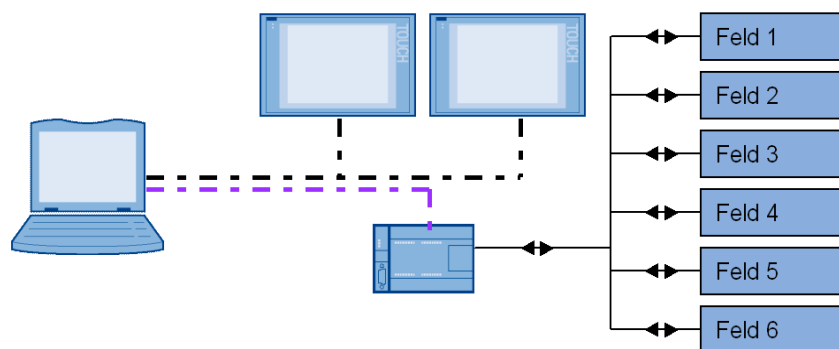


Projektdokumentation

Interaktive Informationswand für den Azubi Messestand

Programm zur Steuerung der Informationsschirme
und dezentralen Peripherie



Patrick Pulito

Sto AG Stühlingen
Abteilung Benutzerservice (FOB)
Weizen, den 31.07.07

in Kooperation mit



Kontakt : info@hygroSENS.com

Inhalt

1	Einleitung	3
1.1	Die Sto AG	3
1.2	Projektbeschreibung / Ziel des Projekts	3
1.3	Projektumfang	4
1.4	Prinzipschaltbild	4
2	Soll-Konzept.....	5
2.1	Anforderungen an Hard- und Software	5
3	Projektrealisierung	6
3.1	Hardware.....	6
3.2	Programmierung	7
3.2.1	<i>Sub Main</i>	7
3.2.2	<i>Sub xmlhandler()</i>	7
3.2.3	<i>Sub GetInputChanges</i>	8
3.2.4	<i>Sub Watchdog_tick</i>	9
3.2.5	<i>Sub runas()</i>	10
4	Ausblick / Folgeaktivitäten	11
5	Fazit	12
6	Danksagung	12
7	Kontakt	12
8	Begriffsdefinitionen.....	13

1 Einleitung

1.1 Die Sto AG

Die Sto AG mit Hauptsitz in Weizen (Südbaden) ist weltweit Marktführer auf dem Gebiet der Fassaden-Dämmsysteme, Fassaden- und Innenbeschichtungen sowie der Betoninstandhaltung.

Zur Sto Aktiengesellschaft gehören unter anderem folgende Tochterunternehmen :

- Sto Cretec GmbH : Betonschutz und –instandsetzung
- Sto Verotec GmbH : Vorhang- und Mischfassadensysteme
- Sto Inotec GmbH : Silo- und Maschinenteknik

Sto ist in Deutschland mit 12 Niederlassungen und ca. 80 Verkaufszentren vertreten. Weltweit sind ca. 6000 Mitarbeiter beschäftigt.

1.2 Projektbeschreibung / Ziel des Projekts

Ziel des Projekts ist die Ansteuerung der interaktiven Informationswand für Messen und Ausstellungen der Sto AG. Sie soll als Blickfang und zur abwechslungsreichen Informationsvermittlung dem Kunden und Interessenten gegenüber dienen.

Die Wand besteht aus 2 TFT-Flachbildschirmen, welche mit einer PowerPoint-Präsentation Informationen zu verschiedenen Teilgebieten rund um die Sto AG zeigen.

Am Rand sind die jeweiligen Wissensgebiete mit Schaltern abgebildet. Durch Betätigung eines Schalters springt PowerPoint automatisch zur jeweiligen Folie und der zugehörige Lichtschlauch leuchtet auf.

Das zu entwickelnde Softwareprodukt steuert die animierte PowerPoint-Präsentation in der Mitte der Wand und schaltet themenabhängig die Lichtschläuche ein bzw. aus. Als dezentrales Peripheriegerät dient ein USB I/O Board der Firma Hygrosens.

Die Programmeinstellungen sind in einer XML-Datei hinterlegt. Unter anderem wird hier gespeichert, welche Folien zu welchem Themengebiet gehören etc.

