

Comvisu V0.95

Benutzerdokumentation

Inhalt

Inhalt	1
Von der Schnittstelle zur Anzeige	1
Übersicht.....	2
Anzeigen und Instrumente	5
Bitmap-Instrumente.....	6
Schnittstellen.....	6
Protokoll	7
Ein- und Ausgangskonfiguration	8
Numerische Eingänge.....	8
Operatoren.....	9
Funktionen	10
Kommandozeilenparameter	12
Anderes.....	13
Lizenz.....	13
Haftungsausschluss.....	13

Von der Schnittstelle zur Anzeige

Die Comvisu ist auf einfachste Bedienung ausgelegt, die Konfiguration erfolgt in sekundenschnelle:

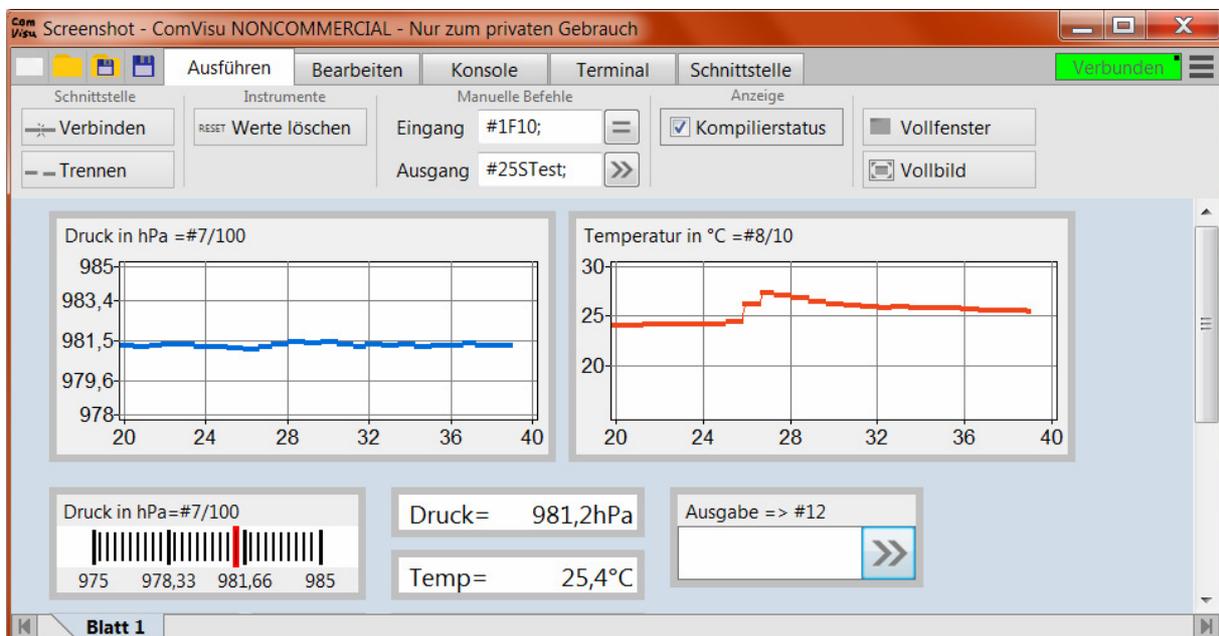
- Das gewünschte Instrument in der Bearbeiten-Oberfläche platzieren.
- Den Eingang konfigurieren, es kann ein einzelner Eingangskanal oder eine Berechnungsformel wie in der Tabellenkalkulation, sogar mit mehreren Kanälen, angegeben werden. Es stehen 1000 vordefinierte Kanäle (#1 bis #999) zur Verfügung.
- Nach dem *Verbinden* können die Instrumente über die Schnittstellen auf einfache Weise beschickt werden.
- Der Befehl "#5F12,3;" beispielsweise weist dem Kanal #5 den Wert 12,3 zu. Die Aktualisierung aller Anzeigen und Instrumente, die von diesem Kanal abhängen, wird von der Comvisu automatisch ausgelöst.
- Die Verarbeitung des Wertes hängt dann von der Anzeige bzw. der Eingangsformel ab, welcher diesen Kanal verwendet. Eine Digitalanzeige wird den Wert 12,3 als solchen anzeigen, eine LED-Anzeige wird den Wert 12,3 als true (logisch eins) interpretieren und das „LED-Element einschalten“.
- Ausgänge werden in gleicher Weise definiert.
- Die Daten können über die serielle Schnittstelle, über UDP-Verbindungen oder per URL-Anfrage übermittelt werden. Auch die Kombination mehrerer Schnittstellen ist möglich.

Alle Befehle setzen sich zusammen aus dem Kanal, gekennzeichnet durch die vorangestellte Raute #, aus dem Formatkennzeichner (*F* für numerische Werte; *S* für Zeichenketten), dem Wert und einem ; als Endezeichen. In der Bearbeiten-Oberfläche können Instrumente in beliebiger Anzahl platziert werden. Jedem Instrument wird im Feld "Eingang" ein Kanal zugewiesen, es kann auch eine Formel mit mehreren Kanälen angegeben werden.

