

CAN open Konfiguration

FAQ No.0004

Part	Version	Revision	Date	Status
de	6.2.3.0	002	2020-03-12	released

Inhalt

Einführung	2
Basic CAN Informationen	2
C6 Schnittstelle	2
C6 Econ / Perform Runtime Konfiguration.....	2
C6 CAN Master	3
Geräte Hinzufügen.....	3
Installieren der eds Datei.....	4
CANopen Manager Konfiguration	6
C6 CANopen Slave Konfiguration	7
PDO (Process Data Object) Konfiguration	8
SDO (Service Data Object) Konfiguration	9
Zusätzliche C6 COMPACT Konfigurationen.....	12
C6 CAN Slave	14
Geräte hinzufügen	14
Beispielcode.....	16
PLC_PRG.....	16
GVL UserODList.....	18
GVL UserVars	22
SDO Parameter für C6 Slave	23
Disclaimer	28



Einführung

Dieses Tutorial beschreibt das grundlegende Setup zur Verwendung der KEB C6-Steuerungen als CAN-Master oder Can-Slave. Das Basis-Setup enthält einige grundlegende Informationen über den CAN-Bus, die PDO- und SDO-Konfiguration, Synchronisation, Heartbeat und Node Guarding.

Basic CAN Informationen

In diesem Kapitel finden Sie einige grundlegende Informationen zum Feldbussystem CANopen.

C6 Schnittstelle

Um den CAN-Feldbus nutzen zu können, müssen Sie eine spezielle Hardware installiert haben. Bei Verwendung eines C6 ist darauf zu achten, dass eine CAN-Erweiterungskarte installiert ist.

Die Erweiterung für C6 Econ / Perform bietet zwei D-Sub-Stecker vom Typ **DE-9M**.

Die Erweiterung für C6 Compact bietet einen D-Sub-Stecker vom Typ **DE-9F**.

Die Pinbelegung ist CANopen-kompatibel. Das erste und letzte Gerät am Feldbus muss einen 120Ω Widerstand für den Busabschluss verwenden.

Connector pin	Beschreibung
2	CAN_L Bus line
7	CAN_H Bus line
5	CAN cable shield
Case	CAN cable shield

C6 Econ / Perform Runtime Konfiguration

Wenn Sie eine C6 Econ oder Perform haben, müssen Sie die Konfigurationsdatei der Runtime PLC ändern. Gehen Sie in den Installationsordner der Runtime (sollte in *C:\Program Files\3S CoDeSys\CoDeSys Control RTE3*) und benennen Sie die Datei "**CoDeSysControl.cfg**" um zu "**CoDeSysControl_original.cfg**" (oder etwas Ähnliches). Dann ändern Sie den Namen "**Econ-SM-Can-CoDeSysControl.cfg**" zu "**CoDeSysControl.cfg**" und starten die Runtime neu. Nun sollten Sie in der Lage sein, die CAN-Schnittstelle richtig nutzen zu können.

C6 CAN Master

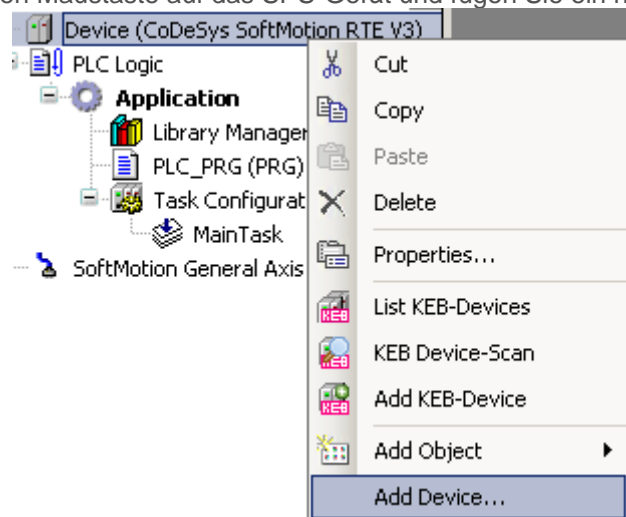
In diesem Kapitel wird beschrieben, wie Sie Ihre C6 als CAN-Master-Gerät konfigurieren. Es gibt einige Unterschiede zwischen der C6 COMPACT und dem C6 Econ/Perform.

Es gibt einige Beispielprojekte, die mit COMBIVIS studio 6 installiert wurden.

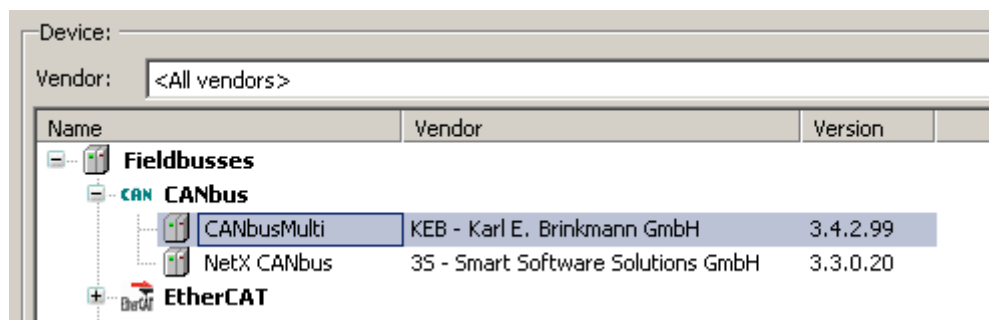
Dieses Kapitel beschreibt den Aufbau eines C6 Econ als CAN-Master mit einem CAN-Slave (auch C6), beginnend mit einem Standardprojekt.

Geräte Hinzufügen

Klicken Sie mit der rechten Maustaste auf das SPS-Gerät und fügen Sie ein neues Gerät hinzu.

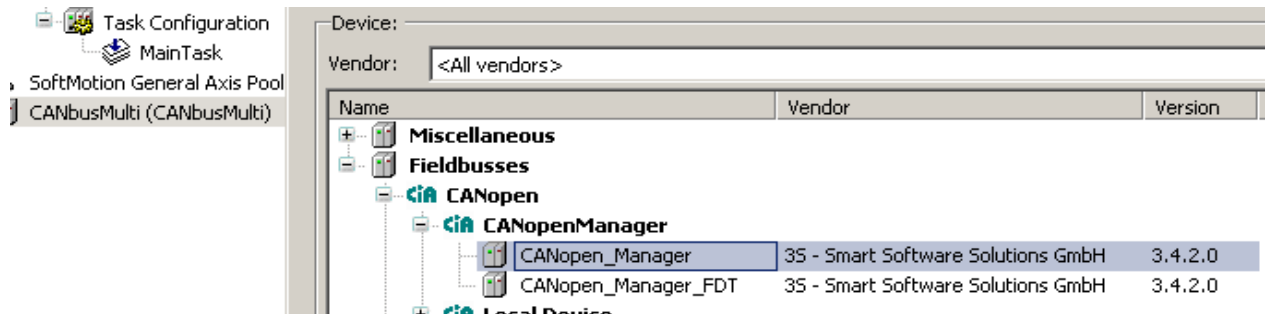


Wählen Sie das CANbusMulti Device aus Ihrem Geräte-Repository aus und fügen Sie es zu Ihrem Projekt hinzu.



Nun müssen wir dem CANbusMulti-Gerät ein CANOpen_Manager-Objekt hinzufügen. Sie können dies einfach mit einem Klick auf das zuletzt hinzugefügte CANbus-Gerät tun, wenn der Dialog Gerät hinzufügen noch geöffnet ist.

