WAGO I/O SYSTEM 750

Nutzung der SysLibFile.lib und der WagoLibFTP.lib für Dateizugriffe

Anwendungshinweis

A114100, Deutsch
Version 1.0.0

WAGO® INNOVATIVE CONNECTIONS

Copyright © 2006 by WAGO Kontakttechnik GmbH & Co. KG
Alle Rechte vorbehalten.

WAGO Kontakttechnik GmbH & Co. KG
Hansastraße 27
D-32423 Minden

Tel.: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

E-Mail: info@wago.com

Web: http://www.wago.com

Technischer Support
Tel.: +49 (0) 571/8 87 – 5 55
Fax: +49 (0) 571/8 87 – 85 55

E-Mail: support@wago.com

Es wurden alle erdenklichen Maßnahmen getroffen, um die Richtigkeit und Vollständigkeit der vorliegenden Dokumentation zu gewährleisten. Da sich Fehler, trotz aller Sorgfalt, nie vollständig vermeiden lassen, sind wir für Hinweise und Anregungen jederzeit dankbar.

Wir weisen darauf hin, dass die im Dokument verwendeten Soft- und Hardwarebezeichnungen und Markennamen der jeweiligen Firmen im Allgemeinen einem Warenzeichenschutz, Markenschutz oder patentrechtlichem Schutz unterliegen.

INHALTSVERZEICHNIS

1 Wichtige Erläuterungen 4
1.1 Rechtliche Grundlagen 4
1.1.1 Urheberschutz 4
1.1.2 Personalqualifikation 4
1.1.3 Bestimmungsgemäßer Gebrauch 4
1.2 Gültigkeitsbereich 5
1.3 Symbole 5

2 Beschreibung 6
2.1 Material 6

3 Nutzung der SysLibFile.lib zum File handling 6

4 Nutzung der WagoLibFTP.lib zum Senden eines Files 8

5 Beispiele 9
5.1 Daten in ein File schreiben und verschicken 9
5.2 Konvertierung von Daten 10
5.2.1 Visualisierung von Daten 11
5.2.2 CSV Datei einlesen 12

1 Wichtige Erläuterungen

Um dem Anwender eine schnelle Installation und Inbetriebnahme der beschriebenen Geräte zu gewährleisten, ist es notwendig, die nachfolgenden Hinweise und Erläuterungen sorgfältig zu lesen und zu beachten.

1.1 Rechtliche Grundlagen

1.1.1 Urheberschutz

Dieses Dokument, einschließlich aller darin befindlichen Abbildungen, ist urheberrechtlich geschützt. Jede Weiterverwendung dieses Dokumentes, die von den urheberrechtlichen Bestimmungen abweicht, ist nicht gestattet.
Die Reproduktion, Übersetzung in andere Sprachen, sowie die elektronische und fototechnische Archivierung und Veränderung bedarf der schriftlichen Genehmigung der WAGO Kontakttechnik GmbH & Co. KG, Minden. Zuwiderhandlungen ziehen einen Schadenersatzanspruch nach sich.

Die WAGO Kontakttechnik GmbH & Co. KG behält sich Änderungen, die dem technischen Fortschritt dienen, vor.
Alle Rechte für den Fall der Patenterteilung oder des Gebrauchmusterschutzes sind der WAGO Kontakttechnik GmbH & Co. KG vorbehalten. Fremdprodukte werden stets ohne Vermerk auf Patentrechte genannt. Die Existenz solcher Rechte ist daher nicht auszuschließen.

1.1.2 Personalqualifikation

Der in diesem Dokument beschriebene Produktgebrauch richtet sich ausschließlich an Fachkräfte mit einer Ausbildung in der SPS-Programmierung, Elektrofachkräfte oder von Elektrofachkräften unterwiesene Personen, die außerdem mit den geltenden Normen vertraut sind. Für Fehlhandlungen und Schäden, die an WAGO-Produkten und Fremdprodukten durch Missachtung der Informationen dieses Dokumentes entstehen, übernimmt die WAGO Kontakttechnik GmbH & Co. KG keine Haftung.

