



The logo for NIDays Engineer Next is centered on a blue gradient background. It features the text "NIDays" in white, enclosed within a white rectangular border. To the right of this, the words "ENGINEER" and "NEXT" are stacked vertically in a large, bold, white sans-serif font. A yellow graphic element, consisting of three parallel lines forming a stylized arrow or chevron shape, is positioned between the two words. The background is decorated with several diagonal stripes: a wide green stripe, an orange stripe, and a red stripe on the left side; and several blue stripes of varying shades on the right side.

NIDays **ENGINEER**
NEXT



CompactRIO, NI Linux Real-Time, and Embedded User Interface In the Field for Automation Tasks in Food Production

Gregor Matter

Geschäftsführer Sotronik GmbH



Ingenieurbüro

Für Software & Elektronik



wo die kreativsten Menschen
am liebsten nachdenken

Übersicht

- In der Industrie sind das deterministische Verhalten und allgemein die Echtzeit Fähigkeit von Applikationen (Steuerungen) oft Voraussetzung.
- Damit kommt man jedoch oft in Konflikt mit den Anforderungen an einer ergonomischen graphischen Bedienoberfläche. Diese erfordert ein Windows Betriebssystem und muss dadurch auf einem zweiten unabhängigen Rechner realisiert werden.
- Die neuen NI Linux Real-Time Systeme mit Embedded-Benutzeroberfläche und Mini DisplayPort bieten diesbezüglich eine äusserst komfortable Lösung dieses Dilemmas.

Zeitliche Entwicklung der Bedienoberfläche

- Wie veränderte sich die Thematik der Bedienoberfläche für eine Maschine oder Anlage



Vorgestern

1970



Gestern

1990



Heute

2010

Morgen

Vorgestern

- Die ersten elektrischen Maschinen in der Lebensmittelindustrie wurden mit Schaltern von Hand gesteuert.
- Automaten hat man mit Relais und später mit einer SPS realisiert.
- Die Bedienung ist mit Tasten und Lampen im Schaltschrank eingebaut.
- Die Kommunikation zwischen Mensch und Maschine ist auf ein Minimum reduziert.
- Mechanisch waren die Maschinen damals sehr robust.
- Nach über 30 Jahren laufen noch viele Tag für Tag.



Gestern

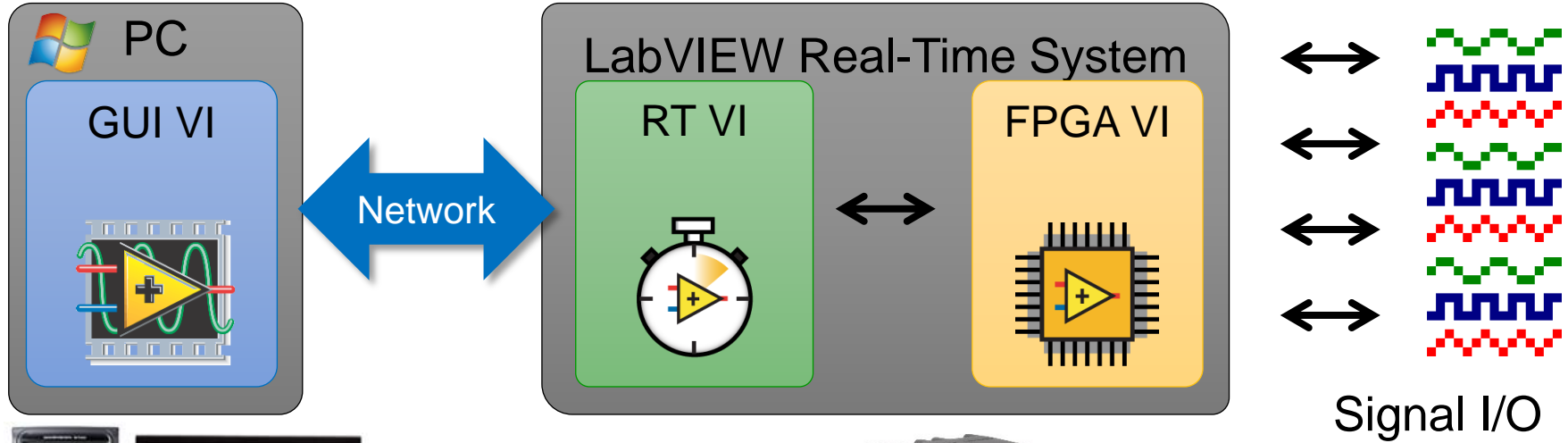
- In den 90er Jahren kamen dann diverse Displays in Kombination mit Taster oder auch als resistive Touch Displays auf.
- Damit wurde das GUI massiv verbessert und auch aussagekräftige Fehleranzeigen wurden realisiert.
- Die Lebensdauer dieser Displays wurde bald zum Problem. Nach 10...15 Jahren fallen diese aus oder sie werden blind.
- Der Wandel der Technik führte dazu dass es keine kompatiblen Ersatzgeräte gibt.



