

# SIMOTION

## Frequently asked Questions

Wie kann man an Antrieben mit projektierten  
Safety Integrated Extended Functions Hardware-  
Komponenten ohne PG tauschen?

**SIEMENS**

Technische Änderungen des Produktes vorbehalten.

## **Copyright**

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhaltes sind nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte vorbehalten, insbesondere für den Fall der Patenterteilung oder GM-Eintragung.

## Allgemeine Hinweise

### Hinweis

Die Applikationsbeispiele sind unverbindlich und erheben keinen Anspruch auf Vollständigkeit hinsichtlich Konfiguration und Ausstattung sowie jeglicher Eventualitäten. Die Applikationsbeispiele stellen keine kundenspezifische Lösungen dar, sondern sollen lediglich Hilfestellung bieten bei typischen Aufgabenstellungen. Sie sind für den sachgemäßen Betrieb der beschriebenen Produkte selbst verantwortlich. Diese Applikationsbeispiele entheben Sie nicht der Verpflichtung zu sicherem Umgang bei Anwendung, Installation, Betrieb und Wartung. Durch Nutzung dieser Applikationsbeispiele erkennen Sie an, dass Siemens über die beschriebene Haftungsregelung hinaus nicht für etwaige Schäden haftbar gemacht werden kann. Wir behalten uns das Recht vor, Änderungen an diesen Applikationsbeispielen jederzeit ohne Ankündigung durchzuführen. Bei Abweichungen zwischen den Vorschlägen in diesen Applikationsbeispielen und anderen Siemens Publikationen, wie z.B. Katalogen, hat der Inhalt der anderen Dokumentation Vorrang.

### Gewährleistung, Haftung und Support

Für die in diesem Dokument enthaltenen Informationen übernehmen wir keine Gewähr.

Unsere Haftung, gleich aus welchem Rechtsgrund, für durch die Verwendung der in diesem Applikationsbeispiel beschriebenen Beispiele, Hinweise, Programme, Projektierungs- und Leistungsdaten usw. verursachte Schäden ist ausgeschlossen, soweit nicht z.B. nach dem Produkthaftungsgesetz in Fällen des Vorsatzes, der groben Fahrlässigkeit, wegen der Verletzung des Lebens, des Körpers oder der Gesundheit, wegen einer Übernahme der Garantie für die Beschaffenheit einer Sache, wegen des arglistigen Verschweigens eines Mangels oder wegen Verletzung wesentlicher Vertragspflichten zwingend gehaftet wird. Der Schadensersatz wegen Verletzung wesentlicher Vertragspflichten ist jedoch auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht Vorsatz oder grobe Fahrlässigkeit vorliegt oder wegen der Verletzung des Lebens, des Körpers oder der Gesundheit zwingend gehaftet wird. Eine Änderung der Beweislast zu Ihrem Nachteil ist hiermit nicht verbunden.

**Copyright© 2006 Siemens A&D. Weitergabe oder Vervielfältigung dieser Applikationsbeispiele oder Auszüge daraus sind nicht gestattet, soweit nicht ausdrücklich von Siemens A&D zugestanden.**

Bei Fragen zu diesem Beitrag wenden Sie sich bitte über folgende E-Mail-Adresse an uns:

<mailto:applications.erlf@siemens.com>

### Qualifiziertes Personal

im Sinne der Dokumentation sind Personen, die mit Aufstellung, Montage, Inbetriebsetzung, Betrieb und Instandhaltung der einzusetzenden Produkte vertraut sind und über die ihrer Tätigkeit entsprechenden Qualifikationen verfügen

z. B.:

- Ausbildung oder Unterweisung bzw. Berechtigung, Stromkreise und Geräte gemäß den Standards der Sicherheitstechnik ein- und auszuschalten, zu erden und zu kennzeichnen.

- Ausbildung oder Unterweisung gemäß den Standards der Sicherheitstechnik in Pflege und Gebrauch angemessener Sicherheitsausrüstung.
- Schulung in Erster Hilfe.

Warnhinweise werden in dieser Dokumentation explizit nicht gegeben. Es wird jedoch ausdrücklich auf die Warnhinweise der Betriebsanleitung für das jeweilige Produkt verwiesen.

## Hinweis auf Exportkennzeichen

AL: N

ECCN: N



## Inhaltsverzeichnis

1	Fragestellung.....	7
2	Lösung.....	8
2.1	Projektstruktur und Hinweise auf Besonderheiten bei der Konfiguration .....	8
2.2	Applikations / Programmstruktur .....	9
2.3	Beschreibung der Programmteile.....	10
3	Detailbeschreibung der Auftragsverwaltung der DPV1-Dienste .....	13
3.1.1	Grundlagen.....	13
3.1.2	Auftragsverwaltung mit den Funktionsbausteinen der LDPV1.....	13
3.2	Antriebsfehlerauswertung .....	17
3.3	Quittierung HW-Tausch.....	22
4	Änderungen.....	25
5	Ansprechpartner.....	26

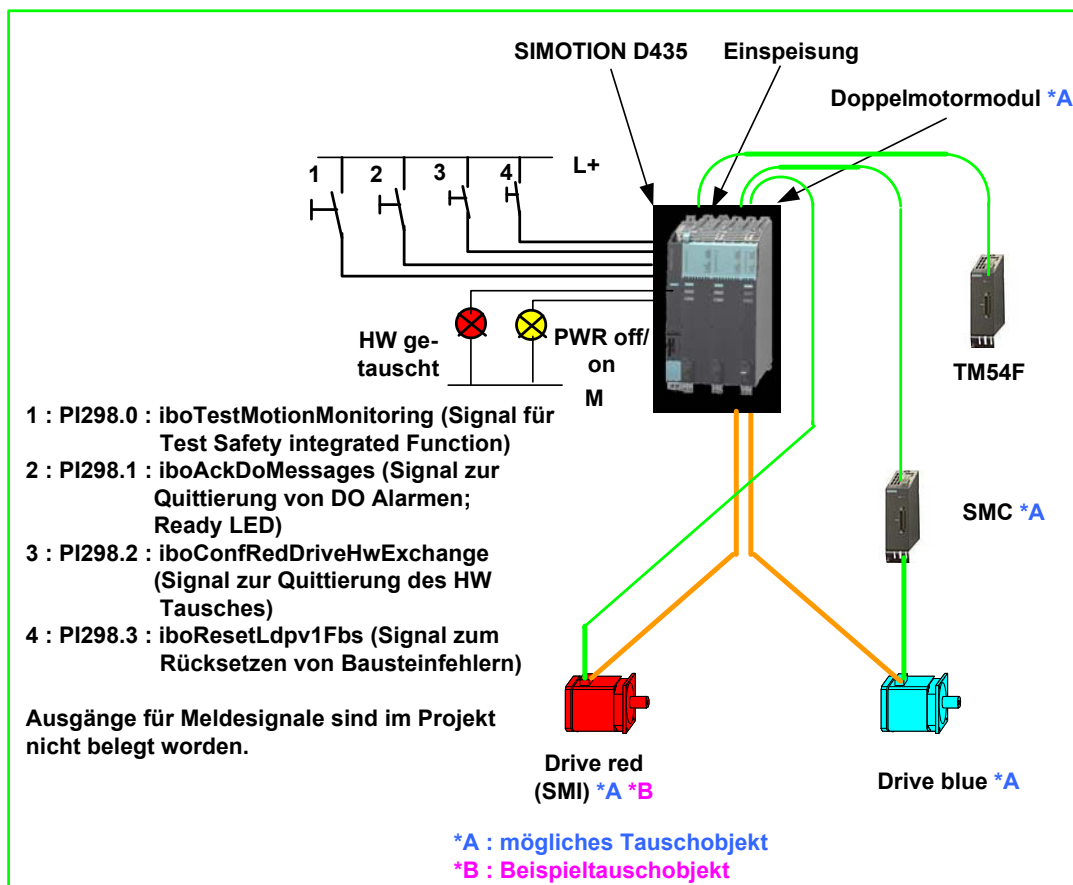
## 1 Fragestellung

Nach dem Tausch einer Komponente an einem Antrieb, in dem die drivebased safety integrated Extended Functions verwendet werden, ist eine Aktualisierung der Hardware-Checksumme notwendig.