1.1.3 Bestimmungsgemäßer Gebrauch

Die Komponenten werden ab Werk für den jeweiligen Anwendungsfall mit einer festen Hard- und Softwarekonfiguration ausgeliefert. Änderungen sind nur im Rahmen der in dem Dokument aufgezeigten Möglichkeiten zulässig. Alle anderen Veränderungen an der Hard- oder Software, sowie der nicht bestimmungsgemäße Gebrauch der Komponenten, bewirken den Haftungsausschluss der WAGO Kontakttechnik GmbH & Co. KG.

Wünsche an eine abgewandelte bzw. neue Hard- oder Softwarekonfiguration richten Sie bitte an WAGO Kontakttechnik GmbH & Co. KG.

1.2 Gültigkeitsbereich

Dieser Anwendungshinweis basiert auf die genannte Hard- und Software der jeweiligen Hersteller sowie auf die zugehörige Dokumentation. Daher gilt dieser Anwendungshinweis nur für die beschriebene Installation.
Neue Hard- und Softwareversionen erfordern eventuell eine geänderte Handhabung.

Beachten Sie die ausführliche Beschreibung in den jeweiligen Handbüchern.

1.3 Symbole



Gefahr
Informationen unbedingt beachten, um Personen vor Schaden zu bewahren.



Achtung
Informationen unbedingt beachten, um am Gerät Schäden zu verhindern.



Beachten
Randbedingungen, die für einen fehlerfreien Betrieb unbedingt zu beachten sind.



ESD (Electrostatic Discharge)
Warnung vor Gefährdung der Komponenten durch elektrostatische Entladung. Vorsichtsmaßnahme bei Handhabung elektrostatisch entladungsgefährdeter Bauelemente beachten.



Hinweis
Routinen oder Ratschläge für den effizienten Geräteeinsatz und die Softwareoptimierung.



Weitere Informationen
Verweise auf zusätzliche Literatur, Handbücher, Datenblätter und INTERNET Seiten.

2 Beschreibung

Im Rahmen dieser Application note wird der Einsatz der SysLibFile.lib und der WagoLibFTP.lib für den Wago Ethernet Controller 750-841 in Verbindung mit Datei Anwendungen erläutert. Es wird beschrieben, wie ein File erzeugt, versendet und gelesen werden kann. Die Bausteine liegen im Source Code vor und dienen damit als Beispiel für die grundsätzliche Benutzung der angeführten Bibliotheken.

2.1 Material

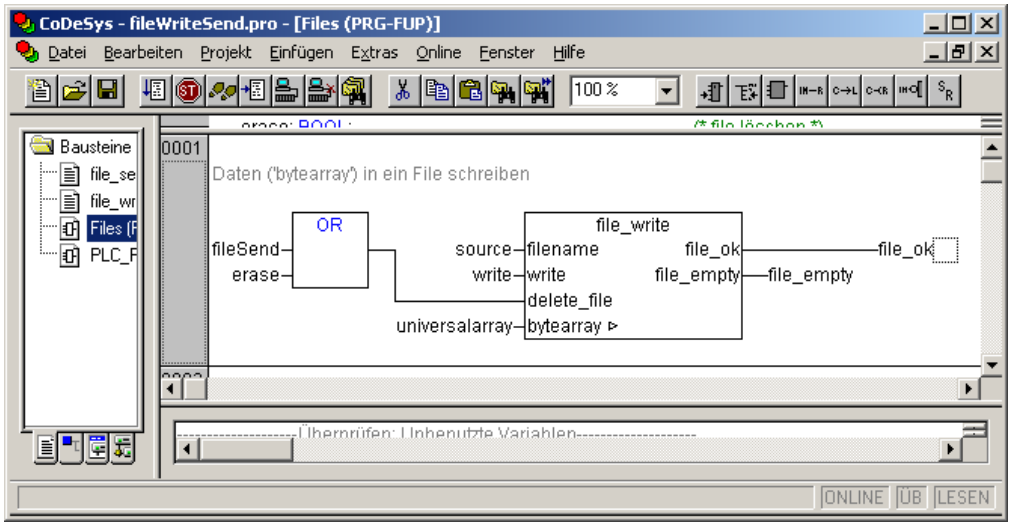
Die Anbindung ist unter den folgenden Voraussetzungen getestet worden:

