



Ingenieurbüro für Industrieautomatisierung

Adresse:

Im Vorderen Großthal 4

D 76857 Albersweiler /Pfalz

Tel:

+49 6345 9496732

Mobil: +49 171 4311359

e-mail:

sales@heisch-automation.de

www.heisch-automation.de

Pult

V 0.4

Bedienungsanleitung

Stand 22.06.2016

(deutsche Version)

Inhaltsverzeichnis

1. Copyrights und Recht zum Kopieren.....	2
2. Über Pult.....	3
3. Installation von Pult2.....	4
4. Programmaufruf und erster Start.....	5
5. Zusammenspiel der Komponenten.....	11
6. Datenkonsistenz: Direkt Modus, Kommando-DB Modus.....	12
7. Hinweise zur XForms Bibliothek	13

1.Copyrights und Recht zum Kopieren

Benutzte Markennamen

Simatic, Simatic S5 und Simatic S7 sind geschützte Produktbezeichnungen der Siemens AG.

rkcat Copyright(c) 2002 ff. Werner Heisch (Heisch Automation)

Pult ist kein geschützter Begriff, dieses Programm Pult (c) Werner Heisch (Heisch Automation)

rk511_server, rk512_server, rktcp_server Copyright(c) 2002 ff. Werner Heisch
(Heisch Automation)

Recht zum Kopieren

Diese Software kann für nicht-kommerzielle Zwecke frei verteilt und vertrieben werden, solange die oben stehenden Copyrightvermerke und diese Betriebserlaubnis in allen Kopien im Source-Code oder als separate Datei enthalten sind.

Kommerzielle Nutzung dieser Software erfordert eine gesonderte Erlaubnis des Autors.

Kommerzielle Nutzung bedeutet in diesem Zusammenhang der Vertrieb dieser Software oder Teile daraus gegen Bezahlung.

Die reine Benutzung dieser Software gilt immer als nicht kommerziell.

Kopiererlaubnis

Unser Interesse besteht darin, die Server der Simatic-Koppelsoftware-Familie rk*_server zu verkaufen.

Falls es kommerzielle Applikationen gibt, die auf diesen Servern basieren, so partizipieren wir automatisch an ihrem Erfolg.

Wir wollen lediglich nicht die Arbeit machen, während andere damit das Geld verdienen.

Deshalb kann rkcat oder Teile daraus frei und kostenlos in kommerzielle Programme eingearbeitet werden, wenn die daraus entstehenden Applikationen unterschiedlich zu rkcat oder entsprechend höherwertig sind. Wir werden uns aber die letzte Entscheidung vorbehalten.

Gewährleistung

Benutzung dieser Software auf eigene Gefahr.

Die Autoren übernehmen keine Haftung für eventuell auftretende Schäden, die sich im Zusammenhang mit der Nutzung dieser Software ergeben!

2.Über Pult

Pult sollte das klassische Testpult mit Lampen und Schaltern ersetzen, mit dem die Simatic-Programmierer der früheren Jahre ihre Programme testeten.

Sowohl Simatic S5 als auch Simatic S7 werden unterstützt.

Pult2 basiert auf den Linux-Simatic-Kommunikations-Servern der rk*_server- Familie.

Da rk512_server, rk511_server und rktcp_server client-seitig die gleiche Schnittstelle benutzen, können sie beliebig gegeneinander ausgetauscht werden. Sie werden im Folgenden entsprechend mit rk*_server bezeichnet.

Pult2 ermöglicht das Bedienen von 48 "Schaltern" und das Ansteuern von 128 "Lampen" sowie die Ein- und Ausgabe von jeweils 18 Worten, bzw. Doppelworten.

In der vorliegenden Version müssen alle zu bedienenden und abzufragenden Elemente in Datenbausteinen liegen. Eine gegebenenfalls notwendige Rangierung in der angeschlossenen Simatic sollte für den Simaticer sicherlich kein Problem darstellen.

Pult2 kann bei Bedarf mehrmals aufgerufen werden, der Einsatz zusammen mit „LogAn“, „rk_cat“ oder anderen rk*_server-basierten Programmen ist problemlos.

Features gegenüber Pult:

- Höhere Transparenz hinsichtlich der Kopplung an die Simatic
- Das Structure-file beinhaltet zusätzlich den Eintrag für das rc-file
- 128 Ausgangsbits gegenüber 96
- 18 Analog-Eingangswerte (Wort bzw Doppelwort) gegenüber 6
- 18 Analog-Ausgangswerte (Wort bzw Doppelwort) gegenüber 6
- Anzeige der Namen der Variablen plus Auskunftsfunktion
- Auskunft über den gekoppelten Server.
- Serverkopplung ein-/ausschaltbar.
- Keine Software in der Simatic notwendig:
Pult hat zur Wahrung der Datenkonsistenz die Datenvorgabe an die Simatic generell nur über einen Kommando-DB und einen zugehörigen Interpreter (FB / FC) erledigt,
Pult2 kann die Datenvorgabe auch direkt erledigen.

3. Installation von Pult(2)

Voraussetzungen:

Pult setzt auf einer installierten **Xforms** - bzw. Xforms-Toolkit Grafikbibliothek auf.