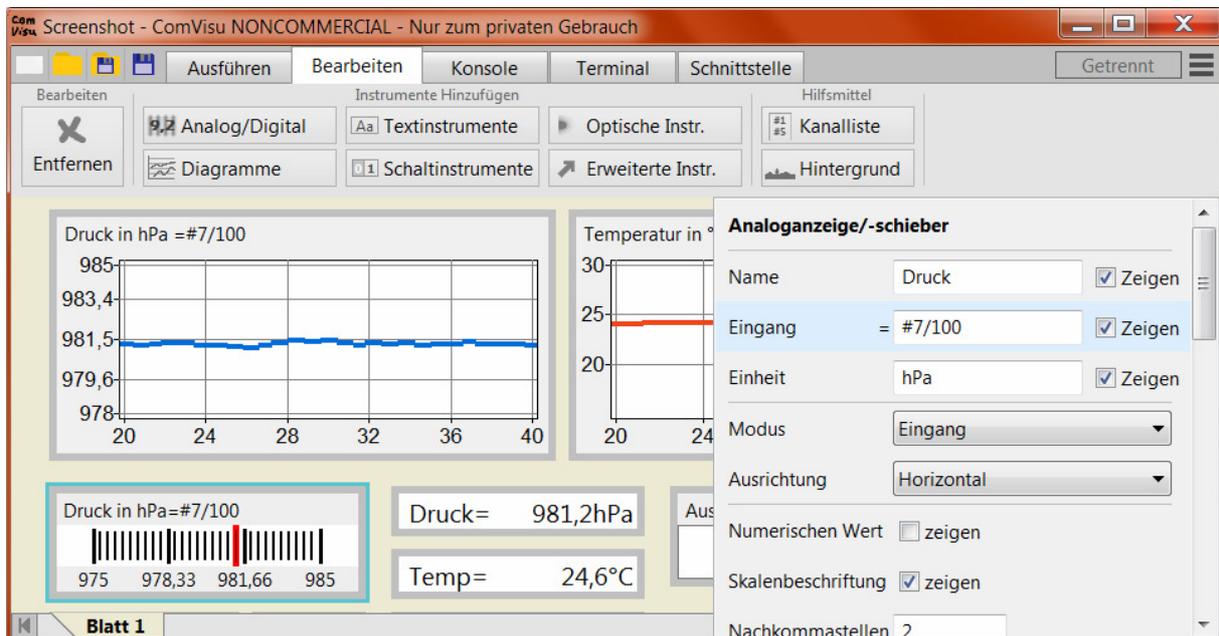
Es können mehrere Blätter mit Instrumenten eingerichtet werden, welche bei verbundener Schnittstelle alle in Echtzeit beschickt werden.

Übersicht

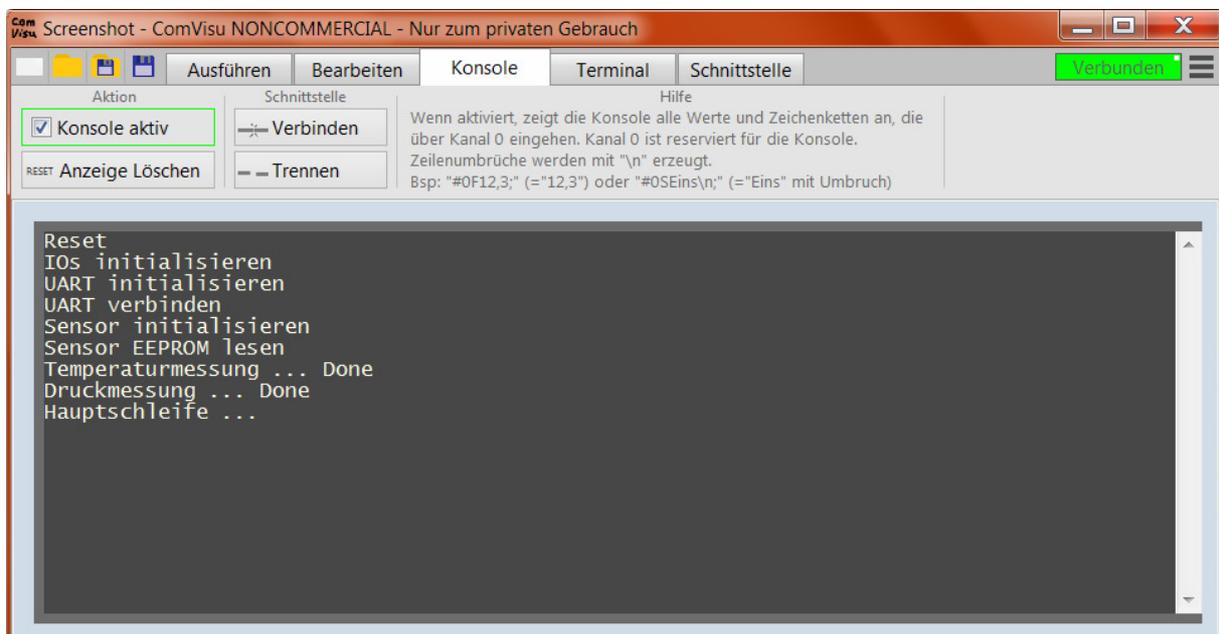
Die Comvisu während dem Betrieb mit aktiver Schnittstelle. Beispielhaft ist eine Visualisierung eines Druck- und Temperatursensors gezeigt.