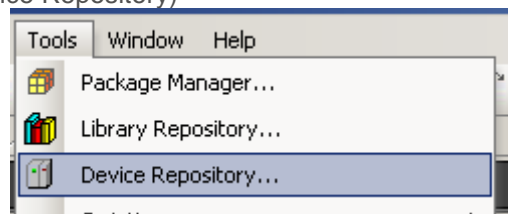
FAQ COMBIVIS studio 6



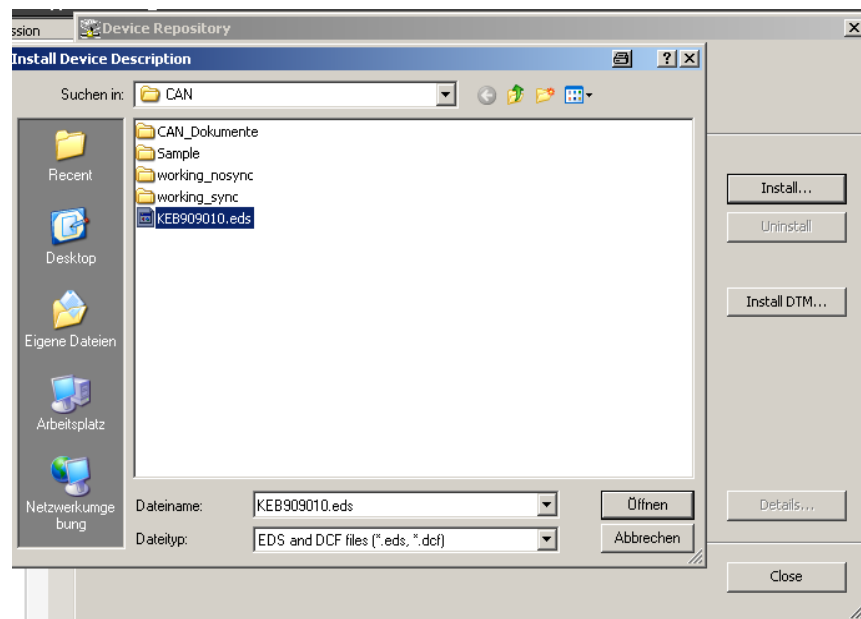
Nun müssen wir ein Slave zum Manager-Gerät hinzufügen. Wenn Sie ein C6 als Slave-Gerät verwenden, müssen Sie die mitgelieferte .eds-Beschreibungsdatei in Ihrem Geräterepository installieren.

Installieren der eds Datei

Um eine eds-Datei zu installieren, müssen Sie das Device Repository öffnen (klicken Sie in COMBIVIS Studio auf Tools und das Device Repository)



Wählen Sie Install und ändern Sie den Dateityp auf EDS und DCF. Wählen Sie die KEB909010.eds aus.



Wenn die eds-Datei installiert ist, können Sie das Gerät an den Manager anhängen und im Projekt verwenden.

FAQ COMBIVIS studio 6

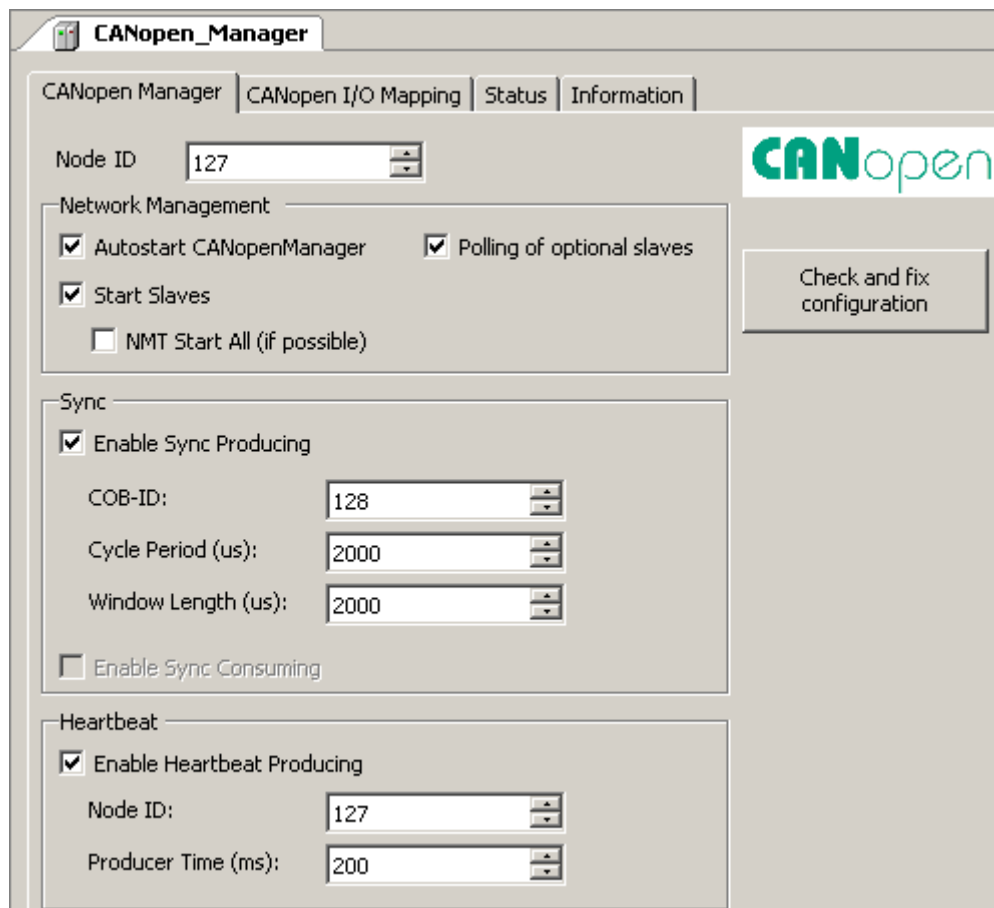


The screenshot shows the COMBIVIS studio 6 interface. On the left, a tree view contains the following items: MainTask, SoftMotion General Axis Pool, CANbusMulti (CANbusMulti), and CANopen_Manager (CANopen). The main window is titled 'Device:' and has a 'Vendor:' dropdown menu set to 'KEB - Karl E. Brinkmann GmbH'. Below this is a table with the following data:

Name	Vendor	Version
C6 CanOpen Slave	KEB - Karl E. Brinkmann GmbH	ProductVersion=0, ProductRevision=1, Filename=KEB909010
KEB Combivert_SoftMotion	KEB - Karl E. Brinkmann GmbH	3.4.2.0

CANopen Manager Konfiguration

Öffnen Sie nun den CANopen_Manager, um die Konfiguration zu bearbeiten.



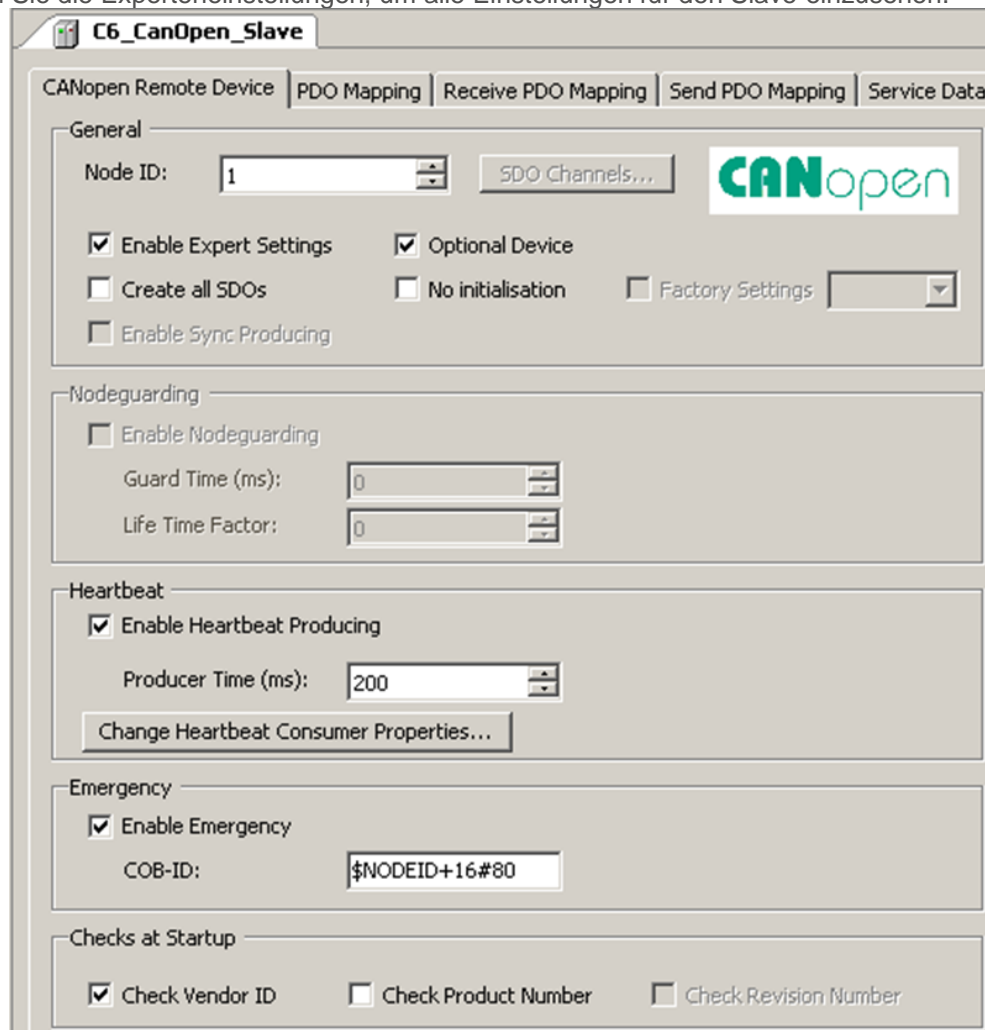
Der CANopen_Manager ist das Master-Gerät und steuert einige grundlegende CAN-Funktionen.