Xforms ist „altmodisch“: nicht objektorientiert, klein schnell robust und eignet sich daher auch für „ältere Mitbürger“ (wie dem Verfasser), die keine Zeit (mehr) haben, sich langwierig in objektorientierte Programmierung einzuarbeiten, nur weil sie ein kleines Programm schreiben wollen / müssen.

Xforms beinhaltet auch das Programm "fdesign", mit dem Sie die grafische Oberfläche Ihrer Applikation erstellen können.

Sollten Sie Xforms noch nicht kennen und noch nicht installiert haben, auf unserer Homepage sind ein paar Hinweise sowie Links.

Pult2 setzt natürlich auch einen lauffähigen **rk*_server** voraus, es ist dabei unwichtig, ob es sich um einen „rk512_server“, „rk511_server“, „rktcp_server“ oder einfach nur zum Test um einen „rk512_sim“ handelt. Es ist dabei ebenfalls unerheblich ob es eine lizenzierte Version oder eine Demoversion ist.

Es soll allerdings ein aktueller rk*_server sein, d.h. V5.10 oder höher.

Installation von pult2

1. Auspacken in ein eigenes Subdirectory
2. Kopieren Sie die Dateien "rk7_client.o" und "rk7_client.h" aus Ihrem rk*-Server-Verzeichnis in das ./src Verzeichnis.
Dies ist wegen der 32/64-Bit- Problematik notwendig, schließlich müssen Sie kompilieren. Damit stellen Sie außerdem sicher, daß die client-Programme optimal auf die rk*-server abgestimmt sind.
3. Ihr src-Verzeichnis beinhaltet die drei Object-Dateien "s5types_32.o", "s5types_64.o" und "s5types_ARMv6.o".
Wählen Sie entsprechend Ihrem Prozessor (32/64Bit / Raspberry) eine der Dateien aus und kopieren Sie sie nach "s5types.o".
(Diese Dateien sind Bestandteil unseres Pakets "convert_lib" und werden nur dort als Quelltext ausgeliefert.)
4. Kompilieren Sie die Applikation mit "make clean" und dann "make"
5. fertig

4. Programmaufruf und erster Start

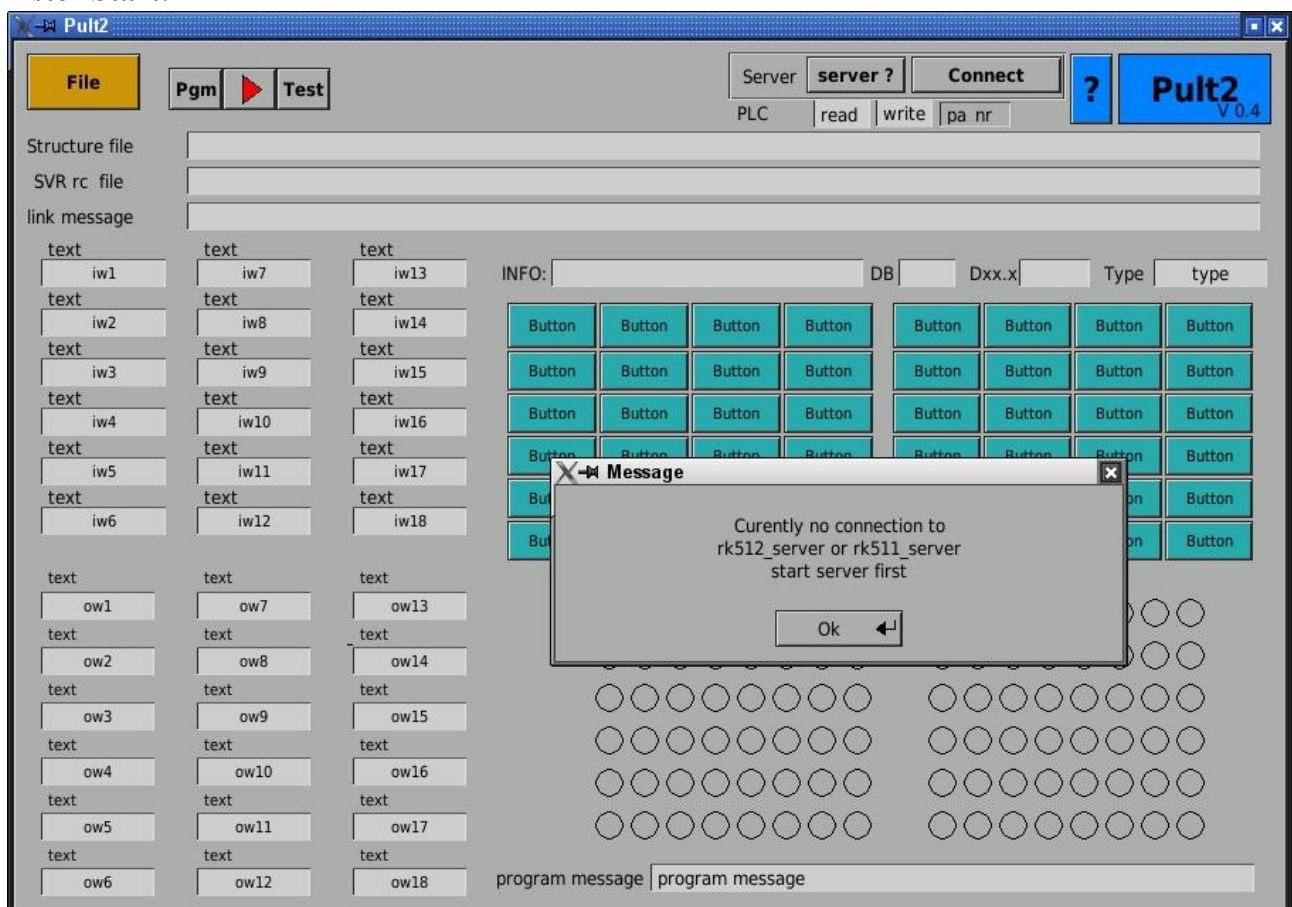
```
pult <rk512_rc_file> // alte Methode
oder:
NEU:
pult [-r <rk512_rc_file> ] [-s <struct_file>]
while:
  [-r <rk512_rc_file>] : rk512-rc-file
  [-s <stru_file>]      : data structure file
```