**Wie ist es möglich nach einem Komponententausch, an einem Antrieb der die drivebased safety integrated Extended Functions verwendet, die geforderte Quittierung ohne PG zu durchzuführen?**

### Hardwarestrukturbeispiel / Topologie

Die Hardwarestruktur setzt auf den 2-Achs SIMOTION Standard Testkoffer von SIEMENS auf. Zur Nutzung der Safety integrated Extended Functions wurde ein TM54F Modul hinzugefügt. Die erforderlichen Quittier- und Rücksetzsignale werden durch Taster erzeugt, die an die SIMOTION Onboard I/O's angeschlossen wurden. Hier könnten auch, wie im Bild unten dargestellt Meldelampen zur Signalisierung bestimmter Zustände angeschlossen werden.



## 2 Lösung

Wird im Servicefall eine Hardwarekomponente getauscht, ohne den Einsatz von PG/PC und Engineeringsoftware SIMOTION SCOUT so muss die erforderlich Quittierung und ggf. auch Meldungen auf einem anderen Weg erfolgen. Dies kann z. B. über ein angeschlossenes HMI Panel oder aber über Hardwaresignale (Taster, Lampen) wie im nachfolgenden Beispiel gezeigt, erfolgen.

Zur Realisierung werden Bibliotheksfunktionen der Standardbibliothek LDPV1 verwendet. Diese Bibliothek ist Bestandteil der SIMOTION Utilities & Applications und stellt Funktionen zur Verfügung, welche typische Koordinierungsaufgaben im Zusammenhand mit azyklischer Kommunikation übernehmen.

Die SIMOTION Utilities & Applications sind im Lieferumfang von SIMOTION SCOUT enthalten.

Im nachfolgenden Beispiel wird gezeigt wie diese aufzurufen und zu parametrieren sind und welche applikativen Ergänzungen noch erforderlich sind.

### 2.1 Projektstruktur und Hinweise auf Besonderheiten bei der Konfiguration

Hinsichtlich der Kommunikation zwischen den Antrieben und SIMOTION ist zu beachten, dass zusätzlich zu den Telegrammen 105 in den Istwerttelegrammen der 3 Wort lange Safety Datenblock als Telegrammverlängerung zu konfigurieren ist. (detaillierte Beschreibung siehe SIMOTION Funktions-handbuch „Motion Control TO Achse elektrisch/hydraulisch, Externer Geber“ Kap. 2.26.1).

Copyright © Siemens AG 2006 All rights reserved

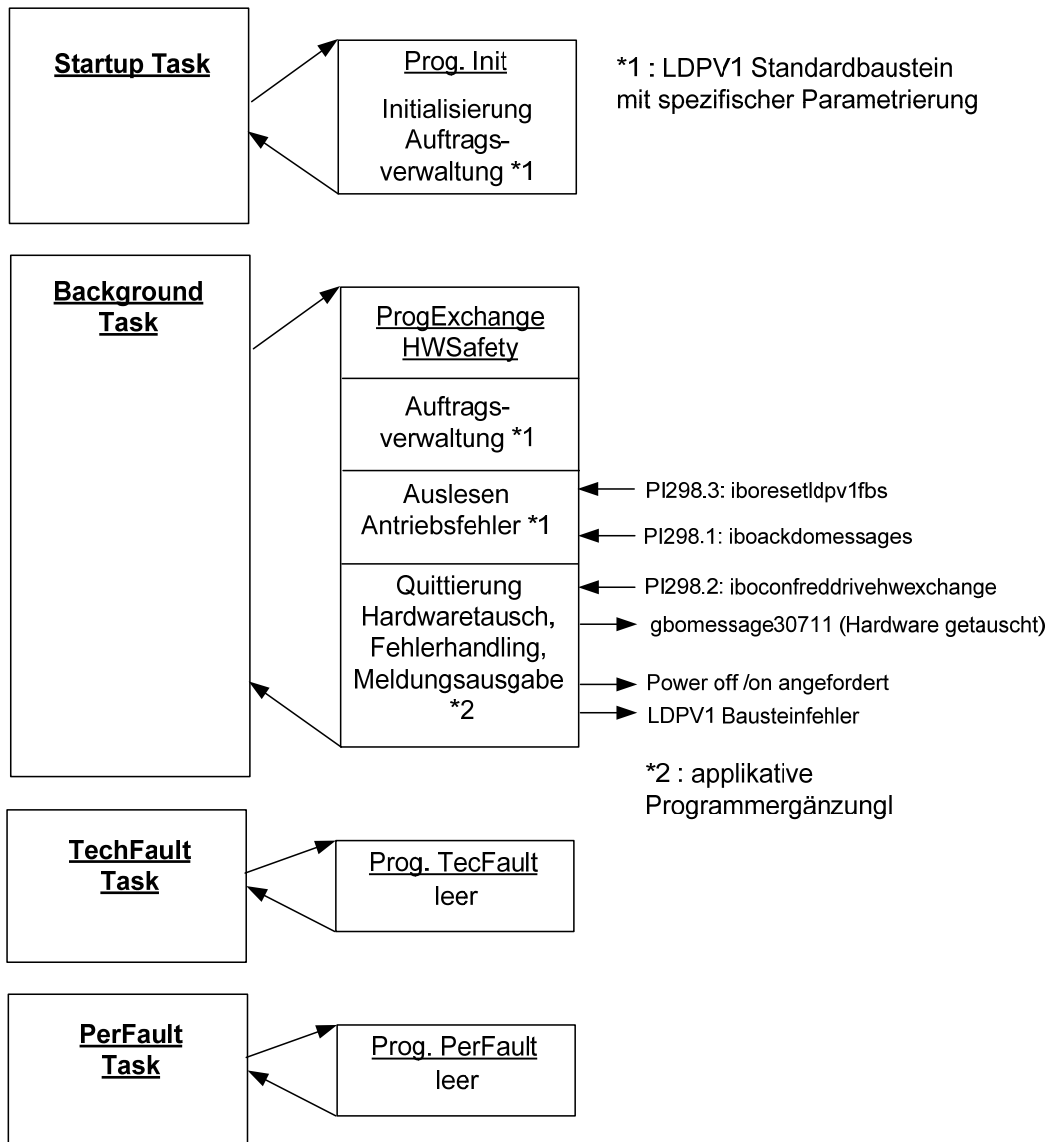
The screenshot shows the SIMOTION SCOUT interface with a project tree on the left and a configuration table for PROFdrive-Telegramm on the right. The table lists telegram configurations for various drive objects.