Bezeichnung	Hersteller	Type
Ethernet-Controller	WAGO	750-841
SysLibFile.lib	3S	
WagoLibFtp.lib	WAGO	

3 Nutzung der SysLibFile.lib zum File handling

Es wird die Programmierung zum Schreiben in ein File erläutert. Als Datenübergabebereich ist ein Datenfeld (‚bytearray') definiert. ‚filename' definiert den Namen, unter dem die Datei dann im Flash gespeichert wird. Der Schreibvorgang wird über die boolsche Variable ‚write' eingeleitet. Mit ‚delete_file' wird die über ‚filename' angegebene Datei gelöscht.

Die boolschen Ausgänge ‚file_ok' und ‚file_empty' geben an, ob die Datei erfolgreich geschlossen wurde und ob die mit ‚filename' angegebene Datei leer ist.



Zum Schreiben eines Files sind grundsätzlich drei Schritte erforderlich:

1. öffnen einer Datei
2. beschreiben der Datei
3. schließen der Datei

Mit dem boolschen Eingang ‚write' wird zuerst

die Funktion „SysFileOpen" aufgerufen. Übergeben wird ihr der Dateiname (‚filename') und ein Zugriffsparameter (‚a'/‚r'/‚w'/‚rw') Es wird entweder eine neue Datei generiert oder eine bestehende Datei geöffnet. Rückgabewert ist die Datei-Nummer, die für die weiteren Funktionen benötigt wird.

Im nächsten Schritt wird die Funktion „SysFileWrite" aufgerufen. Diese braucht als Eingang die Datei-Nummer (aus „SysFileOpen"), die Adresse der zu schreibenden Daten (hier ADR(bytearray)), sowie die Anzahl der zu schreibenden Bytes.

Nachdem nun alle Daten geschrieben wurden muss die Datei mit „SysFileClose" geschlossen werden. Eingangsparameter ist die Datei-Nummer (aus „SysFileOpen"). Zurückgegeben wird ein boolscher Wert (‚file_ok').



Hinweis
Dieser Befehl arbeitet blockierend. Es ist deshalb sinnvoll, File Operationen in einer eigenen Task auszuführen.

Mit ‚delete_file' wird die Funktion „SysFileDelete" aufgerufen. Ihr wird der Dateiname (‚filename') übergeben und sie liefert eine boolsche Variable (‚file_empty') zurück.

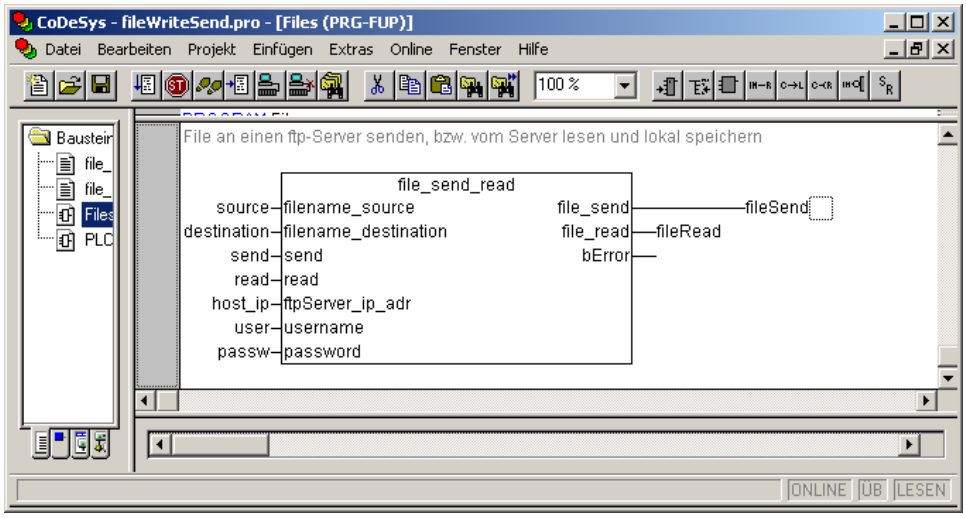
4 Nutzung der WagoLibFTP.lib zum Senden eines Files

Nachfolgend wird die Programmierung zum Versenden einer Datei erläutert. Hierzu wird der Baustein „file_send_read“ verwendet, der intern den FB „FTP_CLIENT“ (aus der „WagoLibFtp.lib“) benutzt.