Normalerweise ist nur das <stru_file> definiert.

Parameter :

- h : Hilftext (Usage) ausgeben.
- s <stru_file> : Darin ist normalerweise alles Notwendige definiert, nicht natürlich nicht, Wenn die Datei noch nicht erstellt ist.
- r <rk512_rc_file> : rk512-rc-file, Falls zusätzlich zu dem in der Konfigurationsdatei in der Kommandozeile ein rc-file angegeben wurde dann gilt das in der Kommandozeile angegebene.

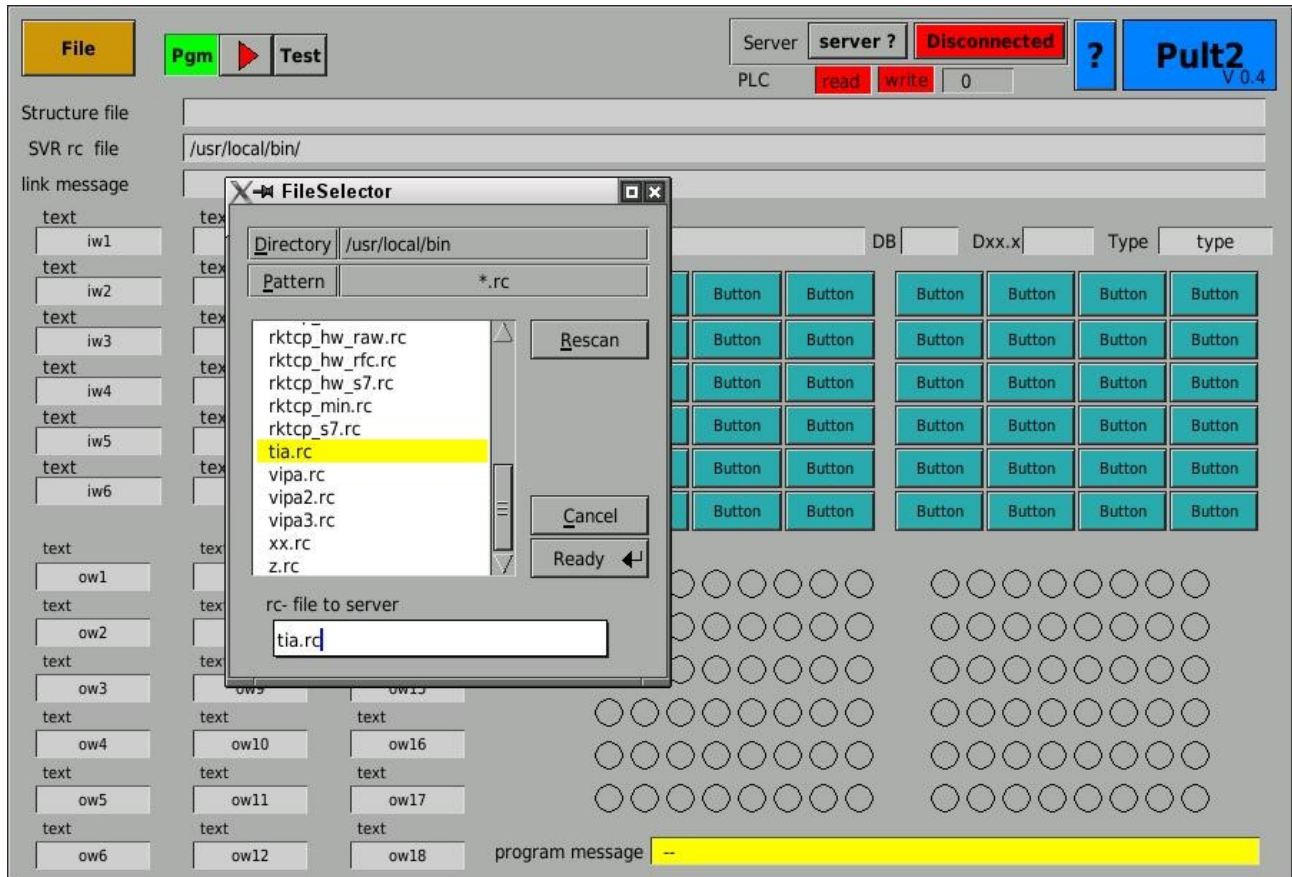
Erster Start:



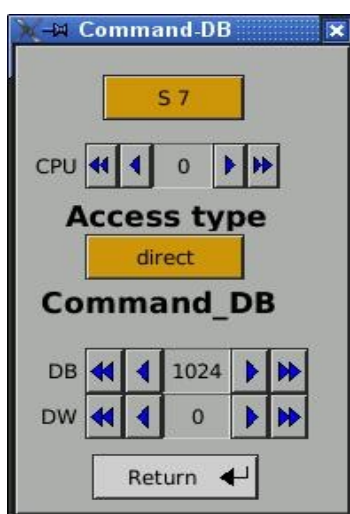
Es macht Sinn, als erstes die Verbindung zu Server zu suchen: Das rc-file.