Objekt	Antriebsobjekt	-Nr.	Telegrammtyp	Eingangsdaten		Ausgangsdaten		SIMOTION Achse
				Länge	Adresse	Länge	Adresse	
1	Drive_blue	4	SIEMENS Telegramm 105, PZD-10/10	10	300..319	10	276..295	Axis_blue
2	Drive_Red	3	SIEMENS Telegramm 105, PZD-10/10	10	296..275	10	296..275	Axis_red
3	Control_Unit	1	SIEMENS Telegramm 380, PZD-2/2	3	276..281	0	---	---
4	TB30	2	Freie Telegrammprojektion mit BICO	2	296..299	2	296..299	---
ohne PZDs (kein zyklischer Datenaustausch)								



## 2.2 Applikations / Programmstruktur

Das Beispielprojekt besteht im Wesentlichen aus LDPV1 Bibliotheksfunktionen und einer kurzen applikativen Ergänzung welche Meldungsausgabe, Quittierung des Hardwaretausches und ein kurzes Fehlerhandling umfasst.



## 2.3 Beschreibung der Programmteile

Das Beispielprogramm ist in den 3 ST Unit's

- dData
- pHWExchange
- pFaultHandl

realisiert

### Unit : dData: globale Konstanten und Variablen.

```
//----- Device Global Constants -----
VAR_GLOBAL CONSTANT
    BUFFER_IDENTIFIER      : INT := 0; //bufferidentifier for SI
    RED_DRIVE_LOG_ADDRESS  : INT := 256;
END_VAR

//----- Device Global Type Definitions -----
TYPE
END_TYPE

//----- Device Global Variables -----
VAR_GLOBAL
    gboMessage30711 : BOOL;
END_VAR
-----
```

### Unit : pHWExchange: enthält die Programme „ProgInIt“ und „ProgExchangeHWSafety“.

„ProgInIt“ : Initialisierung der LDPV1 basierten Auftragsverwaltung.

```
PROGRAM ProgInIt
VAR

END_VAR

//call the function for initialisation of buffer management from LDPV1
sRetsldpvlcombufferinit :=
    fclldpvlcombufferinit(bufferidentifier := BUFFER_IDENTIFIER);

END_PROGRAM
-----
```

**ProgExchangeHWSafety:**

- LDPV1 Auftragsbufferverwaltung
- Detektieren ob Antriebsfehler vorliegen (Abfrage Bit 15 aus Safety Zustandswort)
- Antriebsalarme auslesen
- Prüfen ob Hardwaretausch vorliegt (Safety messages 30711 und 1031 nach Tausch eines Sensor Modules oder Safety message 35150 nach Tausch eines MotorModules liegen vor)
- Hardwaretausch quittieren
- Fehlerhandling

```
PROGRAM ProgExchangeHWSafety
```

```
//call the function for buffer management from LDPV1
sRetsldpv1combuffermanage := fclldpv1combuffermanage();

//check the signal for safety messages (bit 15)
boSafetyessageavailable := _getBit (
    in := axis_red.drivedata.drivesafetyextendedfunctionsinfodata.state,
    n := 15);

// read drive alarms
fbldpv1getRedDriveAlarms(
    enable           := NOT (iboresetLDPV1fbs),
    reset           := iboresetLDPV1fbs,
    bufferidentifier := BUFFER_IDENTIFIER,
    ack             := iboackdomessages,
    filteredalarminfo := TRUE,
    typeofdo       := DO_WITH_TO,
    axis            := Axis_red,
    restartdelay    := T#100ms,
    checksimessages := BY_EXTERNAL_FLAG,
                    // safety messages occurred !
    meessagesactiveexternal := boSafetyessageavailable);

// Check safety messages for 35150 OR 30711 / 1031
IF fbldpv1getreddrivealarms.faultsandalarms.bosimessageactive
THEN
    FOR i16Counter := 0 TO LDPV1_MAX_NO_OF_INFORMATION DO
        IF
fbldpv1getreddrivealarms.faultsandalarms.aul6messagecode[i16Counter] = 35150
OR
(fbldpv1getreddrivealarms.faultsandalarms.aul6messagecode[i16Counter] =
30711 AND
fbldpv1getreddrivealarms.faultsandalarms.ai32messageinfo[i16Counter] = 1031)
            gboMessage30711 := TRUE; // set information bit for // user ! e.g
create a message "If hardware was exchanged, confirm this with //button... "
            END_IF;
        END_FOR;
```

```
END_IF;

// Acknowledge Hardware exchange
fbldpv1RedDriveconfHwExchange (
    execute      := (gboMessage30711 AND iboconfreddrivehwexchange)
                  AND NOT (iboresetLDPV1fbs) ,
    reset        := iboresetLDPV1fbs,
    bufferidentifier := BUFFER_IDENTIFIER,
    logaddress    := RED_DRIVE_LOG_ADDRESS,
    copyramtorom  := TRUE,
    restartdevice := FALSE);

// Error handling
IF fbldpv1RedDriveconfHwExchange.error
THEN
    ; //e.g create a error message

    // Reset information bit / create message for user that pwr off/on
    // is required
ELSIF fbldpv1RedDriveconfHwExchange.done
THEN
    //e.g create a message "hardware exchanged has been confirmed. Please
    //switch power off/on"
    gboMessage30711 := FALSE;
END_IF;
END_PROGRAM
-----
```

### Unit : ProgFaultHandl:

Diese Unit enthält 2 leere Programme zum Aufruf in der TechFaultTast bzw. PeripherieFaultTask.

Diese Tasks können vom Anwender nach eigenen Bedarf ausprogrammiert werden.

## 3 Detailbeschreibung der Auftragsverwaltung der DPV1-Dienste

### 3.1.1 Grundlagen

Nach PROFIDRIVE-Profil ist für PROFIDRIVE-konforme Antriebe nur ein DPV1-Auftrag pro Antriebsgerät zulässig. Werden zwei oder mehrere DPV1-Aufträge gleichzeitig an ein Antriebsgerät abgesetzt, kann es zu Konflikten bei der Auftragsbearbeitung im Antriebsgerät kommen, die dazu führen, dass sich die unterschiedlichen DPV1-Aufträge gegenseitig stören.