Wenn Sie synchrone Kommunikation verwenden möchten, müssen Sie die Synchronisation aktivieren.
Hinweis: Stellen Sie sicher, dass die SPS-Zykluszeit gleich der Zykluszeit ist, die Sie hier im Manager eingestellt haben.

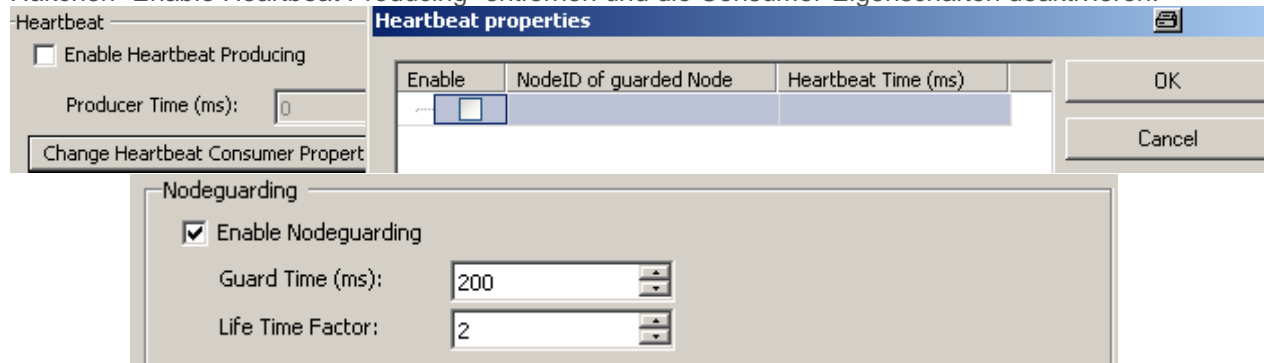
Um nach Slave-Ausfällen zu suchen, können Sie Heartbeat-Produktion oder NodeGuarding aktivieren. Der Heartbeat wird an alle Slaves gesendet und alle Slaves senden eine Antwort darauf. Mit dieser Funktion kann der Master erkennen, ob ein Slave ausgefallen ist und andererseits können die Slaves überprüfen, ob ein Fehler mit dem CAN-Master vorliegt. Die Produzentenzeit ist das Zeitintervall in Millisekunden für den Heartbeat.

C6 CANopen Slave Konfiguration

Nun müssen Sie die Slaves konfigurieren, die an den Master angehängt werden. Aktivieren Sie die Experteneinstellungen, um alle Einstellungen für den Slave einzusehen.



Sie haben zwei Möglichkeiten der Busüberwachung. Eine Methode ist die Heartbeat-Produktion (muss in Master und Slave mit identischer Produzentenzeit aktiviert werden). Die zweite Methode ist das so genannte Nodeguarding. Um die Kontrollkästchen für Nodeguarding zu aktivieren, müssen Sie das Häkchen "Enable Heartbeat Producing" entfernen und die Consumer-Eigenschaften deaktivieren.



Enable	NodeID of guarded Node	Heartbeat Time (ms)
<input type="checkbox"/>		

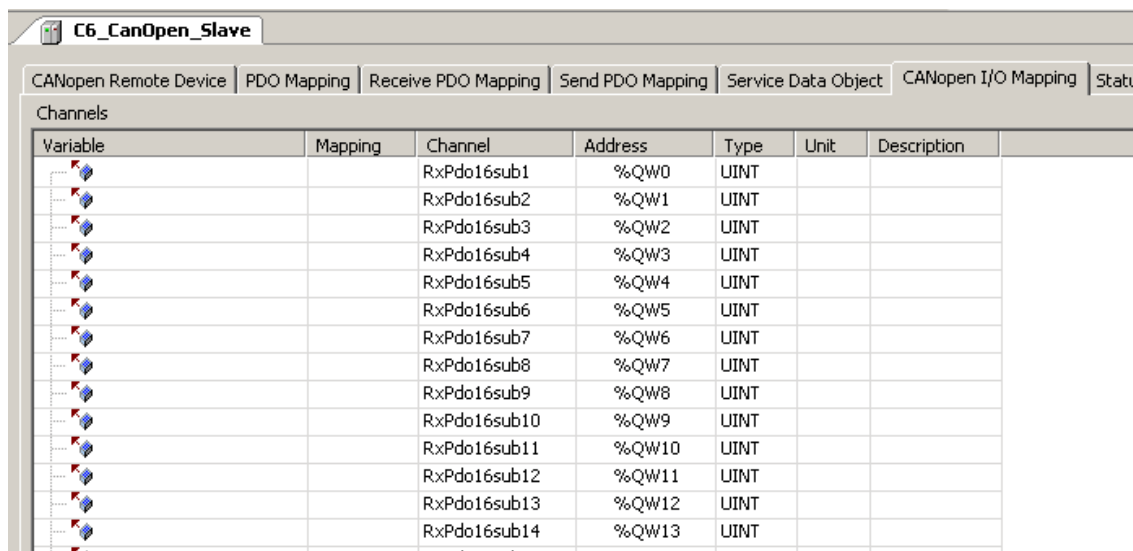
Sie können nur eine dieser beiden Methoden verwenden. Wenn Sie Heartbeat verwenden, stellen Sie sicher, dass alle Heartbeat-Zeiten gleich sind. Wenn NodeGuarding aktiv ist, kann der Master erkennen, wenn ein Slave ausfällt. Der Slave selbst erkennt jedoch nicht, dass der CAN-Master fehlt. Um dies zu beheben, müssen Sie die Lifeguarding-Funktion auf dem Slave aktivieren (Standard aktiv im KEB C6 Slave).

PDO (Process Data Object) Konfiguration

Nun können Sie sich Ihre PDO-Daten ansehen. Es gibt 4 PDO-Gruppen zum Empfangen und 4 Gruppen zum Senden. Erweiterte Einstellungen wie Synchronbetrieb sind im Kapitel SDO-Konfiguration beschrieben.

CANopen Remote Device				PDO Mapping	Receive PDO Mapping	Send PDO Mapping
Name	Index	Subindex	Bitlength			
[-] 1st transmit PDO Parameter	16#1800	16#00				
[-] TxPdo16sub1	16#2014	16#01	16			
[-] TxPdo16sub2	16#2014	16#02	16			
[-] TxPdo16sub3	16#2014	16#03	16			
[-] TxPdo16sub4	16#2014	16#04	16			
[-] 2nd transmit PDO Parameter	16#1801	16#00				
[-] TxPdo16sub5	16#2014	16#05	16			
[-] TxPdo16sub6	16#2014	16#06	16			
[-] TxPdo16sub7	16#2014	16#07	16			
[-] TxPdo16sub8	16#2014	16#08	16			
[-] 3rd transmit PDO Parameter	16#1802	16#00				
[-] TxPdo16sub9	16#2014	16#09	16			
[-] TxPdo16sub10	16#2014	16#0A	16			
[-] TxPdo16sub11	16#2014	16#0B	16			
[-] TxPdo16sub12	16#2014	16#0C	16			
[-] 4th transmit PDO Parameter	16#1803	16#00				
[-] TxPdo16sub13	16#2014	16#0D	16			
[-] TxPdo16sub14	16#2014	16#0E	16			
[-] TxPdo16sub15	16#2014	16#0F	16			
[-] TxPdo16sub16	16#2014	16#10	16			

Auf der Registerkarte CANopen I/O Mapping, können Sie den PDO-Daten ein variables Mapping hinzufügen.



Variable	Mapping	Channel	Address	Type	Unit	Description
		RxPdo16sub1	%QW0	UINT		
		RxPdo16sub2	%QW1	UINT		
		RxPdo16sub3	%QW2	UINT		
		RxPdo16sub4	%QW3	UINT		
		RxPdo16sub5	%QW4	UINT		
		RxPdo16sub6	%QW5	UINT		
		RxPdo16sub7	%QW6	UINT		
		RxPdo16sub8	%QW7	UINT		
		RxPdo16sub9	%QW8	UINT		
		RxPdo16sub10	%QW9	UINT		
		RxPdo16sub11	%QW10	UINT		
		RxPdo16sub12	%QW11	UINT		
		RxPdo16sub13	%QW12	UINT		
		RxPdo16sub14	%QW13	UINT		
		RxPdo16sub15	%QW14	UINT		

Hinweis: Empfangen und Senden werden aus der Slave-Ansicht ausgewählt!