Über ‚filename_source’ und ‚filename_destination’ werden die Quell- und Ziel-Datei-Namen angegeben. In ‚ftpServer_ip_adr’ wird die Ziel-IP-Adresse gespeichert. Die boolsche Variable ‚send’ startet dann das Verschicken der lokalen Datei zum Server.

Mit ‚read’ kann die Datei vom Server gelesen und auf dem lokalen Flash gespeichert werden.

Die boolschen Ausgänge ‚file_send’ und ‚file_read’ geben an, ob die Datei erfolgreich verschickt bzw. gelesen worden ist. Über bError kann man einen Fehlercode auslesen.



Zum Versenden oder Lesen eines Files sind grundsätzlich vier aufeinander folgende Schritte notwendig:

1. Verbindung zum Host herstellen
2. Account auf dem Host öffnen und sich anmelden
3. Schreiben / Lesen von Dateien
4. FTP-Verbindung schließen

Mit ‚send’ oder ‚read’ wird als erstes eine Instanz des Funktionsblocks „FTP_CLIENT“ aus der „WagoLibFtp.lib“ erzeugt. Danach werden oben genannte Schritte durchlaufen. Es wird zuerst die Ziel-IP-Adresse (‚ftpServer_ip_adr’ = ‚HOST_IP’), sowie die Portnummer (‚PORT_NUM’) gesetzt. Danach wird zu diesem Server mit der Action „OPEN“ eine Verbindung hergestellt.

Als nächstes öffnet das Programm über die Action „LOGIN“ einen Account auf dem Server und meldet sich mit ‚USERNAME’ und ‚PASSWORD’ an.

Wenn sich der Controller erfolgreich eingeloggt hat wird das Schreiben bzw. Lesen über die Action „PUT“ bzw. „GET“ gestartet. Der jeweiligen Action wird die Quell- (‚filename_source’ = ‚LPATH’) und Ziel-Datei (‚filename_destination’ = ‚RPATH’) übergeben.

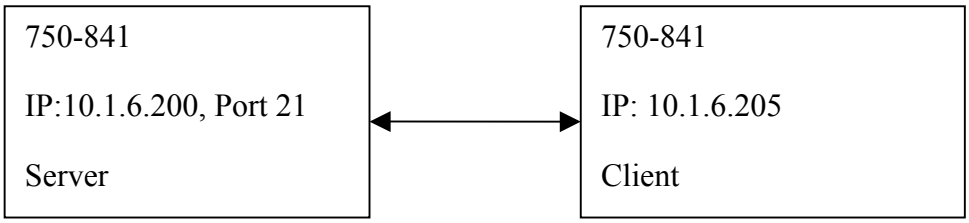
Als letztes wird dann mit „CLOSE“ die Ftp-Verbindung wieder geschlossen.

5 Beispiele

Im folgenden werden Beispiele erläutert, die den Einsatz der SysLibFile.lib Funktionalität verdeutlichen. Die Beispiele greifen dabei auf die Bausteine file_write und file_send_read zurück. Diese Bausteine und Beispiele stehen im Quellcode zur Verfügung und können ggf. eigenständig erweitert und verändert werden.