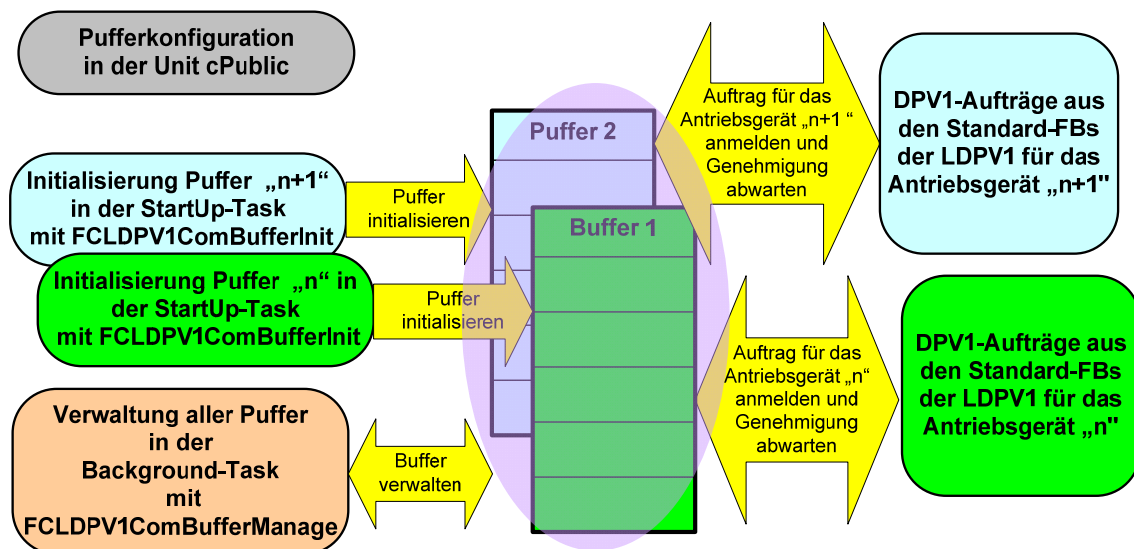
### 3.1.2 Auftragsverwaltung mit den Funktionsbausteinen der LDPV1

Zur Überprüfung, ob nicht bereits ein Auftrag auf dem gewünschten Antriebsobjekt aktiv ist, bietet die LDPV1 eine Auftragsverwaltung an.

Die Einführung dieser Auftragsverwaltung setzt aber auch die Bedingung voraus, dass jede Funktionalität, die DPV1-Dienste verwendet, sie auch in Anspruch nimmt.

Grundprinzip der Auftragsverwaltung ist die Bereitstellung und Verwaltung eines (oder mehrerer) Auftragspuffer, in den sich jede Applikation, die DPV1-Dienste nutzt, einträgt.

#### Aufbau der LDPV1-Auftragsverwaltung



### Konfiguration der Auftragspuffer der LDPV1-Auftragsverwaltung

Die Konfiguration der Auftragspuffer für die Auftragsverwaltung erfolgt in der Unit cPublic der LDPV1-Bibliothek.

```

32 LDPV1_MAX_DO_PER_STATION      : UINT := 32; // maximal number of Drive Objects per s
33 LDPV1_MAX_DO_PER_CU          : UINT := 12; // maximal number of Drive Objects per C
34 LDPV1_MAX_NUMBER_OF_CX32     : UINT := 2;  // maximal number of CX32 (only Startup)
35 LDPV1_MAX_NUMBER_OF_PN_DEVICES : UINT := 10; // maximal number of profinet devices (D
36 LDPV1_MAX_NUMBER_OF_DEVICES_AT_DP : UINT := 10; // maximal number of devices at external
37
38 //----- constants for buffer management -----//
39 LDPV1_MAX_NUMBER_OF_BUFFER_REQUESTS : INT := 30; // maximal numbers of buffer requests
40 LDPV1_MAX_BUFFER_PER_STATION      : INT := 1; // maximal numbers of buffers per station
41 // maximum time of semaphore assignment.
42 // If semaphore assignment wasn't successful within
43 // user-defined time in constant MAX_ALLOCATION_TIME_SERVICE,
44 // then an error will be returned by FC_ComBufferManage.
45 LDPV1_MAX_ALLOCATION_TIME_SERVICE : TIME := T#40s;
46
47 //----- constants for Startup -----//
48 // gtMAX_TIME_STARTUP must be greater than (number of projected CX32) * 90s //
49 LDPV1_MAX_TIME_STARTUP           : TIME := T#5m; // max. Time for Startup System
  
```

*LDPV1\_MAX\_NUMBER\_OF\_BUFFER\_REQUESTS*: Größe der Auftragspuffer

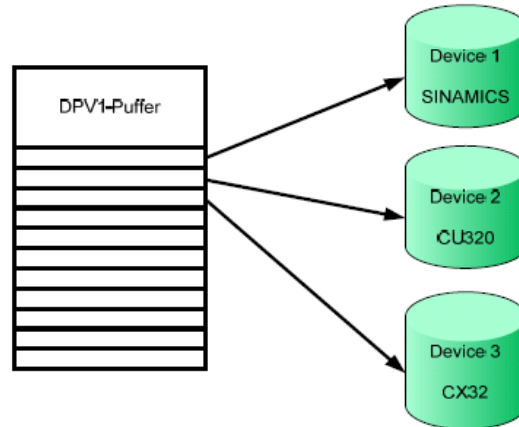
*LDPV1\_MAX\_BUFFER\_PER\_STATION* : Anzahl der Auftragspuffer in der Steuerung

### Die Verwendung der Auftragspuffer ist in 2 Varianten möglich:

- Ein Auftragspuffer für alle Antriebsgeräte. (in dem gezeigten Beispiel genutzt)

Diese Möglichkeit nutzt nicht den technisch maximal möglichen Durchsatz an DPV1-Aufträgen, verhindert aber den Fehler, dass ein Antriebsgerät mit mehreren Puffern arbeitet.

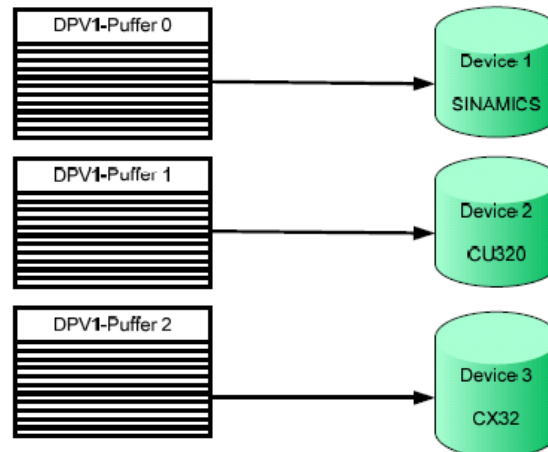
*LDPV1\_MAX\_BUFFER\_PER\_STATION : 1*



- Ein Auftragspuffer für jedes Antriebsgerät.

Diese Variante ermöglicht den technisch maximal möglichen Durchsatz an DPV1-Aufträgen.