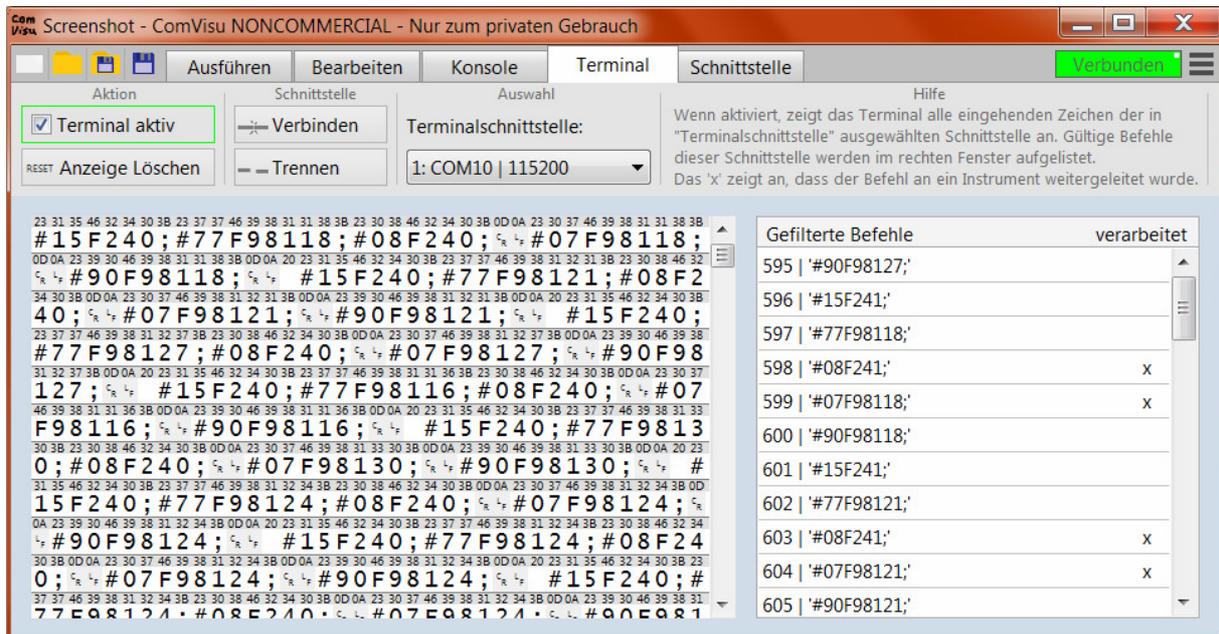
In der Bearbeitungsfläche werden Instrumente platziert und konfiguriert. Das blau hinterlegte Feld Eingang (bzw. Ausgang) im Eigenschafteneditor legt den Kanal fest, das Aussehen der Anzeigen kann weitgehend dem eigenen Bedarf angepasst werden.



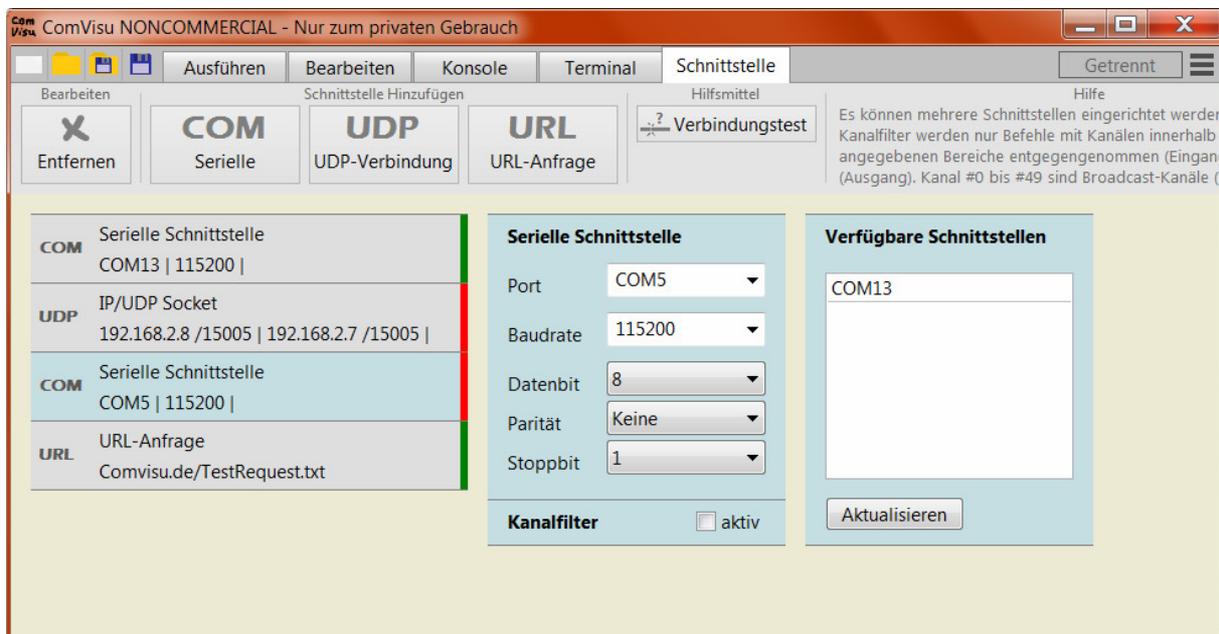
Auf der Konsole werden alle Werte und Zeichenketten angezeigt, die auf Kanal #0 eingehen. Für ein komfortables Debugging wie in einer Konsolenanwendung. "#0SEins\n;" führt zur Anzeige "Eins" mit anschließendem Zeilenumbruch.