Heute

- Aktuell gibt es günstige PC, IPC oder Panel-PC mit Windows.
- Mit z.B. LabVIEW kann effizient ein ergonomisches GUI programmiert werden.
- Dieses lässt sich mit einem Panel-PC auch als Touch Panel in eine Schrank einbauen.
- Über die Jahre haben wir immer wieder Probleme mit der Stabilität.
- Windows ist kein Betriebssystem für 30 Jahre...
- Auch der Determinismus lässt zu Wünschen übrig...
- Die Stabilität und der Determinismus lässt sich mit einer „2-System“ Architektur lösen: Ein Windows System mit der Bedienoberfläche und ein Real-Time System mit dem Steuerprogramm.

Heute – 2-System Lösung

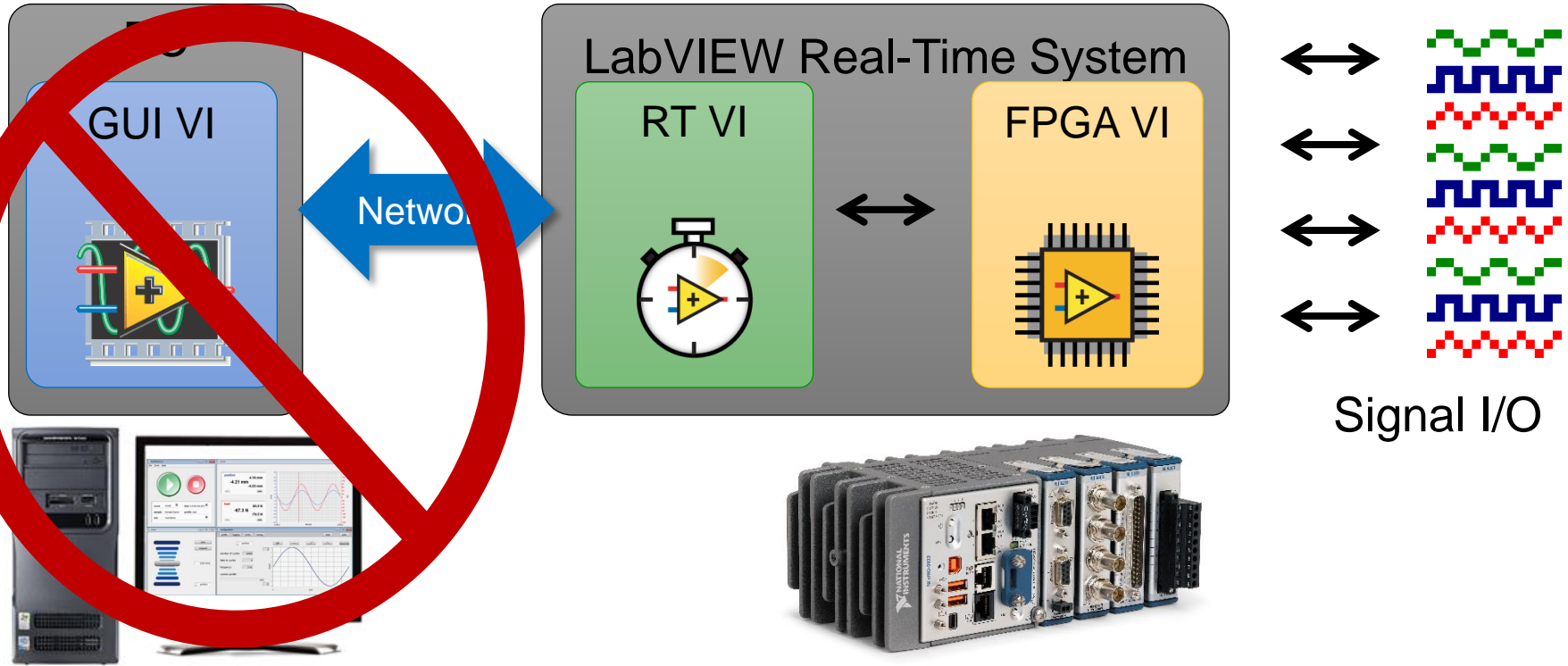


Heute – 2-System Lösung

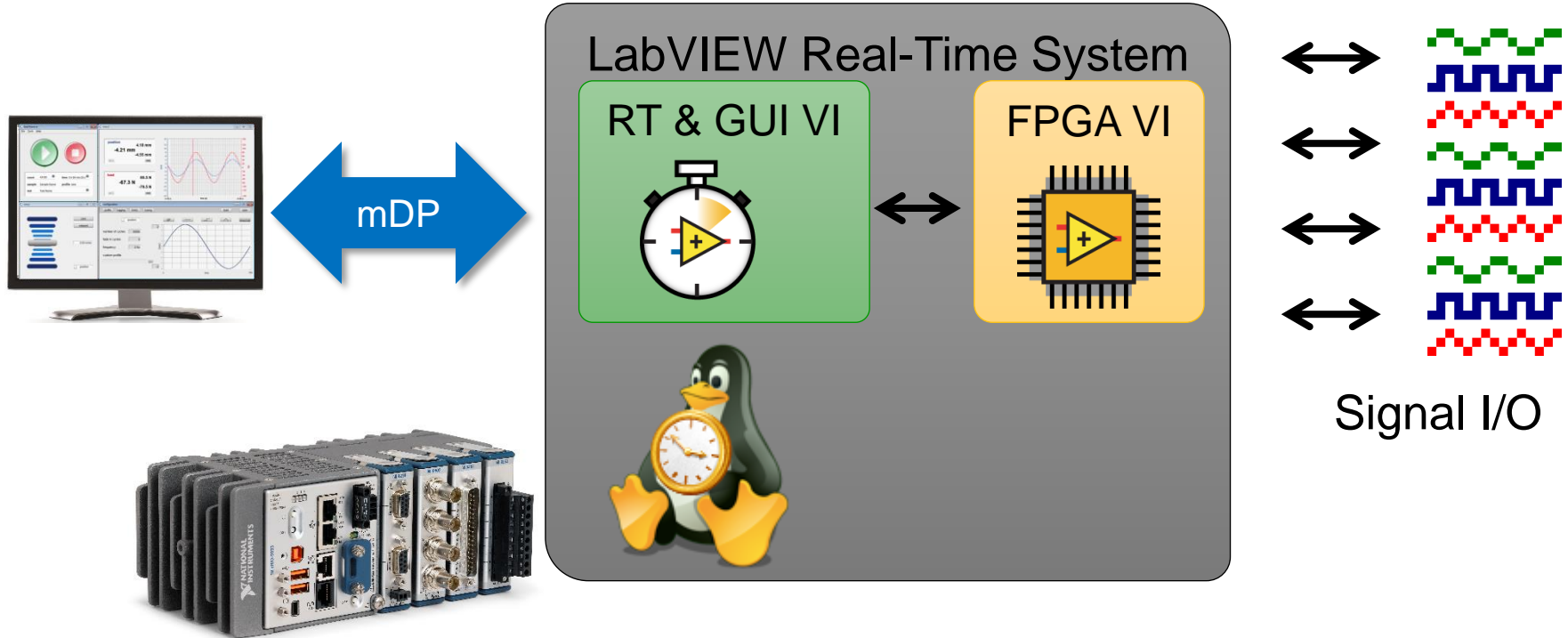
- Mit der LabVIEW Entwicklungsumgebung kann diese 2-System Lösung sehr komfortabel programmiert werden:
 - Windows, Real-Time und FPGA Code wird alles mit der gleichen Programmiersprache geschrieben.
 - Windows und cRIO Code (Real-Time und FPGA) wird in nur einem Projekt verwaltet.
 - Für die Kommunikation zwischen beiden Systemen bietet LabVIEW diverse einfache Möglichkeiten:
 - Network Published Shared Variables
 - Network Streams
 - TCP
- Nachteile der 2-System Lösung:
 - Hardware Kosten
 - Komplexe Software -> Programmierzeit



