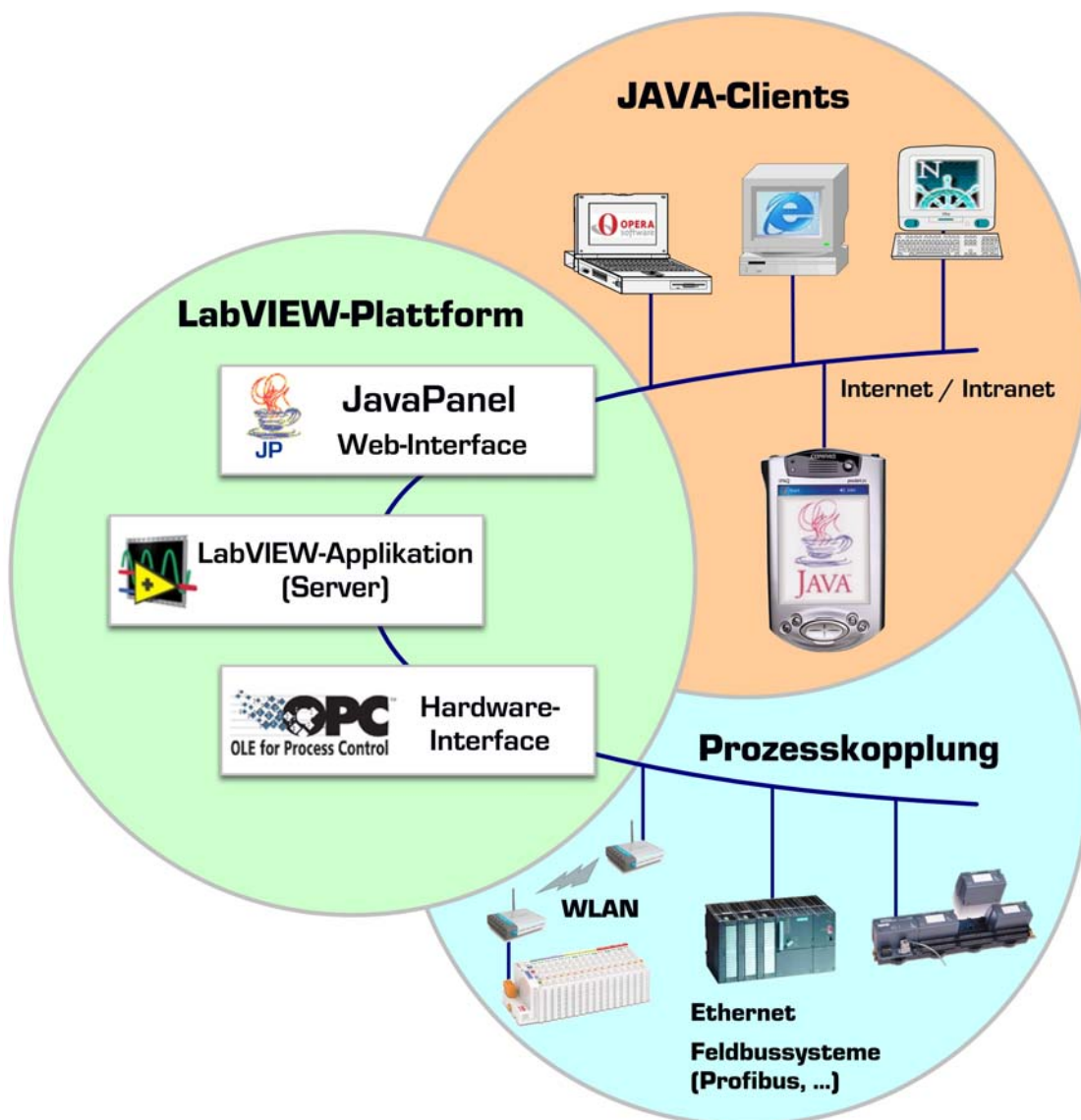


Geringfügige Erweiterung des Programmcodes im Event-Modus

Nutzung der Internet-/Intranetressourcen

► Benutzerspezifische Instanzen

Benutzerspezifische Instanzen heißt, dass jeder Client eine eigene Instanz einer Anwendung erhält. Es ist so möglich, dem Nutzer die Funktion einer Anwendung unabhängig von anderen Benutzern bereitzustellen. Damit wird erreicht, dass eine Anwendung mit benutzerspezifischen Parametern ausgeführt werden kann und jeder Nutzer das für sich relevante Ergebnis erhält. Die Anwendung selbst ist als Klasse (*.vit) auf dem Server abgelegt, von der JavaPanel zur Laufzeit Instanzen bildet.