Zur Fehlersuche können im Terminal die eingehenden Zeichen angezeigt werden, es werden die daraus gefilterten Befehle aufgelistet und markiert, ob sie verarbeitet wurden, d.h. ob ein Instrument vom entsprechenden Kanal abhängt.



Im Schnittstellenfenster werden die projektspezifischen Schnittstellen eingestellt. Die Daten aller beteiligten Schnittstellen werden verarbeitet, auch seriell und Netzwerk gemischt.



Anzeigen und Instrumente

Details zu den Instrumenten finden sich im jeweiligen Eigenschafteneditor.
Es sind folgende Instrumente verfügbar:

Name	Beschreibung	Ein-/Ausgang
<i>Numerische Anzeige</i>	Zur Darstellung eines Wertes als Zahl. Optional mit Angabe einer Einheit.	Eingang (F)
<i>Numerische Ausgabe</i>	Zur Ausgabe eines Wertes, welcher über die Tastatur in ein Textfeld eingegeben wird.	Ausgang (F)
<i>Analoganzeige/ -Schieber</i>	Zur analogen Darstellung eines Wertes als Zeigerinstrument. Optional mit digitaler Wertanzeige. Ebenso als Einstellschieber konfigurierbar zur Ausgabe eines Wertes	Eingang (F); Ausgang (F)
<i>Balkenanzeige</i>	Zur analogen Darstellung eines Wertes als Balken. Ebenso als Einstellschieber konfigurierbar zur Ausgabe eines Wertes.	Eingang (F); Ausgang (F)
<i>Zeitdiagramm</i>	Zur Darstellung von Werten als Kurve über der Zeit	Eingänge (F)
<i>XY-Diagramm</i>	Zur Darstellung von beliebigen Punkten/Kurven	Eingänge (F)
<i>Fernanzeige</i>	Textfeld, welches über den Eingang gesetzt werden kann	Zeicheneingang (S)
<i>Zeichenausgabe</i>	Zur Ausgabe von Text, welcher über die Tastatur in einem Textfeld eingegeben wird	Zeichenausgang (S)
<i>Taster</i>	Tasterelement sendet den Wert 1 bei Drücken des Tasters	Ausgang (F oder S)
<i>Schaltleiste</i>	Schalterleiste, mit einem oder mehreren Schaltelementen, viele Konfiguriermöglichkeiten	Ausgang (F)
<i>LED-Anzeige</i>	Anzeige einer LED mit bis zu 10 Farben, welche entsprechend dem Eingangswert angezeigt werden	Eingang (F)
<i>Blink-LED</i>	LED die bei jedem positivem Eingangswert für eine vorgegebene Zeit aufblitzt. Zur Visualisierung von Ereignissen.	Eingang (F)
<i>LED-Reihe</i>	LED-Reihe zur binären Darstellung von Ganzzahlen mit bis zu 32 bit	Eingang (F)
<i>Bild-Anzeige</i>	Anzeige bis zu 10 Bilder, welche entsprechend des Eingangs angezeigt werden.	Eingang (F)
<i>Tongeber</i>	Erzeugt Dauerton oder Impulse ausgelöst durch positive Eingangswerte mit vordefinierbarer Frequenz	Eingang (F)
<i>E/A-Kanal</i>	Weiterleitung eines Eingangs auf einen Ausgang. Einsetzbar als einfache Weiterleitung an andere Schnittstellen oder zur ausgelagerten Wertberechnung bei zurücksenden an die gleiche Schnittstelle	Universaleingang (F/S); Ausgang (F/S)
<i>Dateirekorder</i>	Aufzeichnung von Werten in eine Datei	Universaleingänge (F/S)
<i>Getaktete Ausgabe</i>	Gibt zyklisch Werte aus	Ausgang (F) und Eingang (F)
<i>Lookup-Tabelle</i>	Zur tabellengestützten Umrechnung von Werten	Stellt Funktion bereit