5.1 Daten in ein File schreiben und verschicken

In dem Beispielprogramm wird ein Bytearray über den Baustein file_write auf dem lokalen Flash gespeichert. Hierzu wird ein Datei-Name angegeben und an den Eingang write eine logische „1“ angelegt. Sobald die Daten erfolgreich gespeichert wurden wird der Ausgang file_ok eingeschaltet.
Als nächstes kann diese Datei dann vom Flash verschickt werden. Dies geschieht über den Baustein file_send_read. Angegeben werden muss noch der Name unter dem das File auf dem Server gespeichert werden soll, sowie die ftp-Server-IP-Adresse (hier 10.1.6.200) als auch der Username und das Passwort.
Wird der Eingang send gesetzt wird das File verschickt. Sobald dies erfolgreich ausgeführt wurde wird der Ausgang file_send eingeschaltet, der wiederum das Löschen des Files auf dem Client auslöst.
Jetzt ist die ursprüngliche Datei nicht mehr lokal, sondern auf dem Server gespeichert.
Über read kann diese Datei wieder zurückgeholt werden.



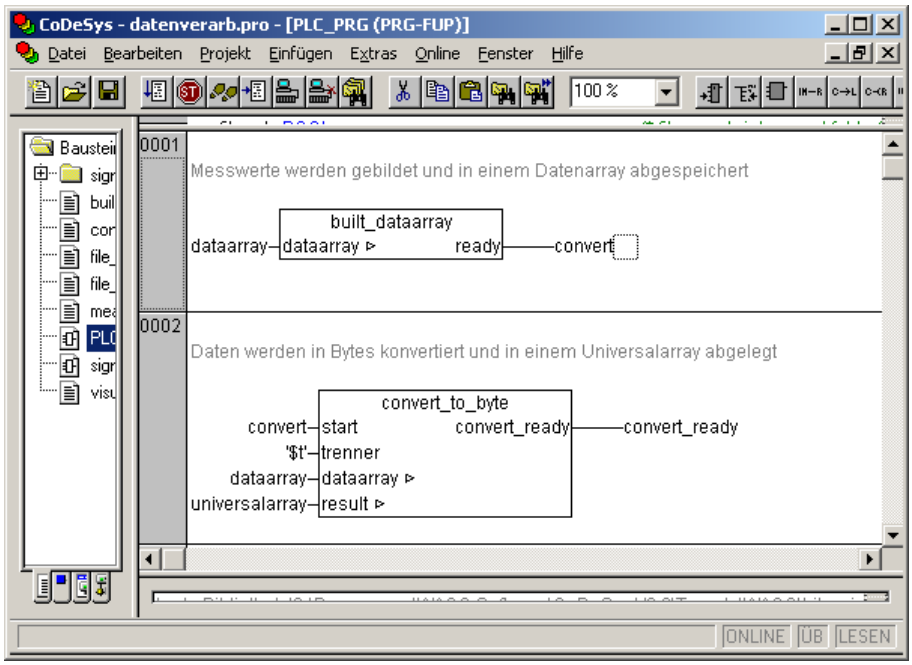
Beispielprogramm: FileWriteSend.pro

5.2 Konvertierung von Daten

Im ersten Beispiel wurde darauf eingegangen wie man Daten in ein File schreibt und dieses dann verschickt.

In diesem weiterführenden Beispiel geht es nun darum nicht ein Feld von Bytes zu haben, sondern ein Feld mit Messwerten (hier in built_dataarray gebildet aus Werten (Int), Zeitstempeln (DT) und Zeiten (Time)).

Da der Baustein file_write ein Array of Bytes erwartet, müssen die Messwerte in Bytes konvertiert werden.