SDO (Service Data Object) Konfiguration

Ein Standard-SDO wird durch die eds-Datei bereitgestellt, die Sie zuvor installiert haben. Mit den Grundeinstellungen können Sie nun arbeiten. Wenn Sie zyklische oder synchrone Daten verwenden wollen, müssen Sie einige zusätzliche Einstellungen auf der Registerkarte SDO im Konfigurationseditor vornehmen. Die grauen Daten sind Teil des Standard-SDO und können nicht bearbeitet oder gelöscht werden. Sie können aber am Ende noch einmal die gleichen Parameter hinzufügen, um die Daten zu überschreiben und Ihre eigenen Einstellungen zu verwenden.

Synchrone PDO

Um eine synchrone Kommunikation zu ermöglichen, müssen Sie die Zykluszeit zu Ihrem SDO hinzufügen.

Wählen Sie auf der Registerkarte SDO „Neu...“ und markieren Sie den Eintrag für den Zeitraum des Kommunikationszyklus und ändern Sie den Wert auf Ihre Zykluszeit.

FAQ COMBIVIS studio 6



le COB-ID: 16#00000181 | 32

Select item from object directory

Index:Subindex	Name	AccessType	Type	Default
16#1006:16#00	communication cycle period	RW	UDINT	0
16#100C:16#00	guard time	RW	UINT	0
16#100D:16#00	Life Time Factor	RW	USINT	16#00
16#1014:16#00	COB ID EMCY	RW	UDINT	\$NODEID+16#80
16#1016	Consumer Heartbeat Time			
16#1017:16#00	Producer Heartbeat Time	RW	UINT	0
16#1400	1st receive PDO Parameter			
16#1401	2nd receive PDO Parameter			
16#1402	3rd receive PDO Parameter			
16#1403	4th receive PDO Parameter			
16#1600	1st receive PDO Mapping			
16#1601	2nd receive PDO Mapping			
16#1602	3rd receive PDO Mapping			
16#1603	4th receive PDO Mapping			
16#1800	1st transmit PDO Parameter			
16#1801	2nd transmit PDO Parameter			
16#1802	3rd transmit PDO Parameter			
16#1803	4th transmit PDO Parameter			
16#1A00	1st transmit PDO Mapping			

Name: communication cycle period

Index: 16# 1006 Bitlength: 32

SubIndex: 16# 0 Value: 2000

OK Cancel New...

Wenn Sie dies getan haben, können Sie für jede Gruppe von PDO-Daten unterschiedliche Einstellungen für die Übertragung wählen. Die Standardeinstellung für das PDO ist der Übertragungstyp 254, d.h. asynchrone Übertragung.

Die Tabelle zeigt die möglichen Übertragungsarten:

Wert (decimal)	Type
0	Azyklisch synchron
1 – 240	Zyklisch synchron. Die Zahl bedeutet die Anzahl der Zyklen zwischen den einzelnen Übertragungen (1 bis 240 Zyklen sind also möglich)
241 – 251	reserviert
252	Synchronous RTR (Remote Transmision Request)
253	Asynchronous RTR (Remote Transmision Request)
254 – 255	Asynchronous Übertragung (standard)

Nun können Sie die Übertragungsarten der verschiedenen PDO-Gruppen ändern. Wenn Sie die erste RX- und erste TX-Gruppe für die synchrone Kommunikation verwenden möchten, dann müssen Sie diese Daten dem SDO hinzufügen:

80	16#1800:16#02	transmission type	1	8
81	16#1400:16#02	transmission type	1	8

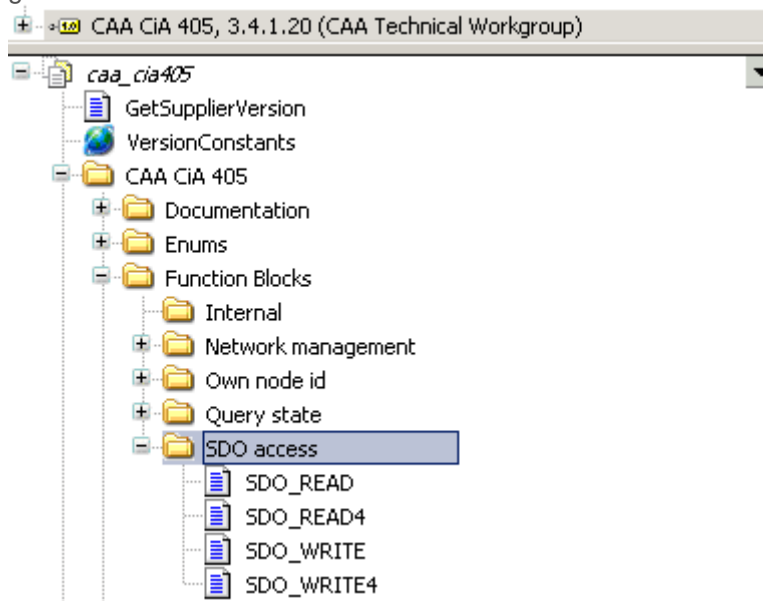
Dies ist eine grundlegende Einrichtung, um synchrone PDO-Daten zu ermöglichen.

Zusätzliche SDO Notizen

Das SDO, das Sie in der grafischen Oberfläche konfigurieren können, wird nur einmal übertragen, wenn sich der Slave mit dem Master verbindet.

Wenn Sie ein SDO senden möchten, während Ihre Anwendung läuft, müssen Sie einen CAA-Funktionsblock verwenden. Um diesen Funktionsbaustein hinzuzufügen, müssen Sie die CiA 405 Bibliothek in Ihr Projekt aufnehmen. Mit den Funktionsblöcken SDO_READ und SDO_WRITE können neue SDO-Daten mit einem Slave ausgetauscht werden.

Eine Liste der verfügbaren SDO-Parameter finden Sie am Ende dieses Dokuments.

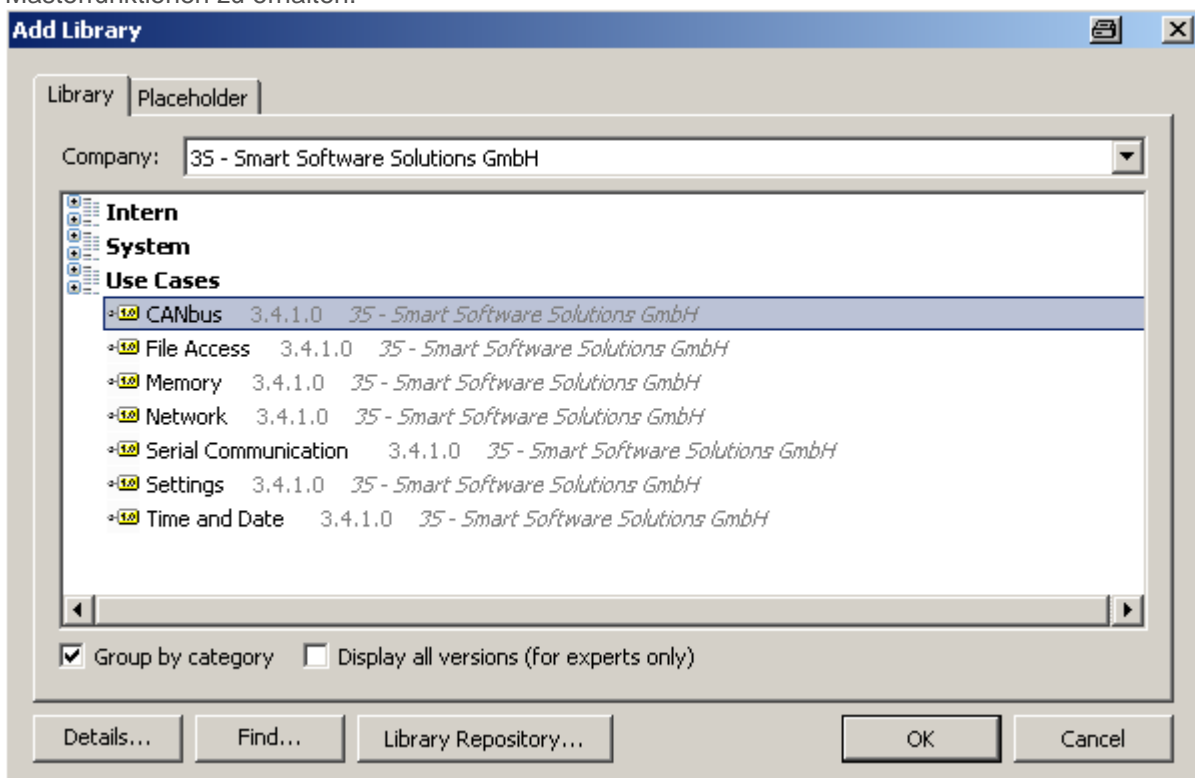


Wenn Sie mit dem Setup fertig sind, kompilieren Sie die Anwendung und loggen sich bei Ihrem Gerät ein.