<i>Bitmap-Taster</i>	Funktion wie normaler Taster mit zuweisbaren Bitmaps	Ausgang (F oder S)
<i>Bitmap-Schalter</i>	Schalter mit zwei Schaltstellungen. Zuweisbare Bitmaps.	Ausgang (F)
<i>Bitmap-Slider</i>	Slider-Instrument mit zuweisbaren Bitmaps	Ausgang (F)

Bitmap-Instrumente

Mit den Bitmap-Instrumenten lässt sich durch Zuweisen von Bilddateien (bmp,png oder jpg) eine beliebige Darstellung der Instrumente erreichen. Die Größe ergibt sich jeweils aus dem ersten der zuweisbaren Bitmaps und können im Instrument nicht getrennt eingestellt werden. Für den Fall, dass kein Bitmap zugewiesen ist, wird hilfsweise eine interne Darstellung gewählt.

Prinzipiell werden transparente Bereiche in PNG-Dateien berücksichtigt und das Zusammenspiel der verschiedenen Bitmaps ist darauf optimiert. So lässt sich beispielsweise im Instrument Bitmap-Schieberegler ein Schieber in beliebiger Form realisieren.

Für jedes Bitmap gibt es außerdem einen „markierten“ Zustand, welcher angezeigt wird, wenn sich die Maus über dem Instrument befindet, zur Anzeige, dass es „klickbar“ ist. Das Bitmap für den markierten Zustand wird immer dem des Grundzustandes überlagert, sodass man wieder von der Transparenz Gebrauch machen kann und bspw. als Markierung nur einen zusätzlichen Rahmen zeichnet. Die Größe des markierten Darstellung sollte immer gleich wie die des zugehörigen Bitmap im Grundzustand sein. Die Bitmaps für den markierten Zustand sind optional, sie werden

Schnittstellen

Die Schnittstellen werden im Menü-Blatt “Schnittstelle” eingestellt.

Als Schnittstellenarten sind möglich:

Serielle Schnittstelle, UDP-Verbindung und URL-Anfrage.

Es ist eine beliebige Anzahl an Schnittstellen möglich, auch verschiedene Schnittstellenarten gemischt. Alle eingerichteten Schnittstellen müssen auch verfügbar sein, ansonsten wird der Verbindungsvorgang abgebrochen.

Ausgehende Befehle werden grundsätzlich an alle Schnittstellen verteilt, nicht jedoch bei wirksamer Kanalfilterung.

Für jede Schnittstelle kann ein Kanalfilter eingestellt werden. Der Filter wirkt sowohl für eingehende als auch ausgehende Befehle. Bei aktiviertem Kanalfilter werden über diese Schnittstelle nur Befehle angenommen bzw. ausgegeben, wenn der codierte Kanal innerhalb des angegebenen Bereiches liegt. Die Kanäle #0 bis #49 sind als Broadcast-Kanäle definiert, d.h. sie lassen sich nicht filtern.

Bei der **Serielle Schnittstelle** ist zu beachten, dass die Comvisu bereits eine erfolgreiche Verbindung anzeigt, wenn lediglich der Port (z.B. COM2) bei

angeschlossenem Gerät richtig eingestellt ist. Fehlerhafte Einstellungen in der Baudrate usw. verhindern eine erfolgreiche Kommunikation, lassen sich aber von Comvisu-Seite nicht feststellen.

Bei der **UDP-Verbindung** werden die IP-Adresse und der Port sowohl für den eigenen Rechner als auch für die Gegenstelle angegeben. Da UDP-Verbindungen per Definition *verbindungslos* sind, ist für die Comvisu nicht festzustellen, ob die Gegenstelle tatsächlich existiert und auf den angegebenen Port reagiert. Entsprechend wird eine erfolgreiche Verbindung bereits dann angezeigt, wenn lediglich die lokale IP-Adresse korrekt ist.