JavaPanel im Praxiseinsatz

Aufgabe

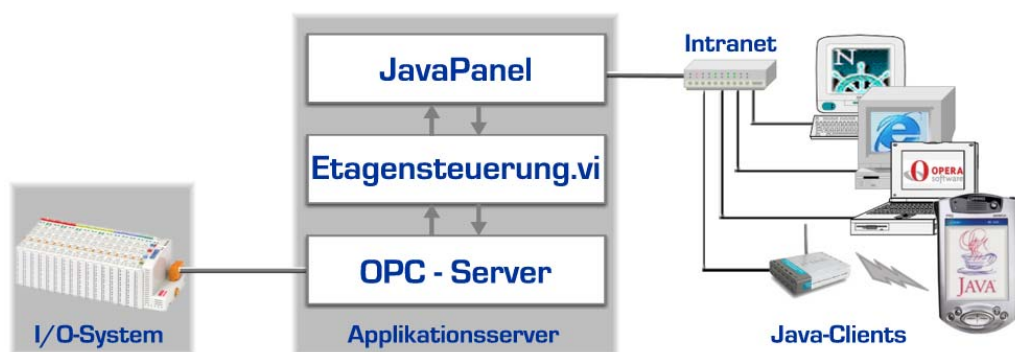
Das folgende Beispiel aus dem Bereich der Gebäudeautomation zeigt, wie mit Hilfe von JavaPanel die Steuerung einer Büroetage webfähig gestaltet wurde. Die konkrete Aufgabe bestand darin, es den Mitarbeitern mittels Webbrowser zu ermöglichen, jeden einzelnen Raum entsprechend seinen Möglichkeiten über das Intranet zu bedienen. Darüber hinaus soll die Bedienung losgelöst von einem bestimmten Arbeitsplatz mit dem PDA (personal digital assistant) durchführbar sein.

Umsetzung

Die Umsetzung gliedert sich in drei Teilschritte:

1. Erstellung des Anwendungs-VI's als übergeordnete Steuerung,
2. Kopplung des Anwendungs-VI's mit der Hardware,
3. Kopplung des Anwendungs-VI's mit dem Web-Interface,
4. Veröffentlichung des Anwendungs-VI's

Für die Beispielanwendung ergibt sich folgende Struktur:



Hardware

Den zentralen Teil der Hardware bildet ein I/O-System mit einem programmierbaren Feldbuscontroller. Die Steuerungselemente des Raumes (Taster, Lampen, Fensterkontakte, Temperatursensoren, ...) sind auf I/O-Klemmen geführt. Als LabVIEW-Plattform dient ein Panel-PC im Schaltschrank.

Software

Der Anteil der Software ist ursprünglich auf das Steuerungsprogramm im Feldbuscontroller begrenzt. Die Programmierung erfolgt nach IEC 61131-3. Das Controllerprogramm ermöglicht das Ein- und Ausschalten der Beleuchtung bei Betätigung der Taster und die Regelung der Raumtemperatur.

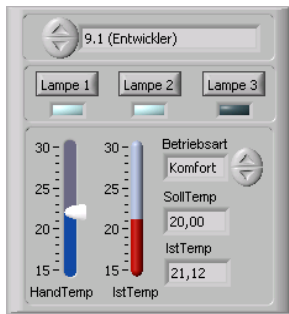
Anwendungs-VI

Das Anwendungs-VI bildet die Schnittstelle zwischen Mensch und Steuerung (HMI) und muss daher die Funktionen des Raumes abbilden.