Morgen – Embedded User Interface



Morgen – Embedded User Interface





Die Bedienung (GUI) direkt am cRIO

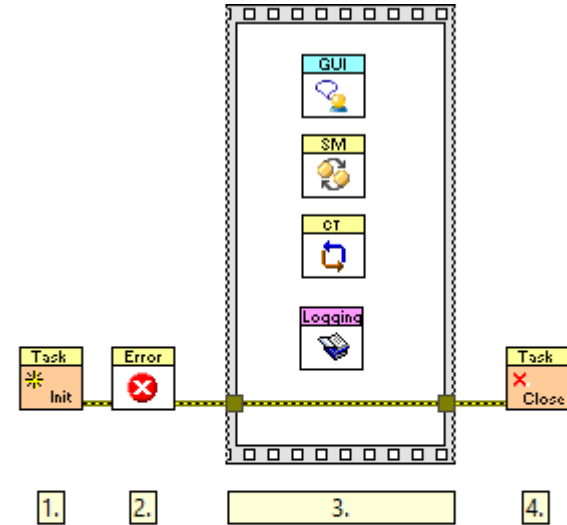
- Die neue cRIO Generation mit Linux Realtime hat einen Bildschirm Treiber und ein Displayport (HDMI).
- Es können normale Full HD Bildschirme angeschlossen werden.
- Über USB wird entweder Maus & Tastatur oder ein Touch Bildschirm angeschlossen.

Morgen

- Das cRIO ist für unsere Kunden in der Lebensmittelindustrie nicht die günstige – über die Jahre jedoch eine sehr robuste Alternative zu einer SPS.
- Mit den neuen Linux cRIO kann das GUI direkt im cRIO programmiert werden.
- Linux basiertes Echtzeitsystem deckt Anforderung an Determinismus und Sicherheit ab.
- Auf dem Markt gibt es sehr günstige grosse kapazitive Touch Bildschirme. Diese bauen wir direkt im Schrank ein.
- Von dieser Konstellation erwarten wir eine lange Lebensdauer. Sie sind sehr robust und Lebensmittel Industrietauglich (Reinigung...)
- Bildschirme mit HDMI Eingang wird es wohl noch sehr lange geben...

Tipps und Tricks

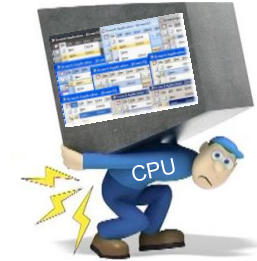
- SW Architektur:
 - Vergessen sie nicht dass es sich um ein Real-Time System handelt!
 - Verwenden sie verschiedene Tasks
 - Zeitkritische Loops dürfen nicht vom GUI zeitlich beeinflusst werden -> eigener getrennter GUI-Task
- Nicht unterstützte Features:
 - Subpanels
 - XControls
 - 3D Graphen
 - LabVIEW Database Connectivity Toolkit



1. Referenzen aller Kommunikationswege zwischen den Tasks (Queues, User Event) öffnen / falls vorhanden, den TCP_Sublayer starten
2. Reagiere auf einen Fehler beim Initialisieren
3. Alle Tasks VIs ausführen (GUI-Task, State-machine-Task, Zyklischer-Task, Logging-Task)
4. Alle Referenzen schliessen.

Tipps und Tricks

- SW Architektur:
 - Halten sie das GUI einfach, ein komplexes GUI belastet die CPU beträchtlich.
 - Beim Ein- und Ausstecken des Bildschirmes wird das Bild nicht automatisch neu aufgebaut. Der Bildschirm bleibt dann schwarz. Dies kann programmatisch durch das periodische Aufrufen eines System-Befehls gelöst werden.