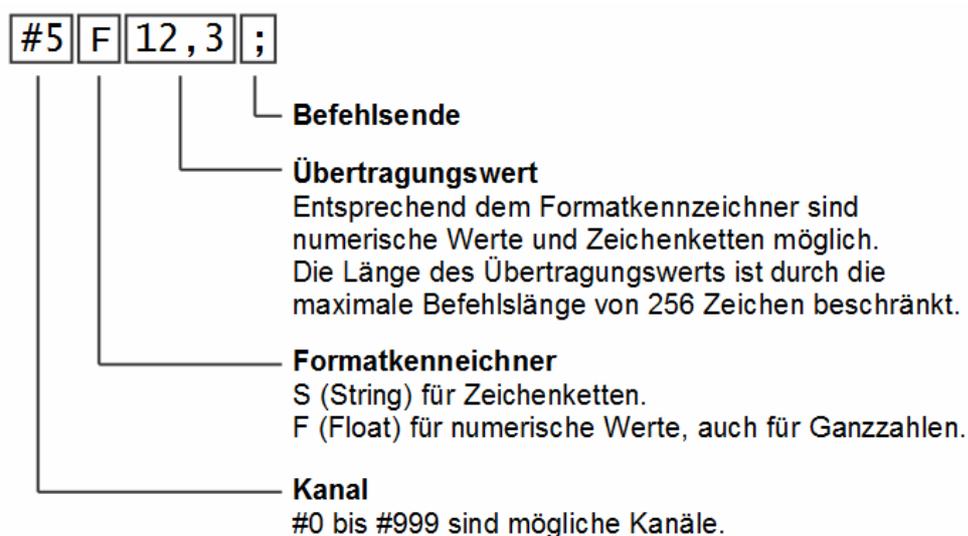
Über die **URL-Anfrage** lassen sich Daten von einem Server, adressiert über die URL abfragen. Die Anfrage erfolgt per „POST“-Anfragemethode. Ein ausgehender Befehl löst die Anfrage aus, der Befehl wird im Dateibereich (Body) der Anfrage übertragen. Die Comvisu wartet dann auf eine Antwort und verarbeitet einen in der Antwort übermittelten Befehl als Eingang. Der Antwortbefehl muss ebenfalls im Dateibereich stehen. im Gegensatz zu den anderen Schnittstellen kann die Gegenstelle bei der URL-Anfrage nur auf einen Befehl reagieren und nicht von sich aus eine Datenübermittlung veranlassen.

Der Status der Anfrage wird nicht ausgewertet, eine fehlgeschlagene Anfrage wird daher von der Comvisu nicht unbedingt erkannt. Da der Dateibereich als Eingang verarbeitet wird, kann dieser auch im Terminal betrachtet werden und so bei der Fehlersuche einer fehlerhaften Verbindung helfen.

Protokoll

Für die Kommunikation zwischen der Comvisu und dem externen Teilnehmer an der seriellen Schnittstelle werden ausschließlich ASCII-Zeichen verwendet. Sowohl für die Ein- als auch für die Ausgabe haben alle Befehle den gleichen Aufbau.

Beispiel: #5F12,3;



Abgesehen von Kanal #0, welcher für die Konsole reserviert ist, sind alle Kanäle gleichwertig und können sowohl Zahlenwerte als auch Zeichenketten annehmen. Wird

einem Kanal mit obigem Befehl ein Wert zugewiesen, dann wird dieser im Kanal gespeichert und alle Anzeigen, welche von diesem Kanal abhängen, werden aktualisiert. Bei Programmstart werden alle Kanäle mit dem Zahlenwert 0 (null) vorbelegt. Dies ist wichtig, wenn in einer Anzeige mehr als ein Kanal verwendet wird, da Befehle immer nur hintereinander übertragen werden können und die Anzeige bei der ersten Berechnung bereits auf alle in der Eingangsformel enthaltenen Kanäle zugreift.

Passive Befehle sind Befehle mit vorangestelltem Schrägstrich (/), z.B. /#5F12,3; Der übermittelte Wert bei passiven Befehlen wird in den Kanalspeicher geschrieben, aber abhängige Anzeigen werden nicht aktualisiert. Das kann notwendig sein, wenn ein Eingang von mehreren Kanälen abhängt, welche sich gleichzeitig ändern.

Ein- und Ausgangskonfiguration

Jedes Instrument, welches an der seriellen Kommunikation teilnimmt, hat im Eigenschafteneditor ein oder mehrere blau bzw. grün hinterlegte Felder in denen die Ein- bzw. Ausgänge eingegeben werden. Für Eingänge mit Zeichenformatierung "S" und für Ausgänge kann jeweils nur ein Kanal angegeben werden, Formeln sind nicht erlaubt. Dies betrifft beispielsweise Taster (Ausgang) und Fernanzeigen (Zeicheneingang).

Beispiel Ausgang:

Signalausgang => #9

Beispiel Eingang für Zeichenketten:

Signaleingang = #7

Numerische Eingänge

Instrumente, die numerische Eingangswerte anzeigen oder verarbeiten, wie z.B. das Zeitdiagramm oder die Analoganzeige, berechnen ihren Wert aus einer Formel, welche ebenfalls im blau hinterlegten Feld mit Aufschrift Eingang im Eigenschafteneditor eingegeben wird. Im einfachsten Fall besteht die Formel aus nur einem Kanal und unterscheidet sich damit nicht von der eines Ausgangs. Es können aber auch komplizierte Formeln in ähnlicher Weise und genauso einfach wie in einer Tabellenkalkulation eingegeben werden.

Beispiel eines numerischen Eingangs:

Eingang = #5* (min(10;#6)+3)

Im Beispiel sieht man die weitreichenden Möglichkeiten, welche die Comvisu für die Eingangsberechnung bereitstellt. Es sind die Grundrechenarten vorhanden, es gilt die Punkt-vor-Strich-Reihenfolge, es können Klammern gesetzt, es können konstante

Zahlenwerte angegeben, es können mehrere Kanäle verwendet und es können die vordefinierten Funktionen, hier die Funktion "min()", genutzt werden.