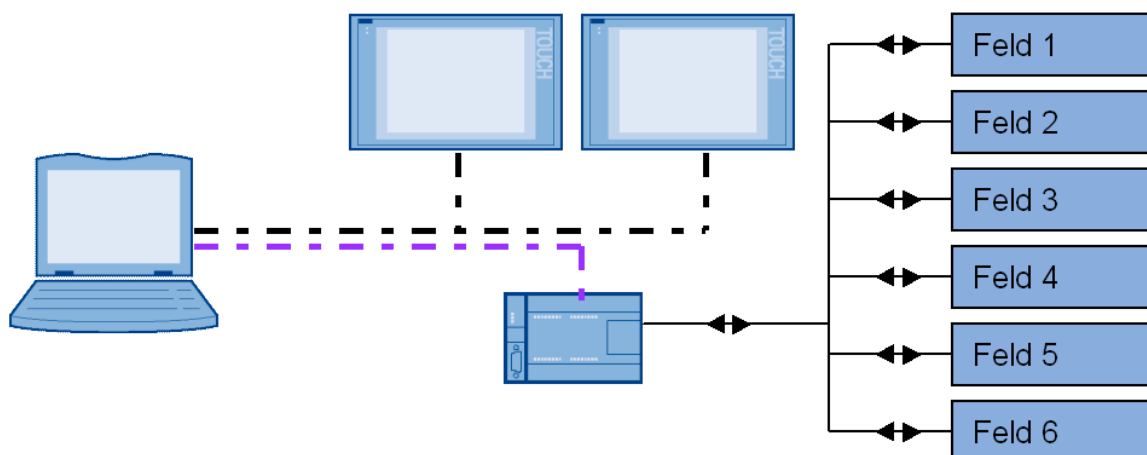
1.3 Projektumfang

Das Projekt umfasst die gesamte programmiertechnische Umsetzung, die Prüfung der Praxistauglichkeit sowie die Inbetriebnahme im Rahmen einer Testumgebung.

Im Einzelnen müssen folgende Schritte realisiert werden :

- Ermitteln der geeigneten Programmiersprache
- Entwicklung der einzelnen Klassen und Module
 - Ansteuerung der Ein- und Ausgänge
 - Steuerung der PowerPoint-Präsentation
 - Einlesen und Auswerten der XML-settings Datei
 - Automatisches, zeitgesteuertes Starten und Herunterfahren des Computers
 - Logging auf der Console und in einer Textdatei
- Funktionstests
- Inbetriebnahme

1.4 Prinzipschaltbild



2 Soll-Konzept

2.1 Anforderungen an Hard- und Software

- Das Programm soll die Ausgänge und die PowerPoint-Präsentation abhängig vom vorgewählten Themengebiet steuern
- Über eine separate XML-Datei sollen Sollwerte eingelesen werden, um Zeiten, Folien, Passwörter etc. variabel zu gestalten
- Bei einer nicht erlaubten Schalterkombination soll eine Fehlermeldung in der Präsentation ausgegeben werden
- Der Computer soll automatisch zeitgesteuert starten und herunterfahren
- Das Programm soll flexibel an neue Anforderungen angepasst werden können

3 Projektrealisierung

Im folgenden wird die Realisierung des Projektes in chronologischer Reihenfolge beschrieben.

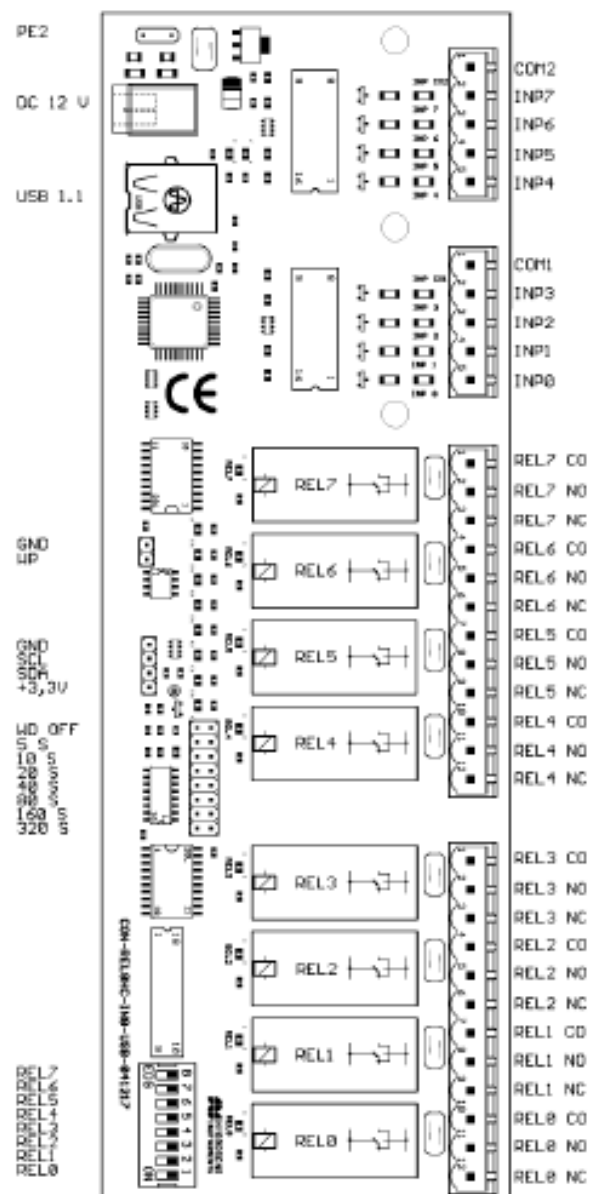
3.1 Hardware

Für die Schaltung wird eine Relais- und Eingangskarte mit USB-Interface verwendet, welche 2 Gruppen mit je 4 Relais (Wechselkontakt) und 8 potentialgetrennte Eingänge besitzt.