Dazu gehören:

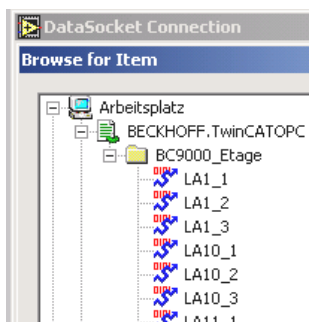
- ▶ Auswahl des Raumes,
- ▶ Schalten der Beleuchtung,
- ▶ Einstellung der Heizungsbetriebsart (Aus, Nacht, Standby, Komfort, Hand, Auto),
- ▶ Einstellung der Solltemperatur für den Handbetrieb,
- ▶ Anzeige der Ist-Temperatur im Raum.

Die Möglichkeit für die Bedienung mit dem PDA wird in Form eines kompakten Frontpanels berücksichtigt.



Kopplung mit der Hardware

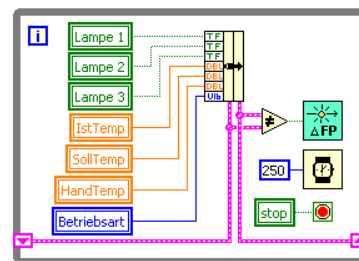
Bei der Kopplung mit der Hardware wird auf Standardlösungen zurückgegriffen. Über die Ethernetschnittstelle des Feldbuscontrollers ist eine unkomplizierte Kommunikation per TCP/IP mit dem Panel-PC möglich. Für den Datenaustausch sorgt ein OPC-Server. OPC ist die Spezifikation einer einheitlichen und herstellerunabhängigen Softwareschnittstelle (auf Windows-Plattformen). Diese eröffnet die Möglichkeit, auf Prozessdaten verschiedener Systeme von beliebigen Herstellern in einheitlicher Art und Weise zuzugreifen. Innerhalb der LabVIEW-Umgebung werden die einzelnen OPC-Items einfach per DataSocket Connection erreicht.



Kopplung mit dem Web-Interface

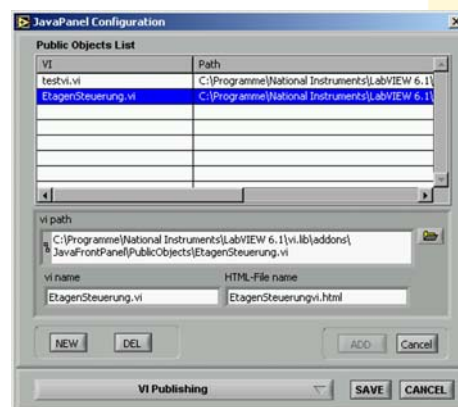
Grundsätzlich ist eine Kopplung mit dem Web-Interface ohne Eingriff in den Programmcode des Anwendungs-VI's möglich. Dazu kann im Auto-Modus des Web-Interface eine Updaterate eingestellt werden, welche die zyklische Aktualisierung des Frontpanels im Browser übernimmt.