FILE-> "load Server rc file"

Wählen Sie das rc-file aus, mit dem der Server läuft :



Mit FILE-> "define Simatic" wir dann die Simatic definiert:



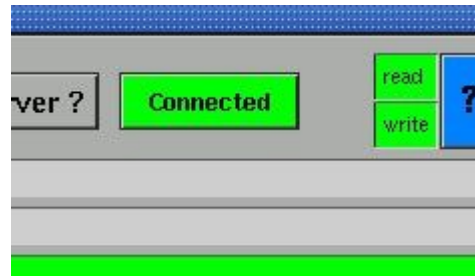
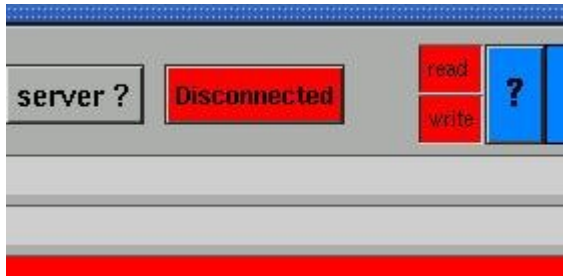
Zur Auswahl stehen "S7" und "S5": die S7 hat byteorientierte Adressnummern, die S5 hat wortorientierte.

Dann wählen Sie die Nummer der CPU an, die im rc-file für die as Ziel-AG definiert ist. Normalerweise wird dies "0" sein.

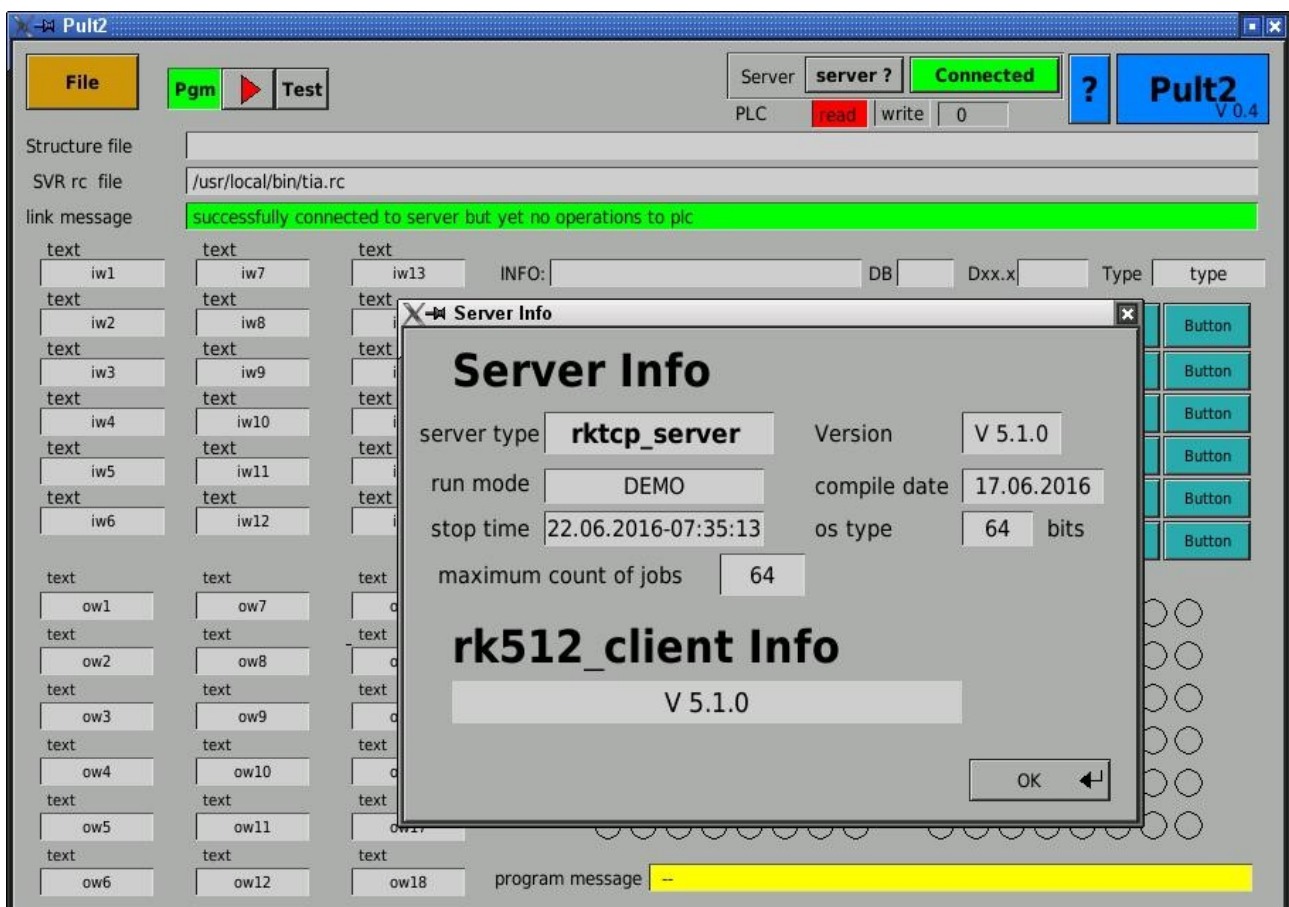
"Accesss Type" : für den Anfang "direct" die Methode mit dem Kommando-DB wird weiter unten erklärt.