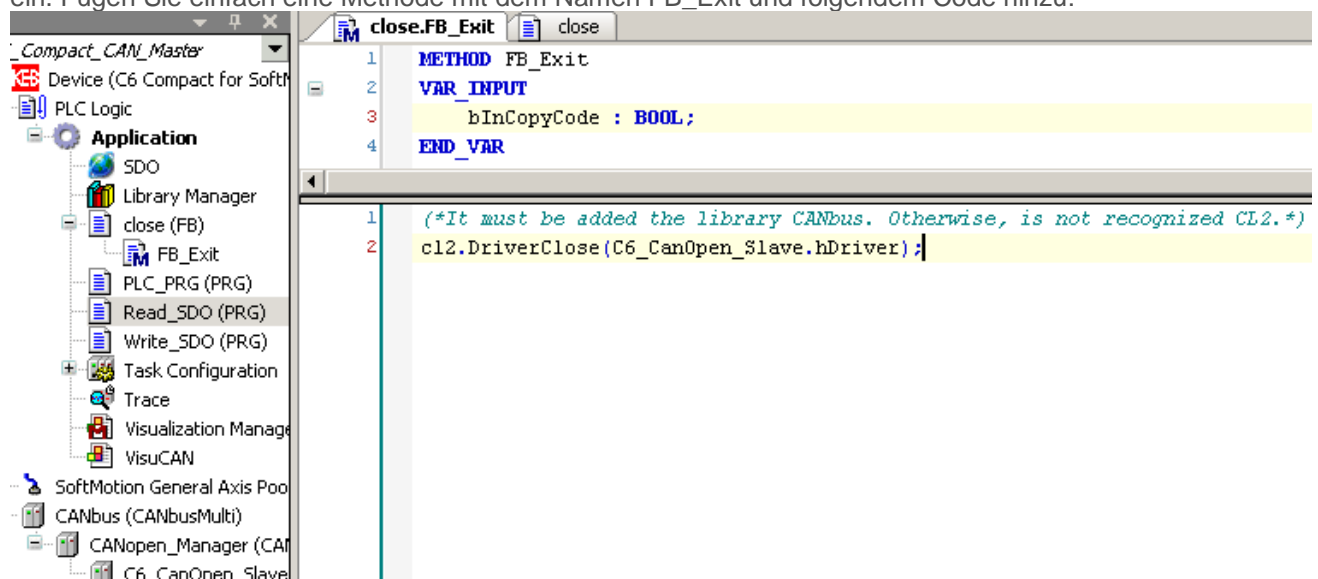
Zusätzliche C6 COMPACT Konfigurationen

Wenn Sie eine C6 Compact als Master verwenden möchten, müssen Sie in Ihrem Projekt einige zusätzliche Konfigurationen vornehmen, wenn Sie einen **C6 als Slave verwenden**.

Zuerst benötigen wir eine zusätzliche Bibliothek, um Zugang zu einigen spezifischen CAN-Masterfunktionen zu erhalten.



Nun müssen Sie in Ihrem Projekt einen neuen Funktionsblock anlegen. Fügen Sie keinen Quellcode ein. Fügen Sie einfach eine Methode mit dem Namen `FB_Exit` und folgendem Code hinzu:



FAQ COMBIVIS studio 6



Anschließend rufen Sie den Funktionsblock im PLC_PRG auf:

A screenshot of a PLC programming software interface. The top window shows three tabs: 'close.FB_Exit', 'close', and 'PLC_PRG'. The 'PLC_PRG' tab is active, displaying the following code:

```
1 PROGRAM PLC_PRG
2 VAR
3   closecan: close;
4 END_VAR
```

The bottom window shows a single line of code:

```
1 closecan();
```

Jetzt können Sie die C6 COMPACT als CAN-Master einsetzen. Der Rest des Setups ist derselbe wie bei dem C6 Econ/Perform.

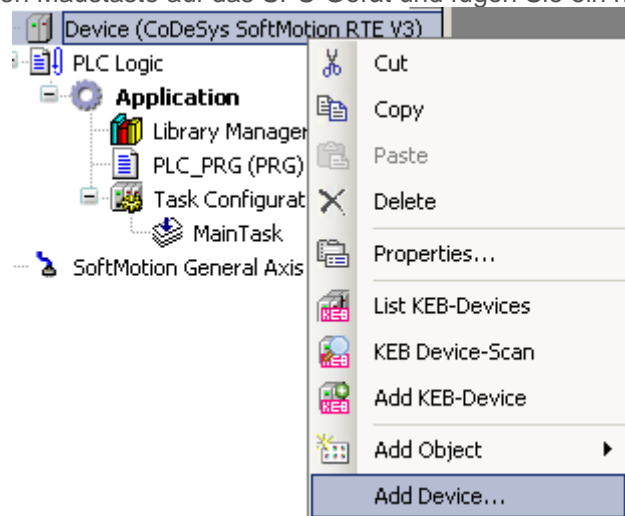
C6 CAN Slave

In diesem Kapitel wird erklärt, wie Sie eine C6 Compact als CAN-Slave einrichten, beginnend mit einem Standardprojekt.

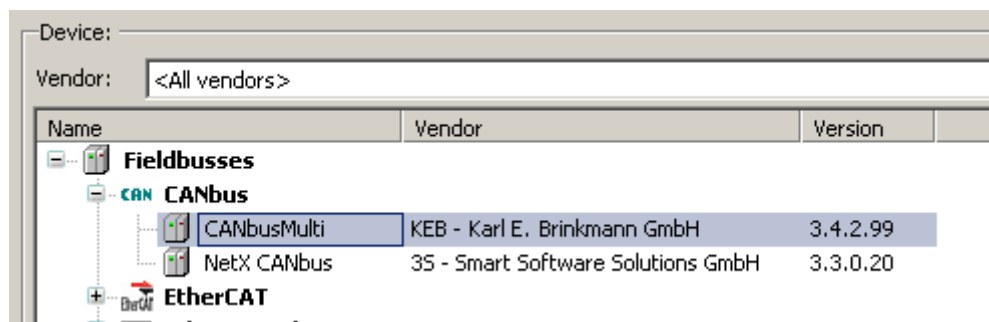
Hinweis: Der C6 Econ/Perform kann nicht als CAN-Slave verwendet werden.

Geräte hinzufügen

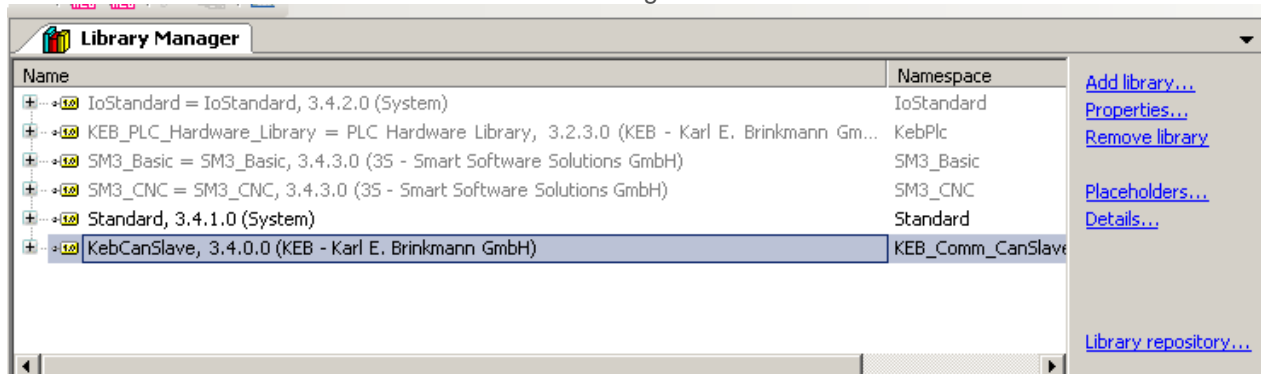
Klicken Sie mit der rechten Maustaste auf das SPS-Gerät und fügen Sie ein neues Gerät hinzu.