An den Eingängen sind die den Themengebieten entsprechenden Vorwahlschalter angeschlossen, die Ausgänge schalten die zugehörigen Lichtschläuche.

Das Softwareprogramm kommuniziert per USB 2.0 mit der Peripheriekarte.

Ansicht und Belegung



3.2 Programmierung

Die Ansteuerung der Peripherie per USB erfolgt durch eine selbst entwickelte DLL-Datei.

Genauere Informationen dazu liegen der CD im PDF-Format bei.

Die Software wird als Konsolenanwendung mit Visual Basic 2005 entwickelt. Die Installation des .NET Frameworks 2.0 wird dabei vorausgesetzt.

3.2.1 Sub Main

Die Verbindung zur Peripherie wird in der *Main()*-Methode hergestellt.

Über den Befehl *OpenDevice(0)* wird die Verbindung zum USBPort hergestellt. Die Funktion quittiert mit *True*, im Fehlerfall mit *False*.

Zu Beginn werden alle Ausgänge auf 0 gesetzt und der Standard-Foliensatz mit allgemeinen Infos rund um die Sto AG wird geladen.

Dies geschieht durch *Process.Start(prs.pps)*.

3.2.2 Sub xmlhandler()

Die Subroutine *xmlhandler()* liest die Einstellungen aus der *settings.xml* Datei.

In der Datei ist jeweils definiert, welche Foliennummern der PowerPoint-Präsentation zu welchem Themengebiet gehören. Ein *XML-Node* ist wie folgt strukturiert :

```
<presentation1>  
  <description>präsi1</description>  
  <input>1</input>  
  <output>1</output>  
  <start>5</start>  
  <end>7</end>  
</presentation1>
```

Das Element `<description>` kennzeichnet das Präsentationsthema. Um bei Schalterbetätigung zu erkennen, welches Präsentationsthema geöffnet werden muss, sind die Felder `<input>` und `<output>` im Node definiert.

Das `<start>` Element definiert die Startfolie, bei welcher das Thema beginnt. Die Endfolie bestimmt bis zu welcher Folie die Präsentation läuft.

Darüber hinaus ist in der `settings.xml` hinterlegt, welche Folien zum allgemeinen Informationsteil gehören, wie lange nach Schalterbetätigung keine weiteren Benutzeraktionen mehr zugelassen sind, und zu welchen Uhrzeiten das System starten bzw. herunterfahren soll.

3.2.3 Sub `GetInputChanges`

In der Ereignisbehandlungsroutine `GetInputChanges` wird auf eine Veränderung des Eingangsregisters reagiert. Als Parameter wird das neue Bitmuster übertragen.

Grundsätzlich werden nur $0 \rightarrow x$ Flanken ausgewertet, $x \rightarrow 0$ Flanken werden ignoriert (wenn der Taster losgelassen wird).

Abhängig vom neuen Eingangsregister wird aus der `settings.xml` gelesen, welches Thema angezeigt werden soll und welcher Lichtschlauch eingeschaltet werden soll.

```
'search topic by new inreg
Console.WriteLine("search topic by new inreg")
For Each topic As presentation In readarray

    If InReg = topic.Input Then
        current_topic = topic
        Exit For
    End If

Next

'set OutReg
OutReg = current_topic.Output
IOctrl1.OutReg = OutReg
```

Das Umschalten funktioniert somit ereignisorientiert. Es bedarf also keiner speicher- und rechenzeitintensiven Endlosschleife, die ständig die Eingänge überwacht.

Nicht zulässige Eingangsregister, wenn z.B. mehrere Taster gleichzeitig gedrückt werden, werden ignoriert. In diesem Fall wird für 3sec ein `GlobalBreak`-Flag gesetzt, welches weitere Eingaben verhindert.

3.2.4 Sub Watchdog_tick

In der Subroutine *Watchdog_tick* wird der Ablauf der Präsentationsfolien gesteuert.

Die globale Variable *currentSlide* speichert die momentan angezeigte Foliennummer. Die Routine schaltet im 10sec Takt eine Folie weiter, bis die Schlussfolie des momentan laufenden Themas erreicht ist.