Die Ein- und Ausgänge werden beim Wechsel in den Ausführenmodus überprüft und übersetzt/kompiliert. Ist der Haken im Menü Anzeige/Kompilierstatus gesetzt, dann wird der Rahmen des entsprechenden Instrumentes jeweils *grün* gefärbt, wenn es sich um eine gültige Formel bzw. Ein- oder Ausgang handelt, jeweils *rot* gefärbt, wenn sie fehlerhaft ist und jeweils *grau* gefärbt, wenn kein Ein- oder Ausgang definiert ist.

Operatoren

Folgende Operatoren stehen bei numerischen Eingängen zur Verfügung:

Arithmetische Operatoren:

'+' - Addieren

'-' - Subtrahieren

'*' - Multiplizieren

'/' - Dividieren

'mod' - Modulo (Rest der Division)

'^' - Potenzieren

'()' - Klammern

Vergleichsoperatoren:

'==' - Gleichheit

'!=' - Ungleichheit

'<' - kleiner als

'>' - größer als

'<=' - kleiner gleich

'>=' - größer gleich

Bitoperatoren:

'bitand' - bitweises Und

'bitor' - bitweises Oder

'bitxor' - bitweises Exklusiv-Oder

Anmerkung: Bitweise Negation siehe *Funktionen*

Logische Operatoren (Bool'sche Algebra):

'and' - Und-Verknüpfung

'or' - Oder-Verknüpfung

'xor' - Exklusiv-Oder-Verknüpfung

Anmerkung: Negation siehe *Funktionen*

Es gilt die übliche mathematische Berechnungsreihenfolge Klammer vor Potenz vor **Punkt vor Strich** (vor Bitoperation vor Vergleich vor logischer Operation).

Vergleichsoperatoren liefern als Ergebnis die Zahlenwerte 0 oder 1. logische Operatoren interpretieren positive Werte als 1, die 0 und negative Werte als 0. Sie liefern als Ergebnis die Werte 0 oder 1.

Funktionen

Die Comvisu stellt verschiedene Funktionen zur Verfügung. Es gibt Funktionen mit fester Anzahl an Funktionsargumenten und solche mit variabler Anzahl. Argumente werden mit Strichpunkt ';' getrennt (da sowohl Punkt als auch Komma als Dezimaltrennzeichen interpretiert werden). Zwischen Groß- und Kleinschreibung wird nicht unterschieden.

Funktion	Beschreibung	Anzahl Argumente	Beispiel
<i>min</i>	Gibt den Mindestwert der angegebenen Argumente zurück. Benötigt mindestens zwei Argumente.	2 +	min(5;#7;#8)
<i>max</i>	Gibt den Maximalwert der angegebenen Argumente zurück. Benötigt mindestens zwei Argumente.	2 +	max(5;#7;#8)
<i>pi</i>	Gibt den Wert Pi= 3,1415926535 zurück.	0	pi()
<i>sheet</i>	Wählt Arbeitsblatt zur Anzeige aus. Blätter sind mit 1 beginnend indiziert.	1	sheet(#5)
<i>if</i>	if(<i>Bedingung</i> ; <i>ErgebnisWahr</i> ; [<i>ErgebnisFalsch</i>]);	2 - 3	if(#5<9;25;#6)
<i>sqrt</i>	Zieht die Quadratwurzel aus dem Argument	1	sqrt(#5)
<i>nRoot</i>	<i>Argument1</i> -te Wurzel aus <i>Argument2</i>	2	nRoot(3;125)
<i>eule</i>	Eulersche Zahl = 2,7182818285	0	eule()
<i>Log</i>	Zehnerlogarithmus aus dem Argument	1	log(#5)
<i>Ln</i>	Natürlicher Logarithmus aus dem Argument	1	ln(#1)
<i>abs</i>	Betrag vom Argument	1	abs(#5)
<i>sin</i>	Sinus vom Argument im Bogenmaß	1	sin(#5)
<i>cos</i>	Cosinus vom Argument im Bogenmaß	1	cos(rad(#5))
<i>tan</i>	Tangens vom Argument im Bogenmaß	1	tan(pi()/4)
<i>arcsin</i>	Inverser Sinus vom Argument im Bogenmaß	1	arcsin(#5)
<i>arccos</i>	Inverser Cosinuss vom Argument im Bogenmaß	1	arccos(#5)
<i>arctan</i>	Inverser Tangens vom Argument im Bogenmaß	1	arctan(#5)
<i>deg</i>	Umrechnung eines Winkels vom Bogenmaß (Argument) in Grad (Ergebnis)	1	deg(arcsin(#5))
<i>rad</i>	Umrechnung eines Winkels von Grad (Argument) in das Bogenmaß (Ergebnis)	1	sin(rad(#5))
<i>time</i>	Aktuelle Uhrzeit als Zahlenwert im Format <i>hhmmss</i>	0	time()