Fertig.

Nun kann die Verbindung zur SPS hergestellt werden: Drücken sie auf den Schalter "Disconnected". Dies sollte die Verbindung zum Server herstellen.



Mit der Schaltfläche "server ?" können Sie den Verbindungsstatus sehen.



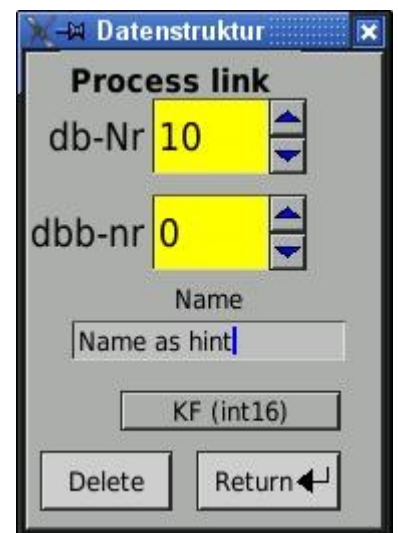
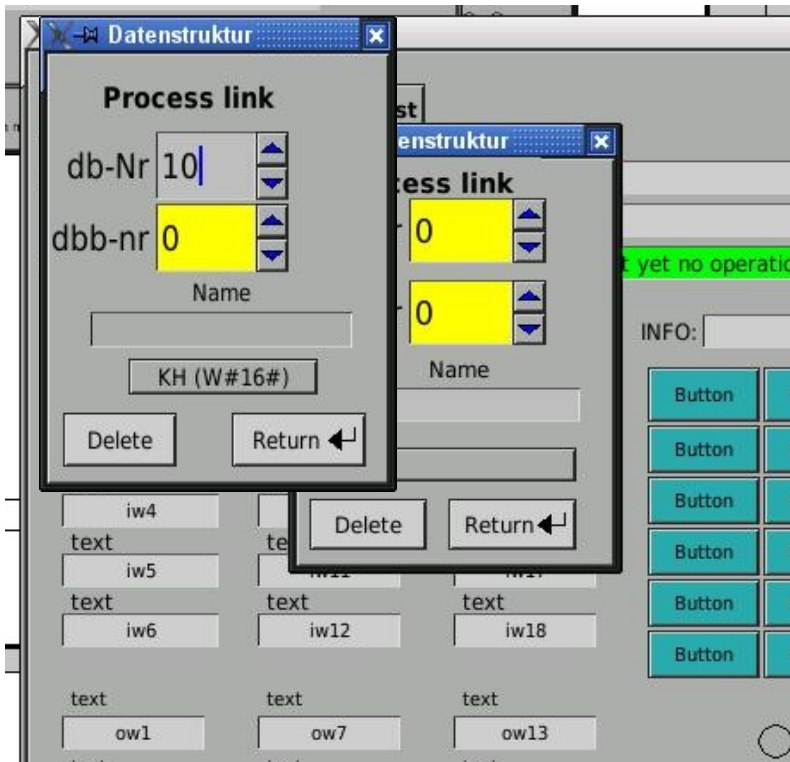
Als nächstes werden die interessierenden Variablen in der SPS angelegt.

ANALOG-Signale Eingänge:

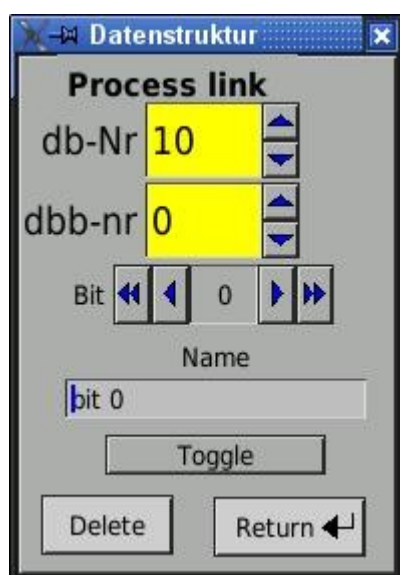
Klicken Sie auf einer der Eingabefelder (der Block links oben)

Wählen Sie hier den Datenbaustein, die Startadresse und das Datenformat aus.

Klicken Sie auf eines der Ausgabefelder (der Block links unten)



Wählen Sie auch eine oder mehrere Bit-Eingänge. Sie können auswählen, ob bei Tastendruck das Bit gesetzt, rückgesetzt oder getoggelt wird.