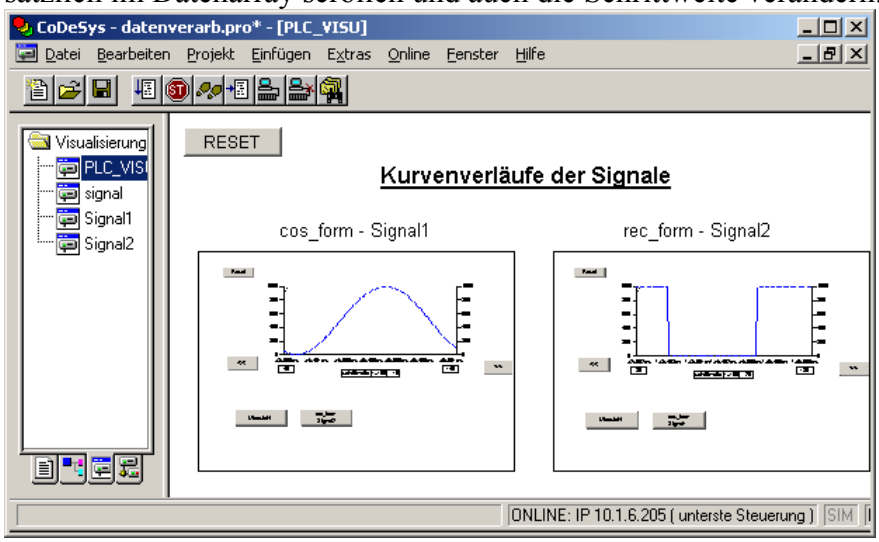
Zur Konvertierung der Messdaten in ein Universalarray wird der Baustein convert_to_byte verwendet. Als Datenübergabebereich sind zwei Datenfelder definiert. Im dataarray sind die ursprünglichen Messwerte gespeichert. In einem zweiten Feld (result) werden dann die konvertierten Daten abgelegt. Die Konvertierung wird über den Eingang start eingeleitet. Über einen weiteren Eingang trenner kann man ein Trennzeichen (hier z.B. Tab) definieren, das die einzelnen Daten von einander trennt. Der boolsche Ausgang convert_ready gibt an wann die Konvertierung abgeschlossen ist und das universalarray zur weiteren Benutzung zur Verfügung steht.

Intern wird zunächst das Bytearray gelöscht. Danach wird ein Pointer auf den Anfang des Bytearrays gesetzt. In einer Schleife wird dann ein Wert ausgelesen, in Bytes konvertiert und der Pointer an die neue Position gesetzt. Anschließend wird ein Trennzeichen eingefügt, der Pointer versetzt und der nächste Wert ausgelesen.

Beispielprogramm: Datenverarbeitung.pro

5.2.1 Visualisierung von Daten

In dem Beispielprojekt ‚Datenverarbeitung.pro' ist außerdem noch eine Visualisierung realisiert. Es werden zwei Signal-Kurvenverläufe (aus dem FB‚measured_value' gebildet über Signalgeneratoren in ‚signals') dargestellt. Es ist möglich zwischen den Verläufen hin- und her zu schalten. Außerdem kann man zusätzlich im Datenarray scrollen und auch die Schrittweite verändern.

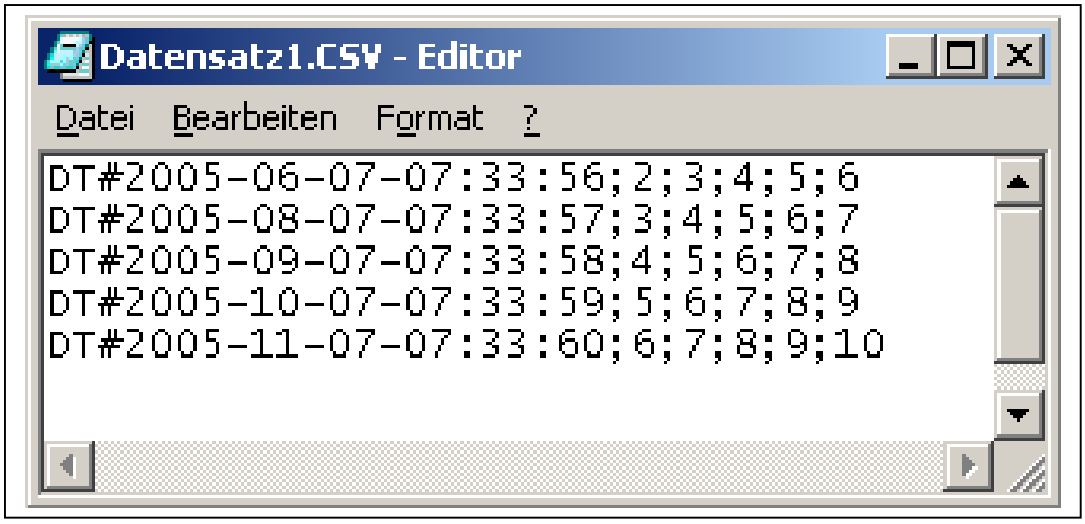


Die einzelnen Kurvenverläufe sind Referenzen einer Hauptvisualisierung (‚signal'). So können die Variablen aus den Funktionsblöcken über Platzhalter beobachtet werden.