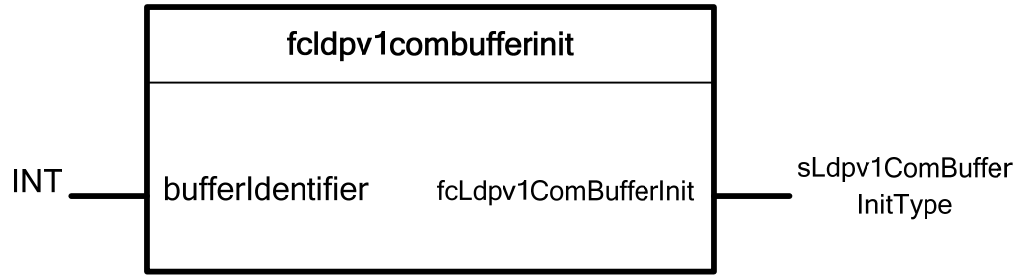
*LDPV1\_MAX\_BUFFER\_PER\_STATION : Anzahl der Antriebsgeräte*



## Initialisierung der LDPV1- Auftragsverwaltung (Startup Task, Programm ProgInit)

Zunächst müssen im Anlauf alle zu verwendenden Puffer mit ***FCLDPV1ComBufferInit*** initialisiert werden

## Schematische KOP – Darstellung



## Parameter fcldpv1combufferinit

Tabelle 3-1: Ein - und Ausgangsparameter

Parameter	Datentyp	P-Typ 1)	P/O Typ 2)	Initial wert	Beschreibung
buffer Identifier	INT	IN	O	2	bufferIdentifier identifiziert den verwendeten Auftragspuffer
fcLdpv1Com BufferInit	sLDPV1Com BufferInitType	OUT	-		Rückgabewert der Funktion zur Fehler-Identifikation und Spezifikation
<sup>1)</sup> Parametertypen: IN = Eingangsparameter, OUT = Ausgangsparameter, <sup>2)</sup> Parameterart: P = Pflichtparameter, O = Optionaler Parameter					

Typ: sLDPV1ComBufferInitType

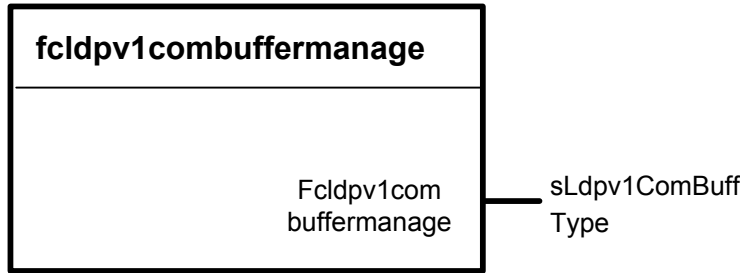
Name	Typ	Bedeutung
fcldpv1combufferinit	sLDPV1ComBufferInitType	
boError	BOOL	Identifiziert, ob ein Fehler anliegt
b32ErrorID	DWORD	spezifiziert einen aufgetretenen Fehler

## LDPV1- Pufferverwaltung (Background Task, Programm ProgExchangeHWSafety)

Nach erfolgter Initialisierung wird aus einer zyklischen Task (z.B. Background) heraus die Funktion **FCLDPV1ComBufferManage** bei jedem Durchlauf aufgerufen. In dieser Funktion wird entschieden, welcher im Puffer eingetragene Auftrag abgearbeitet werden darf.

## Schematische KOP – Darstellung





## Parameter fcldpv1combuffermanage

Tabelle 3-2: Ausgangsparameter

Parameter	Datentyp	P-Typ 1)	P/O Typ 2)	Initial wert	Beschreibung
FCLDPV1ComBufferManage	sLDPV1ComBufferManageType	OUT	-		Rückgabewert der Funktion zur Fehler-Identifikation und Spezifikation
<sup>1)</sup> Parametertypen: IN = Eingangsparameter, OUT = Ausgangsparameter, <sup>2)</sup> Parameterart: P = Pflichtparameter, O = Optionaler Parameter					

Typ: sLdpv1ComBufferManageType

Name	Typ	Bedeutung
FCLDPV1ComBufferManage	sLdpv1ComBufferManageType	
boError	BOOL	Identifiziert, ob ein Fehler anliegt
b32ErrorID	DWORD	spezifiziert einen aufgetretenen Fehler
i16BufferIdentifier	INT	Identifiziert den aktuell verwendeten Auftragspuffer

## Verwendung der LDPV1- Auftragsverwaltung

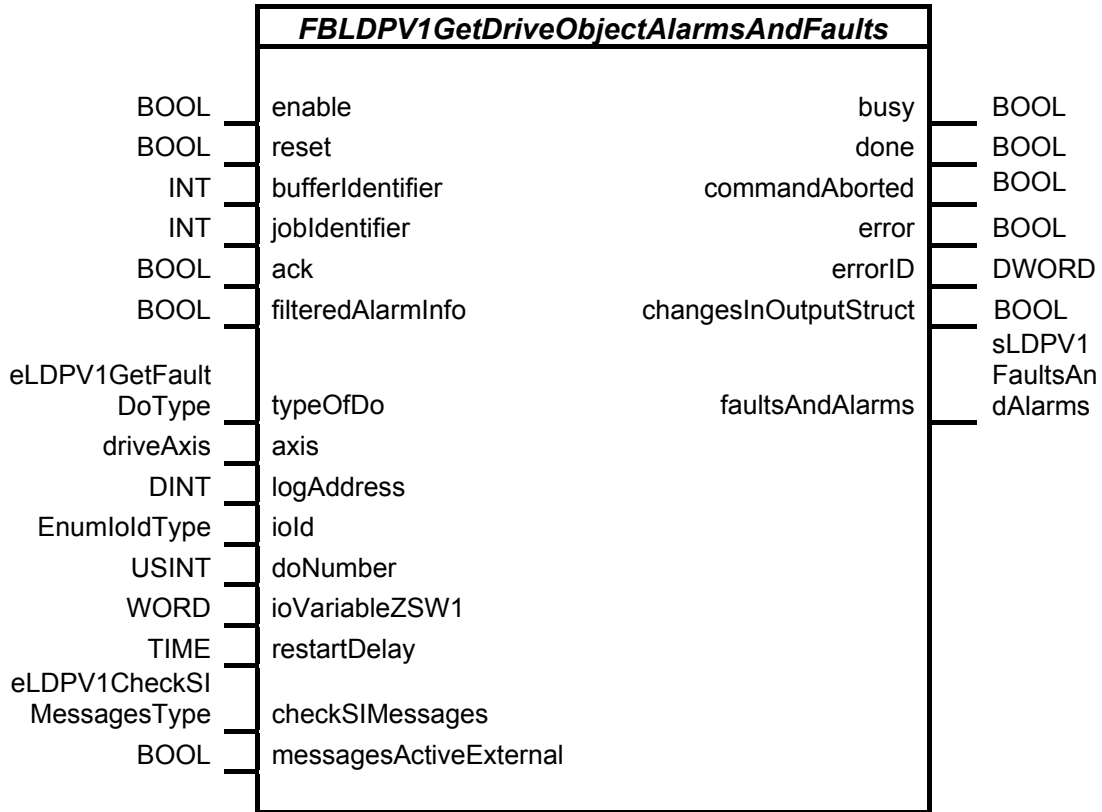
Die LDPV1-Bausteine lassen sich anhand des Parameters „*bufferIdentifier*“ unter die LDPV1-Auftragsverwaltung stellen. Dafür müssen Sie dem verwendeten LDPV1-Baustein die Puffernummer des Antriebsgerätes übergeben. Achten Sie bitte darauf, dass kein Antriebs-gerät mit mehr als einem Auftragspuffer arbeitet.

## 3.2 Antriebsfehlerauswertung

Mit dem Funktionsbaustein **FBLDPV1GetDriveObjectAlarmsAndFaults** (*Instanz im Beispielprogramm : fbldpv1getRedDriveAlarms*) können an einem SINAMICS Driveobjekt Störungen und Warnungen erkannt, die Alarm- bzw. Warnnummern ausgelesen und eine Quittierung ausgelöst werden.