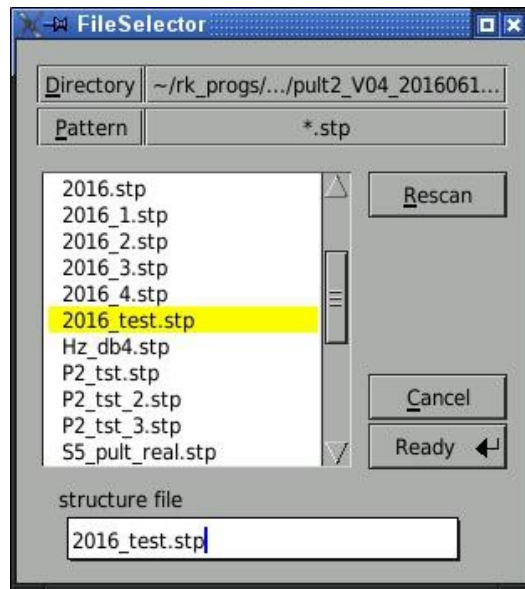


Ausgangsbits werden genau so selektiert, haben aber natürlich keine zu bearbeitende Funktion.

Nach der Definition der Parameter kann alles zusammen in einem structure file gespeichert werden:

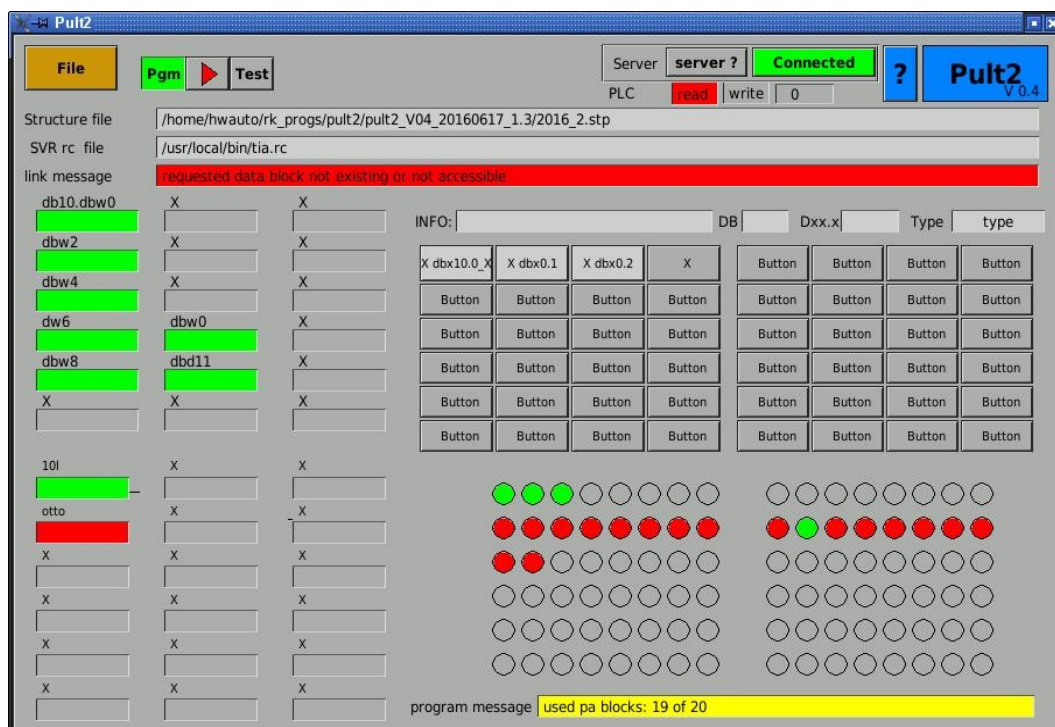
"FILE" -> "store structure" -> zum Ziel-Verzeichnis navigieren und den Namen der files angeben.

Die Dateiergung ".stp" wird automatisch ergänzt.



Testen Sie nun, ob die bereits definierten Variablen auch existieren. Drücken Sie "Test".

Beispiel:



Die definierten und erreichbaren Variablen werden grün angezeigt, die nicht erreichbaren rot.

Starten:

Sie starten die Funktion mit der play-Taste.

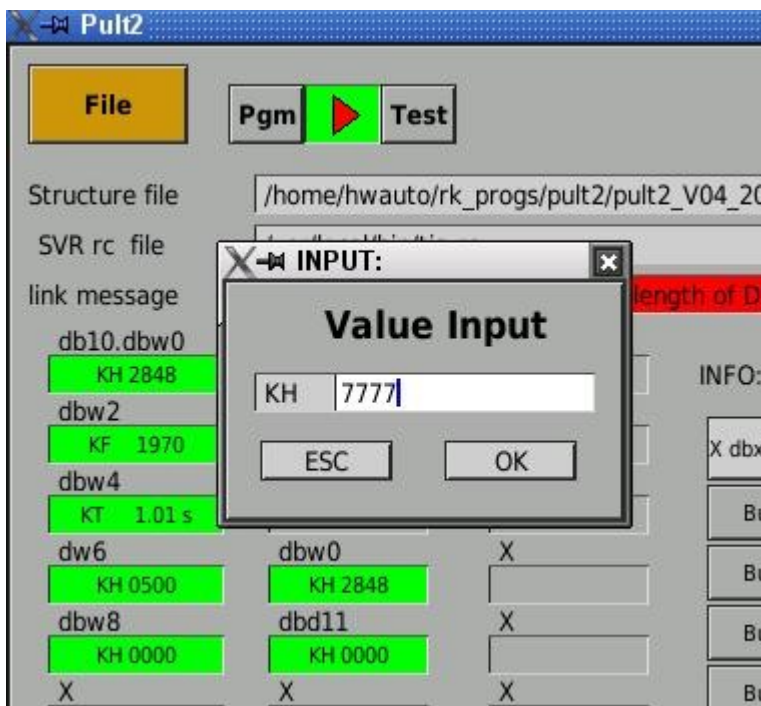


Die Daten werden nun aktualisiert.