Wählen Sie dann das CANbusMulti Device aus Ihrem Geräte-Repository aus und fügen Sie es zu Ihrem Projekt hinzu.



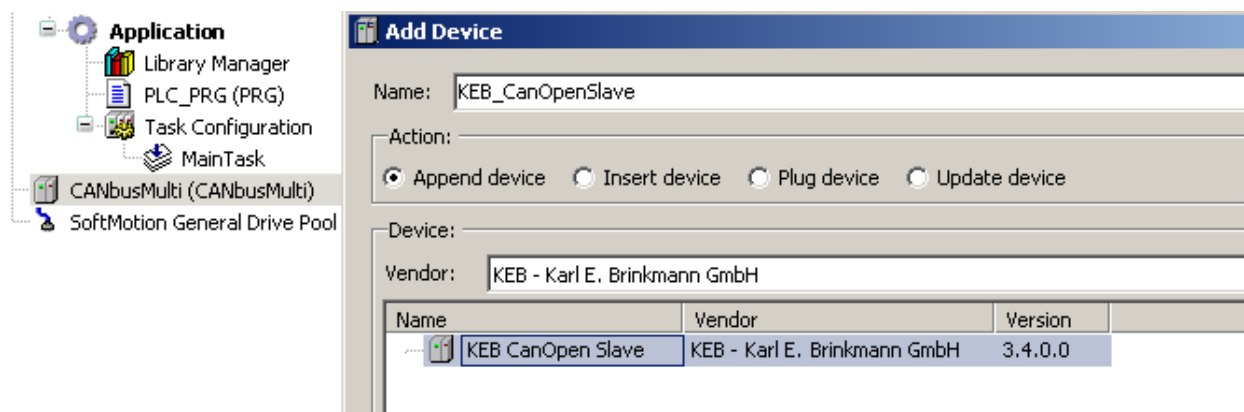
Jetzt müssen wir die KEBCanSlave-Bibliothek hinzufügen.



FAQ COMBIVIS studio 6



Sobald die Bibliothek hinzugefügt ist, können wir das KEB-Gerät dem CAN-Bus einbinden.



Nun der schwierige Teil: Die Einrichtung des ObjectDictionary. Es wird empfohlen, das Beispielprojekt CANopenDevice zu verwenden, welches mit COMBIVIS studio 6 installiert wird, aber Sie finden den benötigten Quellcode ebenfalls im Anhang.

Im Beispielprojektordner befinden sich auch zwei Parameterbeschreibungsdateien, die Sie in Ihrem COMBIVIS Studio 6 installieren müssen (Wählen Sie "Tools"->"KEB-Parameterbeschreibungsdatei importieren" und fügen Sie die beiden Dateien aus dem Musterverzeichnis hinzu).

Um das ObjectDictionary zum Laufen zu bringen, müssen Sie dem Projekt 3 weitere Libraries hinzufügen:

+ [Icon]	Din66019II, 3.4.2.0 (KEB - Karl E. Brinkmann GmbH)	KEB_Comm_Din66019II	3.4.2.0
+ [Icon]	KEB_ChannelHandler, 3.4.2.0 (KEB - Karl E. Brinkmann GmbH)	KEB_Comm_ChannelHandler	3.4.2.0
+ [Icon]	ObjectDictionary, 3.4.2.1 (KEB - Karl E. Brinkmann GmbH)	KEB_ObjectDictionary	3.4.2.1

Wenn diese Bibliotheken enthalten sind, müssen Sie nun folgende Schritte in ihrem PLC_PRG durchführen:

1. Rufen Sie den Channelhandler in jedem Zyklus auf
2. Setup im ersten Zyklus durchführen:
 - a. ObjectDictionary für PDO-Parameter aktivieren
 - b. Hinzufügen von ObjectDictionary zum CAN-Slave
 - c. Synchronmodus aktivieren (wenn Sie möchten)

Hinweis: Im Synchronisationsmodus muss die Zykluszeit der Aufgaben gleich der Synchronisationszeit des CAN sein).

 - d. Registrierung des DIN-Slaves bei Channelhandler
 - e. CAN-Slave als Master am Channelhandler registrieren.

Um die Synchronisation zu ermöglichen, müssen Sie im Setup-Schritt einige Eingabewerte einstellen. Ändern Sie den Eingang **SyncPhase** des Slaves auf Null (0), um den Synchronisationsmodus zu aktivieren. Um es auszuschalten, müssen Sie den Wert 30000 einstellen.

Wenn die **SyncPhase** Null ist, können Sie das **CycleCorrectionVal** ändern. Abhängig von der Buslast und dem Jitter auf dem Bus kann es erforderlich sein, den Wert zu erhöhen, um CAN im synchronen Modus zu erhalten.

Die Ausgabe **CanInSync** zeigt den Synchronisationsstatus an.

FAQ COMBIVIS studio 6

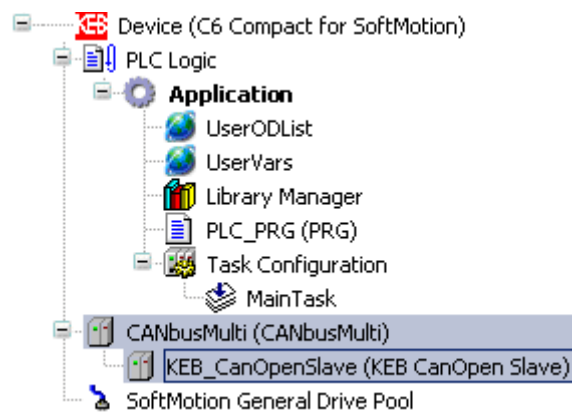


```
KEB_CanOpenSlave.SyncPhase := 0;  
KEB_CanOpenSlave.CycleCorrectionVal := 100;
```

Achtung: Wenn Sie eine C6 Compact verwenden und eine Messkurve herunterladen oder viele Variablen überwachen, ist es möglich, dass der CAN-Bus den Synchronmodus verlässt.

Beispielcode

Dies ist ein Beispielcode aus dem Beispielprojekt, das mit COMBIVIS studio 6 installiert wurde. Denken Sie daran, die Geräte zu Ihrem Projekt hinzuzufügen.



Die Prozessdaten werden dem User Object Dictionary zugeordnet. Sie können die Arrays für Ihre PDO-Daten verwenden.

PLC_PRG

```
PROGRAM PLC_PRG  
VAR
```

```
    started: BOOL;  
    CH : KebChannelHandler;  
    SL : Din_Slave_Udp;
```

```
END_VAR
```

```
CH()); //ALWAYS run the channelhandler
```

```
IF NOT started THEN  
    started := TRUE;
```

```
    //User OD and CanOpen slave 1  
    UserOd.pParaList := ADR(OdParas);  
    UserOd.listSize := SIZEOF(OdParas) / SIZEOF(OD_ITEM);  
    KEB_CanOpenSlave.UserOD := ADR(UserOD);
```

```
    //Enable Sync:  
    KEB_CanOpenSlave.SyncPhase := 0;  
    KEB_CanOpenSlave.CycleCorrectionVal := 100;
```


FAQ COMBIVIS studio 6



```
//the Din66019 UDP slave
CH.RegisterSlave(SL);
//The Canopen slave as master to access the CAN-Od's
CH.RegisterMaster(KEB_CanOpenSlave,1,1);
END_IF
```

FAQ COMBIVIS studio 6



GVL UserODList

```
VAR_GLOBAL
//Automatically generated from xml file <Paras> section with Paralst Editor V3.1.0.8
//=====================================================
```

```
OdParas : ARRAY[0..6] OF OD_ITEM := [
//RxPdo8
(
  Index:=16#2010,           //Index
  SubIdxMod:=16#0000,      //SubIdxMod
  Itype:=16#1000 OR
  OD_TBL_DATATYPE_UINT8 OR
  OD_TBL_OBJTYPE_ARRAY,   //Type
  Prop1:=OD_TBL_PWLVRD_0 OR
  OD_TBL_PWLVRD_0 OR
  OD_TBL_DEFVAL_FIX OR
  OD_TBL_ULIM_FIX OR
  OD_TBL_LLIM_FIX OR
  OD_TBL_RW OR
  OD_TBL_SS_FIX OR
  OD_TBL_PD_YES,          //Prop1
  ReadFuncIdx:=0,         //ReadFuncIdx
  WriteFuncIdx:=0,        //WriteFuncIdx
  LowerLim:=0,            //LowerLim
  UpperLim:=65535,        //UpperLim
  DefValue:=0,            //DefValue
  pData:=ADR(Para2010[0]) //pData
),
//RxPdo16
(
  Index:=16#2011,           //Index
  SubIdxMod:=16#0000,      //SubIdxMod
  Itype:=16#1000 OR
  OD_TBL_DATATYPE_UINT16 OR
  OD_TBL_OBJTYPE_ARRAY,   //Type
  Prop1:=OD_TBL_PWLVRD_0 OR
  OD_TBL_PWLVRD_0 OR
  OD_TBL_DEFVAL_FIX OR
  OD_TBL_ULIM_FIX OR
  OD_TBL_LLIM_FIX OR
  OD_TBL_RW OR
  OD_TBL_SS_FIX OR
  OD_TBL_PD_YES,          //Prop1
  ReadFuncIdx:=0,         //ReadFuncIdx
  WriteFuncIdx:=0,        //WriteFuncIdx
  LowerLim:=0,            //LowerLim
  UpperLim:=65535,        //UpperLim
  DefValue:=0,            //DefValue
  pData:=ADR(Para2011[0]) //pData
),
//RxPdo32
```