Zusätzlich können durch Aktivierung am Eingang *checkSIMessages* Meldungen gelesen werden, die bei Verwendung der dbSI Extended Functions auftreten.

## Schematische KOP – Darstellung



Copyright © Siemens AG 2006 All rights reserved

Das zu überwachende Driveobjekt kann dem Funktionsbaustein auf 3 unterschiedliche Arten übergeben werden. Hier wird nur die Adressierung über das Technologieobjekt erklärt.

Dabei wird das Driveobjekt über den Eingangsparameter *axis* zugewiesen.

Belegung weiterer Eingangsparameter:

<i>axis</i>	:= TO-Achse
<i>typeOfDo</i>	:= DO_WITH_TO
<i>ioVariableZSW1</i>	:= < nicht relevant >
<i>logAddress</i>	:= < nicht relevant >
<i>doNumber</i>	:= < nicht relevant >

Die Antriebsmeldung „30711 SI Motion MM: Defekt in einem Überwachungskanal“ mit der Zusatzinformation „1031: Fehlerhafte Datenübertragung zwischen Überwachungskanal und Sensor Module“ weist auf einen möglichen

Komponententausch an einem Safety-Antrieb hin. In diesem Fall ist es sinnvoll eine weitere Anwendermeldung zu generieren, die darauf hinweist, dass falls tatsächlich eine HW-Komponente getauscht wurde, eine Bestätigung durchgeführt werden soll.

### Parameter FBLDPV1GetDriveObjectAlarmsAndFaults

Tabelle 3-3: Ein - und Ausgangsparameter

Element	P type <sub>1)</sub>	Datentyp	M/O <sub>2)</sub>	Initialwert	Bedeutung
enable	IN	BOOL	M	FALSE	Aktiviert den Baustein
reset	IN	BOOL	O	FALSE	Setzt den Baustein sowie dessen Ausgänge zurück
bufferIdentifier	IN	INT	O	-1	Identifiziert, welcher Auftragspuffer für den internen Gebrauch des DPV1 Dienstes benutzt werden soll. Die Puffer-Verwaltung wird in Unit „fBuffMana“ in der Bibliothek „LDPV1“ realisiert.
jobIdentifier	IN	INT	O	0	Eindeutiger Job-Identifizier für Debugfunktion im Buffer-Management zur Nachverfolgung der DPV1 Dienste
ack	IN	BOOL	O	FALSE	Ist ack gesetzt, versucht der Baustein anstehende Antriebsfehler zu quittieren. Dies geschieht durch schreiben einer „1“ auf Parameter p3981. Der Eingang ack ist flankensensitiv.
filteredAlarmInfo	IN	BOOL	O	FALSE	Ist dieser Wert TRUE, werden nur die aktuell aktiven Warnungen angezeigt. Sonst werden alle 8 Inhalte des Warnpuffers angezeigt. (Also auch nicht mehr aktive Warnungen)
typeOfDo	IN	eLDPV1GetFaultDoType	O	DO_WITH_TO	Gibt an, von welchem Typ das zu überwachende Driveobjekt ist. Folgende Werte sind möglich: DO_WITH_TO DO_WITH_CYCLIC_ZSW1 DO_WITHOUT_CYCLIC_ZSW1
axis	IN	driveAxis	O	TO#NIL	Achsname wie im Projektbaum. Gilt für Driveobjekte mit Technologieobjekt. Falls es kein TO gibt, dann TO#Nil
logAddress	IN	DINT	O	0	Logische Adresse des Driveobjekts Fallunterscheidung je nach <b>typeOfDO</b> 1: DO_WITH_TO keine Bedeutung  2: DO_WITH_CYCLIC_ZSW1 Logische Basisadresse des Driveobjekts  3: DO_WITHOUT_CYCLIC_ZSW1 Beliebige logische Adresse des Gerätes, auf dem sich das Driveobjekt befindet. Kann keine DO -Nummer übergeben werden, muss die logische Basisadresse übergeben werden
lold	IN	enumloldType	O	INPUT	Angabe der Datenrichtung der Basisadresse (Input/Output) 1: DO_WITH_TO keine Bedeutung  2/3: DO_WITH_CYCLIC_ZSW1 und DO_WITHOUT_CYCLIC_ZSW1 Angabe der Datenrichtung der übergebenen

Element	P type <sup>1)</sup>	Datentyp	M/O <sup>2)</sup>	Initialwert	Bedeutung
					logischen Adresse
doNumber	IN	USINT	O	255	DO-Nummer des zu überwachenden Driveobjekts. Fallunterscheidung für die 3 Fälle: 1/2: DO_WITH_TO DO_WITH_CYCLIC_ZSW1 nicht relevant  3: DO_WITHOUT_CYCLIC_ZSW1 Driveobjekt-Nummer des zu überwachenden Driveobjekts ohne zyklisches Zustandswort. Wird das DO durch die Basisadresse des Antriebs in logAddress übergeben, muss die doNumber nicht versorgt bzw. auf den Wert 255 gesetzt werden.
ioVariableZSW1	IN	WORD	O	16#ffff	Eingangsvariable, die im I/O-Bereich auf die Adresse des ZSW1 in der zyklischen Kommunikation gelegt wurde. Gilt für Driveobjekte ohne Technologieobjekt aber mit zyklischer Kommunikation.
restartDelay	IN	TIME	O	T#0s	Wartezeit, die zwischen 2 Überwachungszyklen vergehen muss. Es ist also die Zeit, die zwischen zwei Überwachungsversuchen vergeht.
checkSIMessages	IN	eLDPV1CheckMessagesType	O	DISABLED	Bei DISABLED werden keine SI-Messages ausgelesen. Bei BY_EXTERNAL_FLAG werden die SI-Messages gelesen, wenn das Anstehen von Meldungen am Eingang <b>messagesActiveExternal</b> mit TRUE durch das Anwenderprogramm gemeldet wird. Bei BY_DO_ADDRESS ermittelt der Funktionsbaustein selbstständig durch Lesen von DOx.r2139.5 ob SI-Messages aktiv sind.
messagesActiveExternal	IN	BOOL	O	FALSE	Externer Eingang, der anzeigt, ob SI-Messages aktuell anstehen oder nicht. Dieser Eingang muss vom Anwender versorgt werden, wenn <b>checkSIMessages</b> mit dem Wert BY_EXTERNAL_FLAG versorgt wird. Die SI-Messages werden nur gelesen, wenn dieser Wert auf TRUE steht.
busy	OUT	BOOL	-	FALSE	Auftrag in Bearbeitung
done	OUT	BOOL	-	FALSE	Auftrag erledigt
commandAborted	OUT	BOOL	-	FALSE	Funktion bzw. Auftrag wurde von extern des FB's abgebrochen. Nicht relevant
error	OUT	BOOL	-	FALSE	Fehler im Baustein (z.B.: Fehler bei Befehlsausführung, Befehlsabbruch, Fehler bei der Parameterversorgung)
errorID	OUT	DWORD	-	16#00000000	Fehlerspezifikation
changesInOutputStruct	OUT	BOOL	-	FALSE	Positive Flanke am Ausgang zeigt an, ob sich in der Ausgabestruktur faultsAndAlarms gegenüber dem letzten Durchlauf etwas geändert hat.
faultsAndAlarms	OUT	sLDPV1FaultsAndAlarmsType	-	-	Ausgabestruktur für anstehende Fehler und Warnungen
<sup>1)</sup> Parametertypen:					