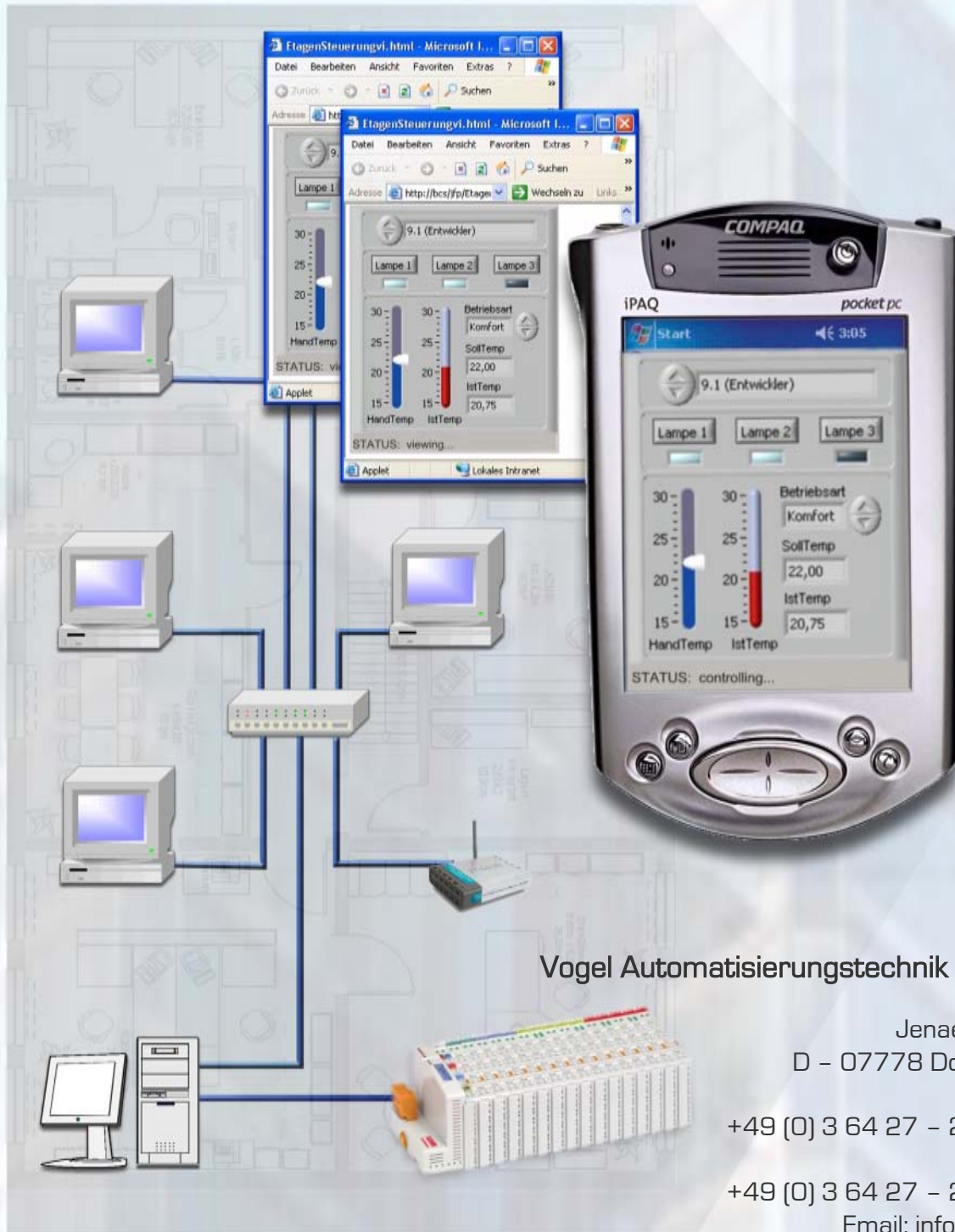
Um jedoch die Vorteile der Ereignissteuerung zu nutzen, muss das Anwendungs-VI die Fähigkeit besitzen, bei Änderungen des Wertes eines Controls bzw. Indicators Ereignisse auszulösen. Dazu ist eine geringfügige Erweiterung des Programmcodes nötig. Diese Erweiterung erfolgt durch das Einfügen eines VI's, welches bei Aufruf das Ereignis auslöst. Die einfachste Möglichkeit zur Detektierung von Änderungen auf dem Frontpanel ist die Überwachung der Prozessvariablen (OPC-Items) in einer Schleife. Ändert sich ein Wert, wird durch das ausgelöste Ereignis der Client aktualisiert.



Veröffentlichung

Wenn sichergestellt ist, dass alle Möglichkeiten der Änderung des Frontpanels durch Ereignisse erfasst sind, kann die Veröffentlichung des VI's erfolgen. Dazu wird mit Hilfe des Publishing-Tools das VI im Dateisystem ausgewählt und per Tastendruck die VI-spezifische HTML-Datei generiert. Das VI steht nun für den entfernten Zugriff zur Verfügung.





Vogel Automatisierungstechnik GmbH

Jenaer Str.7
D - 07778 Dornburg
Phone:
+49 (0) 3 64 27 - 200 30
Fax:
+49 (0) 3 64 27 - 200 31
Email: info@vat.de