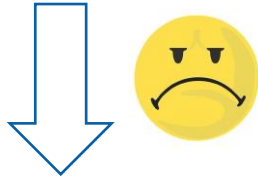


```
xrandr --auto --  
display :0 --  
output DP1 --  
mode 1280x1024
```

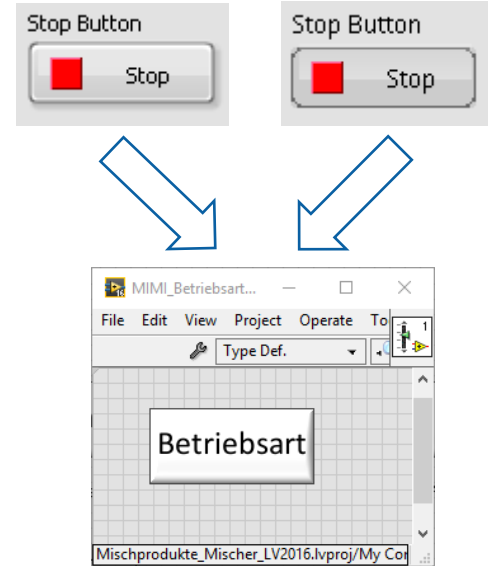


Tipps und Tricks

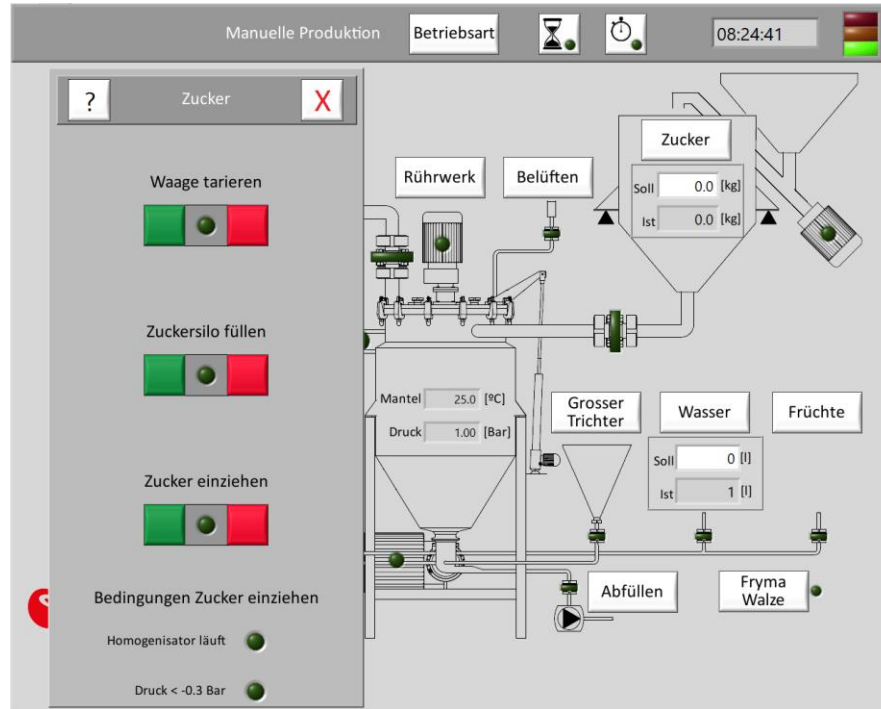
- Fonts und Controls sind nicht gerendert!
 - Definieren sie als erstes Fenster- und Schriftgrößen.
 - Verwenden sie anstelle von Labels eine Graphik mit dem darzustellenden Text
 - Erstellen sie Custom Controls mit Graphen um ein modernes Erscheinungsbild zu bewahren.



Erhöht den Aufwand für das Gestalten der Bedienoberfläche und schränkt die Flexibilität beim nachträglichen Ändern (z.B. Grösse des Fonts oder eines Controls) beträchtlich.



Embedded Bedienoberfläche



Zusammenfassung

Die neuen NI Linux Real-Time Systeme mit Embedded-Benutzeroberfläche und Mini DisplayPort kombinieren den bisher widersprüchlichen Bedarf an Echtzeit Fähigkeit und einer graphischen Bedienoberfläche.



Die Graphik hinkt jedoch noch den Möglichkeiten einer Windows Applikation nach.

Fragen?

Stay Connected During and After NIDays



ni.com/niweekcommunity



facebook.com/NationalInstruments



twitter.com/niglobal



youtube.com/nationalinstruments