Bedienung:

Taster durch anklicken mit der linken Maustaste.

Eingabefelder Analog: Anklicken mit der linken Maustaste, es erscheint ein Eingabefenster:



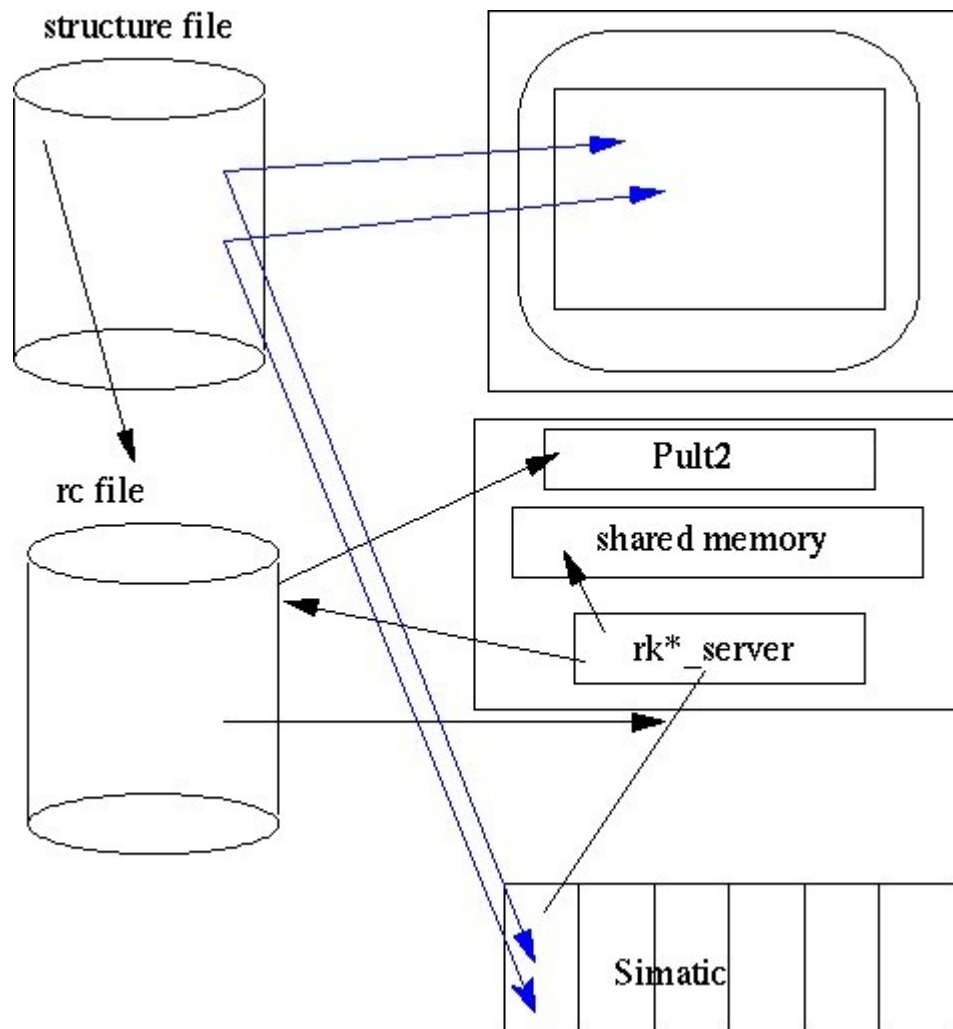
INFO:

Anzeige der Adresse und des Namens einer Variablen:

Bei Ausgabe-Variablen: immer beim anklicken.

Bei Eingabe-Variablen und Tastern: durch Anklicken mit der rechten Maustaste..

5. Zusammenspiel der Komponenten



Wo geht's hier zur Simatic ?

Das rc file teilt dem Server beim Start mit, unter welchem Namen er shared memory allozieren soll. Das rc-file wird ebenfalls von den client-Prozess "Pult" gelesen, um zu erfahren, unter welchem Namen er das shared memory erreicht, denn darüber kommunizieren client (hier: "Pult2") und Server.

Wo liegen die Daten in der Simatic ?

Im structure file können die Daten definiert werden, die auf Pult angezeigt werden. Im structure file ist außerdem der Name des rc files hinterlegt.

6. Datenkonsistenz: Direkt Modus, Kommando-DB Modus

Hintergrund:

Es ist über keinerlei Kopplung möglich, ohne Zuhilfenahme von Interpretern Variablen bis auf Bit-Ebene unter Wahrung der Datenkonsistenz zu manipulieren, da die gängigen Übertragungsverfahren (RK511, RK512, Profibus, Ethernet) in der Regel als kleinste Einheit ein Byte oder ein 16-Bit-Wort transportieren.