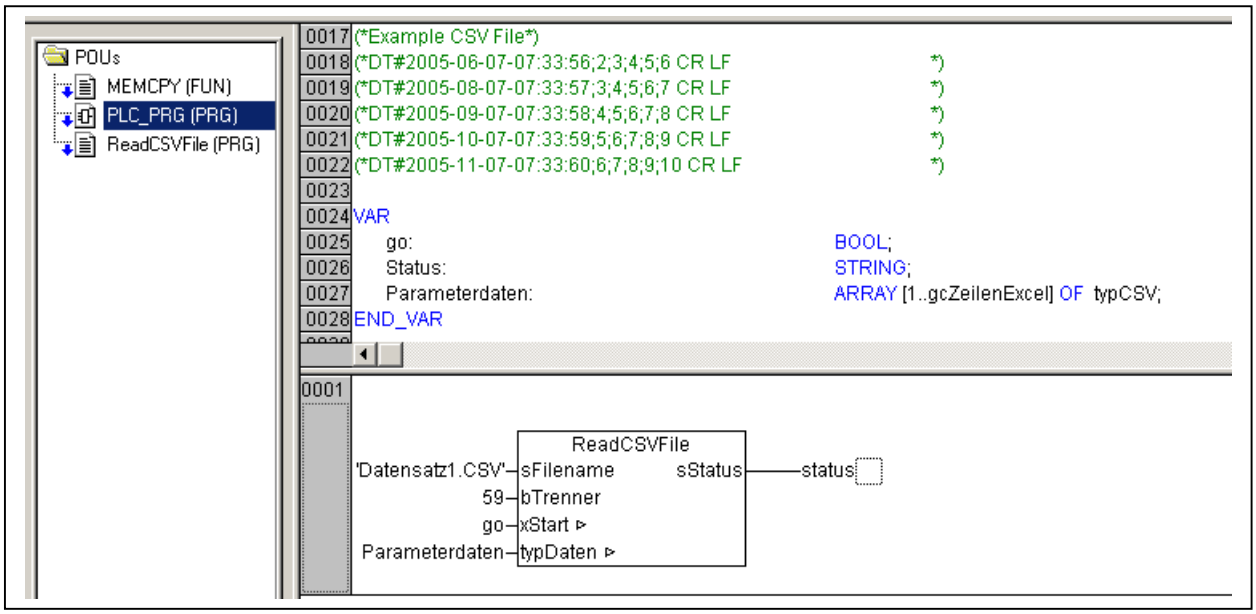
Beispielprogramm: Datenverarbeitung.pro

5.2.2 CSV Datei einlesen

Im folgenden Beispielprojekt wird eine CSV Datei eingelesen. Die Daten werden in einem Array vom Typ typCSV abgelegt.



Der Datensatz setzt sich dabei aus einem Datum und 5 Parameterwerten zusammen. Die einzelnen Werte werden durch ein Semikolon getrennt. Jeder Datensatz wird durch ein CR LF abgeschlossen.



Über die Variable xStart wird der Einlesevorgang aktiviert.

bTrenner ist der ASCII Wert des Trennzeichens, z.B.59 für das Semikolon

In den globalen Variablen wird der Aufbau der Datei parametriert.

```
gc_RawDataSize:UINT:=500;
gc_ParameterCount:INT:=5;
gcZeilenExcel:INT:=5;
```

gc_RawDataSize definiert die maximale Dateigröße in Bytes.

gc_ParameterCount legt die Anzahl der Parameterwerte fest.

gcZeilenExcel legt die Anzahl der Datensätze fest.

Beispielprogramm: CSV_Datei_Lesen.pro, Datensatz1.csv

WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D-32385 Minden
Hansastraße 27 • D-32423 Minden
Telefon: 05 71/8 87 – 0
Telefax: 05 71/8 87 – 1 69
E-Mail: info@wago.com

Internet: http://www.wago.com