Element	P type <sub>1)</sub>	Datentyp	M/O <sub>2)</sub>	Initialwert	Bedeutung
IN = Eingangsparameter, OUT = Ausgangsparameter, IN/OUT = Durchgangsparameter					
2) Parameterart: M = Pflichtparameter, O = Optionaler Parameter					

Tabelle 3-4: Enum für eLDPV1GetFaultDoType

Enum-Bezeichner	Kommentar
DO_WITH_TO	Driveobjekte mit Technologieobjekt. (Hier nur Antriebe/Achsen)
DO_WITH_CYCLIC_ZSW1	Driveobjekte mit zyklischer Kommunikation
DO_WITHOUT_CYCLIC_ZSW1	Driveobjekte ohne zyklische Kommunikation

Tabelle 3-5: Enum für eLDPV1CheckSIMessagesType

Enum-Bezeichner	Kommentar
DISABLED	Das Auslesen der bei Verwendung der dbSI Extended Functions auftretenden Meldung ist deaktiviert.
BY_EXTERNAL_FLAG	Das Auslesen der bei Verwendung der dbSI Extended Functions auftretenden Meldung ist aktiviert. Das Anstehen der Meldungen wird dem Funktionsbaustein über den Eingang messagesActiveExternal durch die Anwendung übergeben.
BY_DO_ADDRESS	Das Auslesen der bei Verwendung der dbSI Extended Functions auftretenden Meldung ist aktiviert. Das Anstehen der Meldungen wird vom Funktionsbaustein durch Auslesen des Parameters DOx.r2139.5 selbstständig übernommen.

Tabelle 3-6: Ausgabestruktur sLDPV1FaultsAndAlarmsType

Kommentar	Parameter	Datentyp	Beschreibung
	boFaultActive	BOOL	Fehler am Objekt aktiv
	i16NumberOfFaults	INT	Anzahl der aktuell aktiven Fehler
	au16FaultCode[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF UINT	Ausgabe der max. 8 aktuellsten Fehlermeldungen aus r945
	ai32FaultInfo[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF DINT	Ausgabe von Zusatzinformationen aus r949
	adtFaultComeDateAndTime [0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF DT	Ausgabe der Systemzeit zum Zeitpunkt „Fehler gekommen“, entweder in Einheit „Zeit seit Einschalten“ oder im Format UTC im Datenformat Date_And_Time r948/2130
	adtFaultGoneDateAndTime [0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF DT	Ausgabe der Systemzeit zum Zeitpunkt „Fehler behoben“, entweder in Einheit „Zeit seit Einschalten“ oder im Format UTC im Datenformat Date_And_Time r2109/2136
	boAlarmActive	BOOL	Warnung am Objekt aktiv
	i16NumberOfAlarms	INT	Anzahl der aktuellen Warnungen (bei filteredAlarmInfo TRUE nur die aktiven Warnungen)

Kommentar	Parameter	Datentyp	Beschreibung
	au16AlarmCode[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF UINT	Ausgabe der max. 8 aktuellsten Alarmmeldungen aus r2122
	ai16AlarmInfo[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF DINT	Ausgabe von Zusatzinformationen aus r2124
	adtAlarmComeDateAndTime[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF DT	Ausgabe der Systemzeit zum Zeitpunkt „Warnung gekommen“, entweder in Einheit „Zeit seit Einschalten“ oder im Format UTC im Datenformat Date_And_Time r2123/2145
	adtAlarmGoneDateAndTime[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF DT	Ausgabe der Systemzeit zum Zeitpunkt „Warnung gegangen“, entweder in Einheit „Zeit seit Einschalten“ oder im Format UTC im Datenformat Date_And_Time r2125/2146
<b>! Relevant bei Safety-! HW-Tausch</b>	boSIMessageActive	BOOL	SI Meldung am Objekt aktiv
	i16NumberOfMessages	INT	Anzahl der aktuell ausgelesenen Meldungen
<b>! Relevant bei Safety-! HW-Tausch</b>	au16MessageCode[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF UINT	Ausgabe der max. 8 aktuellsten Meldungen aus r9747
<b>! Relevant bei Safety-! HW-Tausch</b>	ai32MessageInfo[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF DINT	Ausgabe von Zusatzinformationen aus r9749

	ar32MessageInfo[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF REAL	Ausgabe von Zusatzinformationen aus r9753 im Format Float
	adtMessageComeDateAndTime[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF DT	Ausgabe der Systemzeit zum Zeitpunkt „Meldung gekommen“, entweder in Einheit „Zeit seit Einschalten“ oder im Format UTC im Datenformat Date_And_Time r9754/9748
	adtMessageGoneDateAndTime[0..LDPV1_MAX_NO_OF_INFORMATION]	ARRAY OF DT	Ausgabe der Systemzeit zum Zeitpunkt „Meldung gegangen“, entweder in Einheit „Zeit seit Einschalten“ oder im Format UTC im Datenformat Date_And_Time r9756/9755

### 3.3 Quittierung HW-Tausch

Mit dem Funktionsbaustein

**FBLDPV1ConfirmationOfComponentExchangeWithSafety** wird der Tausch einer Hardware-Komponente mit aktivierten drivebased safety integrated Extended Functions quittiert.

Die Antriebsmeldung „30711 SI Motion MM: Defekt in einem Überwachungskanal“ mit der Zusatzinformation „1031: Fehlerhafte Datenübertragung zwischen Überwachungskanal und Sensor Module“ weist auf einen möglichen Komponententausch an einem Safety-Antrieb hin.

Im Beispielprogramm wird zur Anzeige der Meldung die Boolesche Variable „gboMessage30711“ auf TRUE gesetzt. Hiermit könnte zusätzlich wie in der Hardwarestruktur auf S. 6 angedeutet über einen Digitalausgang eine Meldelampe „HW wurde getauscht“ angesteuert werden, die einen Komponententausch signalisiert. Alternativ könnte eine Meldung über ein HMI System generiert werden.

Zur Quittierung des Hardwaretausches wird im Beispielprogramm die Inputvariable „iboconfreddrivehwexchange“ verwendet. An diesen Digitaleingang könnte ein Quittiertaster angeschlossen werden („Quit. HW Tausch“).

Ein weiterer Taster am Digitaleingang „iboresetLDPV1fbs“ soll im Fehlerfall dazu dienen anstehende Bausteinfehler rücksetzen zu können.