<i>date</i>	Aktuelles Datum als Zahlenwert im Format YYYYMMDD	0	date()
<i>datetime</i>	Aktuelles Datum und Uhrzeit als Zahlenwert im Format YYYYMMDDhhmmss	0	datetime()
<i>null</i>	Gibt unabhängig von den Parametern den Wert Null (0) zurück.	1 +	null(#5)
<i>ans</i>	Gibt das Ergebnis (eng: Answer) der vorigen Berechnung zurück	0	0,7*ans()+0,3*#5
<i>round</i>	Rundet den Funktionswert auf volle Stellen oder auf die <i>n</i> te Nachkommastelle, wenn zweites Funktionsargument vorhanden	1 - 2	round(#5) round(#5; 2)
<i>not</i>	Logische Negation des Arguments (für Vergleiche)	1	if(not(#5<0);2;4)
<i>bitnot</i>	Bitweise Negation des ersten Arguments, interpretiert als Ganzzahl mit der Bitbreite angegeben im zweiten Argument (1 bis 32 Bit).	2	bitnot(#5;16)
Speicherwertige Funktionen			
<i>count</i>	Zählt Anzahl der eigenen Aufrufe	0	count() +null(#5)
<i>last</i>	Gibt den Argumentwert des letzten Aufrufs zurück (Verzögerung um einen Aufruf)	1	last(#5)
<i>z</i>	Gleiche Funktion wie <i>Last()</i> nur anderer Name	1	z(#5)
<i>seqMin</i>	Gibt den kleinsten Wert aller Aufrufe zurück	1	seqMin(#5)
<i>seqMax</i>	Gibt den größten Wert aller Aufrufe zurück	1	seqMax(#5)
<i>FloatAvg</i>	Gibt den gleitenden Mittelwert (eng: Floating Average) vom ersten Argument zurück. Das zweite Argument gibt die Anzahl der berücksichtigten Werte an.	2	floatAvg(#5;25)

Kommandozeilenparameter

Wenn das Programm über die Eingabeaufforderung (oder über eine Verknüpfung oder Batch-Datei) gestartet wird, ist die Angabe einer Projektdatei und von Startoptionen möglich.

`Comvisu.exe [Projektdatei] [-Startoption1] [-Startoption2] etc.`

Der erste Parameter nach dem Programmnamen gibt die zu ladende Projektdatei mit der Endung `.visu` an. Darauffolgend können folgende Startoptionen, auch in Kombination, angegeben werden:

Parameter	Beschreibung
<code>-FullScreen</code>	Programm startet im Vollbildmodus
<code>-FullWindow</code>	Programm startet im Vollfenstermodus
<code>-Connect</code>	Programm startet im Ausführen-Modus und verbindet Schnittstellen
<code>-RunMode</code>	Programm startet im Ausführen-Modus
<code>-Lang=de</code>	Programm startet in deutscher Sprache
<code>-Lang=en</code>	Programm startet in englischer Sprache
<code>-Lang=zh</code>	Programm startet in chinesischer Sprache

Beispiele:

```
Comvisu.exe TestProjekt.visu
```

```
Comvisu.exe TestProjekt.visu -FullScreen -Connect -Lang=en
```

Anderes

/ ComVisu ist in FPC/Lazarus geschrieben und verwendet deren Bibliotheken. Quellcode, Binaries, Lizenzhinweise und Informationen zu FPC und Lazarus unter <http://www.lazarus-ide.org>. Lizenz: Modifizierte LGPL.

/ ComVisu verwendet die Komponente TACChart. Download- und Lizenzinformationen unter <http://wiki.freepascal.org/TACChart>. Lizenz: Modifizierte LGPL.

/ ComVisu (nur Linuxversion) verwendet die Bibliothek Free Pascal Qt4 Binding. Download- und Lizenzinformationen unter <http://users.telenet.be/Jan.Van.hijfte/qtforfpc/fpcqt4.html> oder unter http://wiki.lazarus.freepascal.org/Qt4_binding. Lizenz: Modifizierte LGPL.

/ ComVisu verwendet die Schnittstellenbibliotheken Ararat Synapse und Synaser, welche in übersetzter Form im Programm integriert sind. Für Ararat Synapse und Synaser gelten die folgenden Bedingungen:

```
Copyright (c)1999-2002, Lukas Gebauer
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

Neither the name of Lukas Gebauer nor the names of its contributors may
be used to endorse or promote products derived from this software without
specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.
```

Lizenz

Nur zum privaten Gebrauch.

Die gewerbliche Nutzung ist nicht ohne schriftliche Einwilligung des Rechteinhabers gestattet.

Haftungsausschluss

Der Autor haftet nicht für Schäden, die aufgrund des Einsatzes oder einer Fehlfunktion des Programmes resultiert. Mit Benutzen des Programms erklärt sich der Nutzer mit diesem Haftungsausschluss einverstanden. Die Verwendung in sicherheitsrelevanten Anwendungen ist nicht gestattet. Die Verwendung in Anwendungen, wo eine Fehlfunktion des Programmes zur Gefährdung von Menschen und Sachgütern führen kann, ist nicht gestattet.

Autor und Rechteinhaber:
Dipl.-Ing. Janik Österle