FAQ COMBIVIS studio 6



```
(
  Index:=16#2012,                //Index
  SubIdxMod:=16#0000,           //SubIdxMod
  Itype:=OD_TBL_ITEM_ENTRIES_8 OR
  OD_TBL_DATATYPE_UINT32 OR
  OD_TBL_OBJTYPE_ARRAY,        //Type
  Prop1:=OD_TBL_PWLVWR_0 OR
  OD_TBL_PWLVRD_0 OR
  OD_TBL_DEFVAL_FIX OR
  OD_TBL_ULIM_FIX OR
  OD_TBL_LLIM_FIX OR
  OD_TBL_RW OR
  OD_TBL_SS_FIX OR
  OD_TBL_PD_YES,              //Prop1
  ReadFuncIdx:=0,             //ReadFuncIdx
  WriteFuncIdx:=0,           //WriteFuncIdx
  LowerLim:=0,                //LowerLim
  UpperLim:=-1,              //UpperLim
  DefValue:=0,                //DefValue
  pData:=ADR(Para2012[0])     //pData
),
//TxPdo8
(
  Index:=16#2013,                //Index
  SubIdxMod:=16#0000,           //SubIdxMod
  Itype:=16#1000 OR
  OD_TBL_DATATYPE_UINT8 OR
  OD_TBL_OBJTYPE_ARRAY,        //Type
  Prop1:=OD_TBL_PWLVWR_0 OR
  OD_TBL_PWLVRD_0 OR
  OD_TBL_DEFVAL_FIX OR
  OD_TBL_ULIM_FIX OR
  OD_TBL_LLIM_FIX OR
  OD_TBL_RO OR
  OD_TBL_SS_FIX OR
  OD_TBL_PD_YES,              //Prop1
  ReadFuncIdx:=0,             //ReadFuncIdx
  WriteFuncIdx:=0,           //WriteFuncIdx
  LowerLim:=0,                //LowerLim
  UpperLim:=65535,            //UpperLim
  DefValue:=0,                //DefValue
  pData:=ADR(Para2013[0])     //pData
),
//TxPdo16
(
  Index:=16#2014,                //Index
  SubIdxMod:=16#0000,           //SubIdxMod
  Itype:=16#1000 OR
  OD_TBL_DATATYPE_UINT16 OR
  OD_TBL_OBJTYPE_ARRAY,        //Type
  Prop1:=OD_TBL_PWLVWR_0 OR
  OD_TBL_PWLVRD_0 OR
  OD_TBL_DEFVAL_FIX OR
```

FAQ COMBIVIS studio 6



```
OD_TBL_ULIM_FIX OR
OD_TBL_LLIM_FIX OR
OD_TBL_RO OR
OD_TBL_SS_FIX OR
OD_TBL_PD_YES,           //Prop1
ReadFuncIdx:=0,         //ReadFuncIdx
WriteFuncIdx:=0,        //WriteFuncIdx
LowerLim:=0,            //LowerLim
UpperLim:=65535,        //UpperLim
DefValue:=0,            //DefValue
pData:=ADR(Para2014[0]) //pData
),
//TxPdo32
(
  Index:=16#2015,        //Index
  SubIdxMod:=16#0000,    //SubIdxMod
  Itype:=OD_TBL_ITEM_ENTRIES_8 OR
OD_TBL_DATATYPE_UINT32 OR
OD_TBL_OBJTYPE_ARRAY,   //Type
  Prop1:=OD_TBL_PWLVRD_0 OR
OD_TBL_PWLVRD_0 OR
OD_TBL_DEFVAL_FIX OR
OD_TBL_ULIM_FIX OR
OD_TBL_LLIM_FIX OR
OD_TBL_RO OR
OD_TBL_SS_FIX OR
OD_TBL_PD_YES,           //Prop1
ReadFuncIdx:=0,         //ReadFuncIdx
WriteFuncIdx:=0,        //WriteFuncIdx
LowerLim:=0,            //LowerLim
UpperLim:=-1,           //UpperLim
DefValue:=0,            //DefValue
pData:=ADR(Para2015[0]) //pData
),
//cfgId
(
  Index:=16#160F,        //Index
  SubIdxMod:=16#0000,    //SubIdxMod
  Itype:=OD_TBL_ITEM_ENTRIES_0 OR
OD_TBL_DATATYPE_UINT32 OR
OD_TBL_OBJTYPE_VAR,      //Type
  Prop1:=OD_TBL_PWLVRD_0 OR
OD_TBL_PWLVRD_0 OR
OD_TBL_DEFVAL_FIX OR
OD_TBL_ULIM_FIX OR
OD_TBL_LLIM_FIX OR
OD_TBL_RO OR
OD_TBL_PD_NO,           //Prop1
ReadFuncIdx:=0,         //ReadFuncIdx
WriteFuncIdx:=0,        //WriteFuncIdx
LowerLim:=1001001,      //LowerLim
UpperLim:=1001001,      //UpperLim
DefValue:=1001001,      //DefValue
```

FAQ COMBIVIS studio 6



```
pData:=ADR(ParaCfgId)           //pData
)
];
END_VAR
```

FAQ COMBIVIS studio 6



GVL UserVars

VAR_GLOBAL

```
Para2010 : ARRAY [0..15] OF BYTE; //RxPdo8
Para2011 : ARRAY [0..15] OF WORD; //RxPdo16
Para2012 : ARRAY [0..7] OF DWORD; //RxPdo32
Para2013 : ARRAY [0..15] OF BYTE; //TxPdo8
Para2014 : ARRAY [0..15] OF WORD; //TxPdo16
Para2015 : ARRAY [0..7] OF DWORD; //TxPdo32
ParaCfgId : UDINT := 1001001; //fixed value for cfg id
UserOd : ObjectDictionary;
```

END_VAR



SDO Parameter für C6 Slave

Index	Subindex	Name	Bitlänge	Standard Wert
16#1006	16#00	communication cycle period	32	0
16#100C	16#00	guard time	16	0
16#100D	16#00	Life time factor	8	0
16#1014	16#00	COB ID EMCY	32	\$NODEID+16#80
16#1016	16#01	Consumer Heartbeat time	32	0
16#1017	16#00	Producer Heartbeat time	16	0
16#1400		1st receive PDO Parameter		
	16#01	COB-ID used by PDO	32	\$NODEID+16#200
	16#02	transmission type	8	254
16#1401		2nd receive PDO Parameter		
	16#01	COB-ID used by PDO	32	\$NODEID+16#300
	16#02	transmission type	8	254
16#1402		3rd receive PDO Parameter		
	16#01	COB-ID used by PDO	32	\$NODEID+16#400
	16#02	transmission type	8	254
16#1403		4th receive PDO Parameter		
	16#01	COB-ID used by PDO	32	\$NODEID+16#500
	16#02	transmission type	8	254
16#1600		1st receive PDO Mapping		
	16#00	number of mapped objects	8	4
	16#01	1st PDO Mapping	32	16#20110110
	16#02	2nd PDO Mapping	32	16#20110210
	16#03	3rd PDO Mapping	32	16#20110310
	16#04	4th PDO Mapping	32	16#20110410
16#1601		2nd receive PDO Mapping		
	16#00	number of mapped objects	8	4
	16#01	1st PDO Mapping	32	16#20110510
	16#02	2nd PDO Mapping	32	16#20110610
	16#03	3rd PDO Mapping	32	16#20110710
	16#04	4th PDO Mapping	32	16#20110810
16#1602		3rd receive PDO Mapping		
	16#00	number of mapped objects	8	4
	16#01	1st PDO Mapping	32	16#20110910
	16#02	2nd PDO Mapping	32	16#20110A10
	16#03	3rd PDO Mapping	32	16#20110B10
	16#04	4th PDO Mapping	32	16#20110C10
16#1603		4th receive PDO Mapping		