Am Ende eines Themas, oder nachdem mindestens 10min lange kein Taster gedrückt wurde, wird automatisch der Foliensatz mit allgemeinen Infos gestartet. Gleichzeitig werden alle Lichtschläuche ausgeschaltet.

```
'repeating main presentation or count to the end of the current presentation
If currentSlide = mainTopic._End Then

    currentSlide = mainTopic.Start

ElseIf currentSlide = current_topic._End Then

    currentSlide = mainTopic.Start
    IOctrl.OutReg = 0

Else
    currentSlide += 1
End If

SendKeys.SendWait(currentSlide.ToString)
SendKeys.SendWait("{ENTER}")

Catch ex As Exception
    Console.WriteLine(ex.Message)
    Console.ReadLine()
```

3.2.5 Sub runas()

Die *Function runas()* ermöglicht ein automatisiertes, zeitgesteuertes Herunterfahren des Systems. Die Funktion ruft eine *.batch* Datei auf, die lokal auf dem Rechner gespeichert ist. Diese Datei initiiert einen kontrollierten Shutdown mit entsprechendem Eintrag im Ereignisprotokoll des Betriebssystems.

```
Public Function runas() As Boolean

    Try
        'set password to run the shutdown batch file with administrator
        Dim pwd As New Security.SecureString
        Dim pwdstring As String = "password"

        For Each chr As Char In pwdstring.ToCharArray
            pwd.AppendChar(chr)
        Next

        Process.Start("C:\shut_1800.bat", "██████████", pwd, System.Environment.MachineName)

        Return True

    Catch ex As Exception

        Console.WriteLine("error: " & ex.Message)

        Return False

    End Try

End Function
```

Der automatische Systemstart morgens wird durch das BIOS gestartet. Hier sind Zeiten zum Start hinterlegt, außerdem ist das Starten durch WakeOnLAN möglich.

4 Ausblick / Folgeaktivitäten

Einstellungen wurden bewusst in eine externe Datei ausgelagert, um den Quellcode so allgemein wie möglich zu halten. Erweiterungen lassen sich somit problemlos integrieren. Der Quellcode ist lückenlos dokumentiert.

Folgende Erweiterungen sind geplant für Q4/2007 :

- eMail Benachrichtigung im Fehlerfall
- Statistikmodul
- Anzeige im WebBrowser
- Integration einer LiveCam
- Content Management im Browser als Webanwendung

5 Fazit

Das Projekt wurde erfolgreich abgeschlossen, alle Anforderungen des Pflichtenheftes wurden erfüllt.

Erweiterungen des Programms sind problemlos möglich, weil der Quellcode bewusst allgemein gehalten wurde, und die Parameter der Applikation durch die *settings.xml* Datei an jede andere Lösung adaptiert werden kann.

Inbetriebnahme der produktiven Informationswand ist in KW35/2007 geplant.

6 Danksagung

Besonderer Dank gilt Hr. *Jürgen Bauer*, Hr. *Matthias Gut*, Hr. *Felix Böhler* von der Abteilung FOB sowie den Kollegen der Abteilung TVA mit Hr. *Beil* am Hauptsitz in Weizen.

Die Herren haben uns mit Rat und Tat unterstützt, und standen mit Know-how und Anregungen zur Seite.

Nicht zuletzt sei an dieser Stelle Hr. *Friedrich* von der Fa. *Hygrosens* GmbH dankend erwähnt. Die Firma hat uns ein USB I/O-board zur Verfügung gestellt.

7 Kontakt

Bei Fragen und Anregungen wenden Sie sich bitte an Hr. Friedrich | Fa. Hygrosens :
info@hygrosens.com

8 Begriffsdefinitionen

Begriff	Erklärung
XML	Die Extensible Markup Language (engl. für „erweiterbare Auszeichnungssprache“), abgekürzt XML, ist eine Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form von Textdateien. XML wird bevorzugt für den Austausch von Daten zwischen unterschiedlichen IT-Systemen eingesetzt, speziell über das Internet.
Visual Basic	Visual Basic (Abk. VB) ist eine objekt-orientierte Programmiersprache, die auf dem Microsoft .NET Framework basiert.
USB	Der Universal Serial Bus (USB) ist ein serielles Bussystem zur Verbindung eines Computers mit externen Geräten.
Optokoppler	Der Optokoppler ist ein opto-elektronisches Verbund-Bauelement, welches aus einem meist Infrarotstrahlung emittierenden und einem die Strahlung empfangenden Bauelement besteht.
LED	Eine Leuchtdiode (auch Lumineszenz-Diode, kurz LED für Light Emitting Diode bzw. lichtemittierende Diode) ist ein elektronisches Halbleiter-Bauelement.
DLL	Dynamic Link Library (DLL) bezeichnet allgemein eine Dynamische Bibliothek, meist bezieht sich der Begriff jedoch auf die unter dem Betriebssystem Microsoft Windows verwendete Variante.