Einige Visualisierungssysteme gehen, um ein Bit zu manipulieren, generell folgenden Weg:
Wort mit Bit aus der SPS holen(1), Bit in Rechner manipulieren(2), Wort in die SPS zurückschreiben(3).

Dadurch ist aber die Datenintegrität nicht gewahrt, d.h. alle Bits in dem Wort, die sich seit dem Lesen in der SPS(1) verändert haben (durch SPS, Bedien-Panel oder ein weiteres Visualisierungssystem) werden nun wieder mit ihrem alten Wert überschrieben.

Allerdings: werden die Bit-Kommandos konsequent getrennt, d.h. nur ein Rechner / eine Visualisierung schreibt auf das betroffene Wort, die Simatic liest nur, ergibt sich diese Problematik natürlich nicht.

Da Pult2 / Pult aber eine Testhilfe ist, kann eine klare Trennung nicht vorausgesetzt werden. Um zu verhindern, dass es beim Einsatz von *Pult2* zu entsprechenden Dateninkonsistenzen kommt, der Programmierer daraufhin „Programmierfehler“ sucht, die sich eigentlich nur aus dem Testaufbau ergeben, wurde auf die gängige Methode des Kommando-FBs zurückgegriffen: Ein FB (S7: FC) wertet ein gesendetes Kommando (z.B. Setze in DB45 das D 4.5) aus und führt es durch.

Für S5 bzw. S7 liegen jeweils ein FB/FC sowie ein Kommando-DB bei. Die Bausteine sind frei wählbar.

Der FB/FC muss lediglich in die SPS überspielt und im OB 1 aufgerufen werden.
Als Parameter wird der Kommando-DB angegeben werden.

Pult2 kann mehrere Kommandos zwischenspeichern und schickt jeweils ein Kommando an den Kommando-DB, das der FB/FC im gleichen Zyklus verarbeitet.

!!!! Hinweis:

!!!! Der Kommando-FB für die S5 überprüft nicht, ob die angegebene Zieladresse
!!!! zulässig ist.

!!!! Falls also z.B. auf ein nicht existierendes Datenbit geschrieben werden soll,
!!!! so geht die Steuerung auf Stop.

!!!! Der Einsatz des beiliegenden Kommando-FBs ist deshalb nur zu Testzwecken
!!!! zulässig, zum Betrieb einer Anlage ist der Einsatz aus o.a. Grund untersagt.

Hinweis zum Aufbau des Kommando-DB:

Da die RK511 (S5) und TCP/IP (in S7) asynchron zum Zyklus der Steuerung Daten überträgt, stellt ein Prüfwort am Ende des Kommando-DB sicher, daß die Daten komplett übertragen wurden. Der Kommando-FB führt das Kommando nur aus, wenn das erste und das letzte Wort im Kommando-DB gleich sind.

7. Hinweise zur XForms Bibliothek

Der Begriff XForms steht als Abkürzung für **Forms Library, A Graphical User Interface Toolkit for X**. XForms ist eine für heutige Begriffe sehr schlanke Grafikbibliothek. Entwickelt bereits vor ein paar Jahren von T.C.Zhao und Mark Overmars, bietet sie immer noch eine gute Grundlage für die Anwendungsentwicklung unter X11.

Xforms verfügt außerdem über eine hervorragende Dokumentation sowie über eine ganze Reihe von beiliegenden kleinen Beispielprogrammen, die es besonders einem Anfänger sehr erleichtern, schnell zu eigenen Applikationen zu kommen.

Xforms ist hochportabel. Kennzeichnend für die Portabilität und somit auch die Brauchbarkeit für Systeme, die nicht KDE oder Gnome verwenden, ist folgendes Zitat aus der Dokumentation:

The library uses only the services provided by the Xlib and should run on all workstations that have X installed on them.

Es wird bei der Weiterentwicklung außerdem sehr viel Wert auf Abwärtskompatibilität gelegt. Vermutlich sind es besonders diese Eigenschaften, weswegen die library in der universitären Forschung nach wie vor eingesetzt wird. Xforms ist leider in den aktuellen Linux-Distributionen nicht mehr enthalten.

Das ist aber kein Problem, denn Xforms ist einfach zu installieren:

- Download von der [XForms Downloadpage](#).
- Wählen Sie die neueste xforms-Version aus.
- **Tipp:** Laden Sie die Dokumentation "xforms.pdf" ebenfalls herunter.
- **Tipp:** Xforms benötigt zusätzlich die libXpm-devel und die libjpeg-devel -Bibliotheken. Sie sind Bestandteil jeder gängigen Distribution.
- Auspacken des Archivs in ein eigenes Verzeichnis und wechseln dorthin.
- ./configure
- (gg. make clean)
- make

Als root geht es weiter mit

- make install
- ldconfig

Nach dieser Bearbeitung befinden sich sowohl die Laufzeitumgebung als auch die Entwicklerfiles auf der Platte, und zwar unterhalb von /usr/local/. /usr/local/bin enthält außerdem den GUI-Designer "fdesign". Zum Linken von Xforms-Applikationen ist ggf. die Option -lXpm anzugeben.