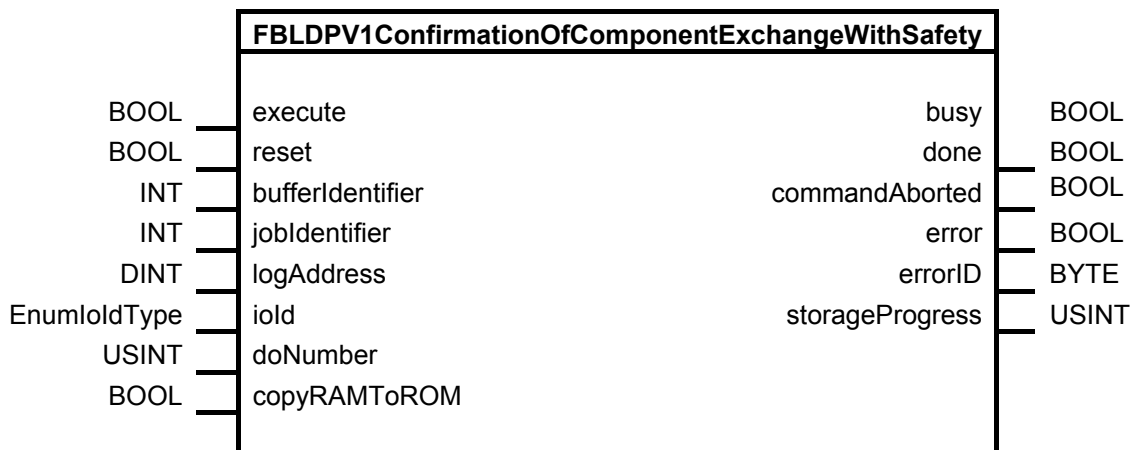
Nach dem erfolgreichen Ablauf des Funktionsbausteins müssen vom Anwender noch folgende Handlung durchgeführt werden:

- POWER ON bei allen Komponenten durchführen.

Im Beispielprogramm könnte zur Anzeige das ein PWR off/on erforderlich ist eine weitere Meldelampe „PWR off/on“ angesteuert werden (mit der fallenden Flanke von „gboMessage30711“).

Vor dem erneuten Betreten des Gefahrenbereichs und vor der Wiederaufnahme des Betriebs müssen alle durch den Komponententausch betroffenen Antriebe durch einmaliges Anwählen der Funktion STO und durch kurzes Verfahren in beiden Richtungen (+/-) bei aktivierter Sicherheitsüberwachungsfunktion (SLS, falls parametrierbar) auf korrekte Funktion überprüft werden.

### Schematische KOP – Darstellung



### Parameter FBLDPV1ConfirmationOfComponentExchangeWithSafety

Tabelle 3-7: Ein - und Ausgangsparameter

Element	P type <sup>1)</sup>	Datentyp	M/O <sup>2)</sup>	Initialwert	Bedeutung
execute	IN	BOOL	M	FALSE	Startet die Bausteinfunktionalität
reset	IN	BOOL	O	FALSE	Setzt den Baustein sowie dessen Ausgänge zurück
bufferIdentifier	IN	INT	O	-1	Indiziert, welcher Auftragspuffer für den internen Gebrauch des DPV1 Dienstes benutzt werden soll. Die Puffer Verwaltung wird in Unit „fBuffMana“ in der Bibliothek

Element	P type <sup>1)</sup>	Datentyp	M/O <sup>2)</sup>	Initialwert	Bedeutung
					„LDPV1“ realisiert.
jobIdentifier	IN	INT	O	0	Eindeutiger Job-Identifizierer für Debugfunktion in Buffer-Management zur Nachverfolgung der DPV1 Dienste. Siehe Dokumentation LDPV1_BufferMana zur Funktion <b>FCLDPV1ComBufferDiag</b>
logAddress	IN	DINT	O	0	Eine beliebige logische Adresse der SINAMICS Komponente. Alternativ kann auch die Diagnoseadresse angegeben werden
iold	IN	eloldType	O	INPUT	Angabe der Datenrichtung der logischen Adresse (Input/Output)
doNumber	IN	USINT	O	255	DO-Nummer des Driveobjekts, auf dem der Baugruppentausch quittiert werden soll. Wird das DO durch die Basisadresse des Antriebs in logAddress übergeben, muss die <b>doNumber</b> nicht versorgt werden
copyRAMtoROM	IN	BOOL	O	FALSE	Mit copyRAMtoROM = TRUE wird abschließend noch ein Kopieren von RAM nach ROM für das Gerät, auf dem sich das Driveobjekt befindet, angestoßen. Der Baustein meldet fertig, wenn RAM nach ROM beendet ist.
busy	OUT	BOOL	-	FALSE	Auftrag in Bearbeitung
done	OUT	BOOL	-	FALSE	Überprüfung/ Quittierungsversuch abgeschlossen
commandAborted	OUT	BOOL	-	FALSE	Funktion bzw. Auftrag wurde von extern des FB's abgebrochen. Nicht relevant.
error	OUT	BOOL	-	FALSE	Fehler im Baustein (z.B.: Fehler bei Befehlsausführung, Befehlsabbruch, Fehler bei der Parameterversorgung)
errorID	OUT	DWORD	-	16#00000000	Fehlerspezifikation
storageProgress	OUT	USINT	-	0	Zyklische Ausgabe des Speicherfortschritts von 0 bis 100% bei Ausführung von RAM nach ROM.
<sup>1)</sup> Parametertypen: IN = Eingangsparameter, OUT = Ausgangsparameter, IN/OUT = Durchgangsparameter <sup>2)</sup> Parameterart: M = Pflichtparameter, O = Optionaler Parameter					

Tabelle 3-8: Fehlermeldungen

Fehlernummer [HEX]	Bedeutung
0	fehlerfrei
1001	Es wurde ein neuer execute ausgelöst, obwohl reset aktiv ist



Fehlernummer [HEX]	Bedeutung
4110	Interner Fehler: Der interne Schritt Merker des FB's hat einen ungültigen Wert.
4111	Ein oder mehrere Parameter nicht lesbar. Für genauere Analyse siehe sRetReadDriveParameter.parameterResult[] innerhalb der Bausteininstanz und Systemhilfe zu _readDriveMultiParameter.
4112	Angesprochene Control-Unit ist ausgelastet. Timeout bei Leseaufträgen.
4113	Ein oder mehrere Parameter nicht schreibbar. Für genauere Analyse siehe sRetWriteDriveParameter.parameterResult[] innerhalb der Bausteininstanz und Systemhilfe zu _writeDriveMultiParameter.
4114	Angesprochene Control-Unit ist ausgelastet. Timeout bei Schreibaufträgen.
4115	Die an den Funktionsbaustein übergebene logische Adresse ist fehlerhaft. Kontrolle der logischen Adresse z. B. in HW-Konfig
4116	Angesprochene Baugruppe meldet Stationsausfall. Kontrolle der Baugruppe erforderlich
4117	Die über die logische Adresse angesprochene Baugruppe ist keine SINAMICS Baugruppe.
4380	Fehler bei Vorbereitung Hardware-Reset. CU.p0972 kann nicht auf Wert = 2 geschrieben werden.

## 4 Änderungen

Tabelle 4-1: Änderungen/Verfasser

Version	Datum/Änderung
1.0	05.03.2009 Ersterstellung
1.1	23.04.2009 Überarbeitung
1.2	21.10.2010 Erweiterung für Motormodul-Tausch

## 5 Ansprechpartner

*Applikationszentrum*

---

SIEMENS

Siemens AG  
Drive Technology  
I DT MC PM APC  
Frauenauracher Str. 80  
Erlangen  
Fax: 09131-98-1297  
mailto: [applications.erlf@siemens.com](mailto:applications.erlf@siemens.com)

---