FAQ COMBIVIS studio 6



Index	Subindex	Name	Bitlänge	Standard Wert
	16#00	number of mapped objects	8	4
	16#01	1st PDO Mapping	32	16#20110D10
	16#02	2nd PDO Mapping	32	16#20110E10
	16#03	3rd PDO Mapping	32	16#20110F10
	16#04	4th PDO Mapping	32	16#20111010
16#1800		1st transmit PDO Parameter		
	16#01	COB-ID used by PDO	32	\$NODEID+16#180
	16#02	transmission type	8	254
16#1800		2nd transmit PDO Parameter		
	16#01	COB-ID used by PDO	32	\$NODEID+16#280
	16#02	transmission type	8	254
16#1800		3rd transmit PDO Parameter		
	16#01	COB-ID used by PDO	32	\$NODEID+16#380
	16#02	transmission type	8	254
16#1800		4th transmit PDO Parameter		
	16#01	COB-ID used by PDO	32	\$NODEID+16#480
	16#02	transmission type	8	254
16#1A00		1st transmit PDO Mapping		
	16#00	number of mapped objects	8	4
	16#01	1st PDO Mapping	32	16#20140110
	16#02	2nd PDO Mapping	32	16#20140210
	16#03	3rd PDO Mapping	32	16#20140310
	16#04	4th PDO Mapping	32	16#20140410
16#1A01		2nd transmit PDO Mapping		
	16#00	number of mapped objects	8	4
	16#01	1st PDO Mapping	32	16#20140510
	16#02	2nd PDO Mapping	32	16#20140610
	16#03	3rd PDO Mapping	32	16#20140710
	16#04	4th PDO Mapping	32	16#20140810
16#1A02		3rd transmit PDO Mapping		
	16#00	number of mapped objects	8	4
	16#01	1st PDO Mapping	32	16#20140910
	16#02	2nd PDO Mapping	32	16#20140A10
	16#03	3rd PDO Mapping	32	16#20140B10
	16#04	4th PDO Mapping	32	16#20140C10
16#1A03		4th transmit PDO Mapping		
	16#00	number of mapped objects	8	4
	16#01	1st PDO Mapping	32	16#20140D10
	16#02	2nd PDO Mapping	32	16#20140E10

FAQ COMBIVIS studio 6



Index	Subindex	Name	Bitlänge	Standard Wert
	16#03	3rd PDO Mapping	32	16#20140F10
	16#04	4th PDO Mapping	32	16#20141010
16#2010		RxPdo8		
	16#01	RxPdo8sub1	8	
	16#02	RxPdo8sub2	8	
	16#03	RxPdo8sub3	8	
	16#04	RxPdo8sub4	8	
	16#05	RxPdo8sub5	8	
	16#06	RxPdo8sub6	8	
	16#07	RxPdo8sub7	8	
	16#08	RxPdo8sub8	8	
	16#09	RxPdo8sub9	8	
	16#0A	RxPdo8sub10	8	
	16#0B	RxPdo8sub11	8	
	16#0C	RxPdo8sub12	8	
	16#0D	RxPdo8sub13	8	
	16#0E	RxPdo8sub14	8	
	16#0F	RxPdo8sub15	8	
	16#10	RxPdo8sub16	8	
16#2011		RxPdo16		
	16#01	RxPdo16sub1	16	
	16#02	RxPdo16sub2	16	
	16#03	RxPdo16sub3	16	
	16#04	RxPdo16sub4	16	
	16#05	RxPdo16sub5	16	
	16#06	RxPdo16sub6	16	
	16#07	RxPdo16sub7	16	
	16#08	RxPdo16sub8	16	
	16#09	RxPdo16sub9	16	
	16#0A	RxPdo16sub10	16	
	16#0B	RxPdo16sub11	16	
	16#0C	RxPdo16sub12	16	
	16#0D	RxPdo16sub13	16	
	16#0E	RxPdo16sub14	16	
	16#0F	RxPdo16sub15	16	
	16#10	RxPdo16sub16	16	
16#2012		RxPdo32		
	16#01	RxPdo32sub1	32	
	16#02	RxPdo32sub2	32	

FAQ COMBIVIS studio 6



Index	Subindex	Name	Bitlänge	Standard Wert
	16#03	RxPdo32sub3	32	
	16#04	RxPdo32sub4	32	
	16#05	RxPdo32sub5	32	
	16#06	RxPdo32sub6	32	
	16#07	RxPdo32sub7	32	
	16#08	RxPdo32sub8	32	
	16#09	RxPdo32sub9	32	
	16#0A	RxPdo32sub10	32	
	16#0B	RxPdo32sub11	32	
	16#0C	RxPdo32sub12	32	
	16#0D	RxPdo32sub13	32	
	16#0E	RxPdo32sub14	32	
	16#0F	RxPdo32sub15	32	
	16#10	RxPdo32sub16	32	
16#2013		TxPdo8		
	16#01	TxPdo8sub1	8	
	16#02	TxPdo8sub2	8	
	16#03	TxPdo8sub3	8	
	16#04	TxPdo8sub4	8	
	16#05	TxPdo8sub5	8	
	16#06	TxPdo8sub6	8	
	16#07	TxPdo8sub7	8	
	16#08	TxPdo8sub8	8	
	16#09	TxPdo8sub9	8	
	16#0A	TxPdo8sub10	8	
	16#0B	TxPdo8sub11	8	
	16#0C	TxPdo8sub12	8	
	16#0D	TxPdo8sub13	8	
	16#0E	TxPdo8sub14	8	
	16#0F	TxPdo8sub15	8	
	16#10	TxPdo8sub16	8	
16#2014		TxPdo16		
	16#01	TxPdo16sub1	16	
	16#02	TxPdo16sub2	16	
	16#03	TxPdo16sub3	16	
	16#04	TxPdo16sub4	16	
	16#05	TxPdo16sub5	16	
	16#06	TxPdo16sub6	16	
	16#07	TxPdo16sub7	16	

FAQ COMBIVIS studio 6



Index	Subindex	Name	Bitlänge	Standard Wert
	16#08	TxPdo16sub8	16	
	16#09	TxPdo16sub9	16	
	16#0A	TxPdo16sub10	16	
	16#0B	TxPdo16sub11	16	
	16#0C	TxPdo16sub12	16	
	16#0D	TxPdo16sub13	16	
	16#0E	TxPdo16sub14	16	
	16#0F	TxPdo16sub15	16	
	16#10	TxPdo16sub16	16	
16#2015		TxPdo32		
	16#01	TxPdo32sub1	32	
	16#02	TxPdo32sub2	32	
	16#03	TxPdo32sub3	32	
	16#04	TxPdo32sub4	32	
	16#05	TxPdo32sub5	32	
	16#06	TxPdo32sub6	32	
	16#07	TxPdo32sub7	32	
	16#08	TxPdo32sub8	32	
	16#09	TxPdo32sub9	32	
	16#0A	TxPdo32sub10	32	
	16#0B	TxPdo32sub11	32	
	16#0C	TxPdo32sub12	32	
	16#0D	TxPdo32sub13	32	
	16#0E	TxPdo32sub14	32	
	16#0F	TxPdo32sub15	32	
	16#10	TxPdo32sub16	32	



Disclaimer

KEB Automation KG reserves the right to change/adapt specifications and technical data without prior notification. The safety and warning reference specified in this manual is not exhaustive. Although the manual and the information contained in it is made with care, KEB does not accept responsibility for misprint or other errors or resulting damages. The marks and product names are trademarks or registered trademarks of the respective title owners.

The information contained in the technical documentation, as well as any user-specific advice in verbal or in written form are made to the best of our knowledge and information about the application. However, they are considered for information only without responsibility. This also applies to any violation of industrial property rights of a third-party.

Inspection of our units in view of their suitability for the intended use must be done generally by the user. Inspections are particularly necessary, if changes are executed, which serve for the further development or adaptation of our products to the applications (hardware, software or download lists). Inspections must be repeated completely, even if only parts of hardware, software or download lists are modified.

Application and use of our units in the target products is outside of our control and therefore lies exclusively in the area of responsibility of the user.

KEB Automation KG
Südstraße 38 • D-32683 Barntrup
fon: +49 5263 401-0 • fax: +49 5263 401-116
net: www.keb.de • mail: info@keb.de