

BECKHOFF New Automation Technology

Handbuch | DE

sACN 2016

TwinCAT 3

Inhaltsverzeichnis

1	Vorwort	5
1.1	Hinweise zur Dokumentation	5
1.2	Sicherheitshinweise	6
2	Übersicht	7
2.1	Update History	8
3	Programmierung	9
3.1	Funktionsbausteine	9
3.1.1	FB_sACN	9
3.1.2	FB_sACN_Ex	12
3.1.3	FB_UDP_Connection	16
3.2	Strukturen, Schnittstellen	17
3.2.1	Strukturen	17
3.2.2	Schnittstellen	18
4	Beispiel	21
5	Anhang	23
5.1	Eigenschaften in Bezug auf Zeit	23
5.2	Statusanzeige	23
5.3	Zielhost und Ziel-Universe	24
5.4	Multicast-Gruppe	25

1 Vorwort

1.1 Hinweise zur Dokumentation

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.

Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.

Das Fachpersonal ist verpflichtet, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentlichte Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiter entwickelt.

Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.

Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

Marken

Beckhoff®, TwinCAT®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.

Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

EtherCAT®

EtherCAT® ist eine eingetragene Marke und patentierte Technologie lizenziert durch die Beckhoff Automation GmbH, Deutschland

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet.

Zu widerhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

1.2 Sicherheitshinweise

Sicherheitsbestimmungen

Beachten Sie die folgenden Sicherheitshinweise und Erklärungen!
Produktspezifische Sicherheitshinweise finden Sie auf den folgenden Seiten oder in den Bereichen Montage, Verdrahtung, Inbetriebnahme usw.

Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

Qualifikation des Personals

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen vertraut ist.

Erklärung der Symbole

In der vorliegenden Dokumentation werden die folgenden Symbole mit einem nebenstehenden Sicherheitshinweis oder Hinweistext verwendet. Die Sicherheitshinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

GEFAHR

Akute Verletzungsgefahr!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht unmittelbare Gefahr für Leben und Gesundheit von Personen!

WARNUNG

Verletzungsgefahr!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht Gefahr für Leben und Gesundheit von Personen!

VORSICHT

Schädigung von Personen!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, können Personen geschädigt werden!

HINWEIS

Schädigung von Umwelt oder Geräten

Wenn der Hinweis neben diesem Symbol nicht beachtet wird, können Umwelt oder Geräte geschädigt werden.



Tipp oder Fingerzeig

Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

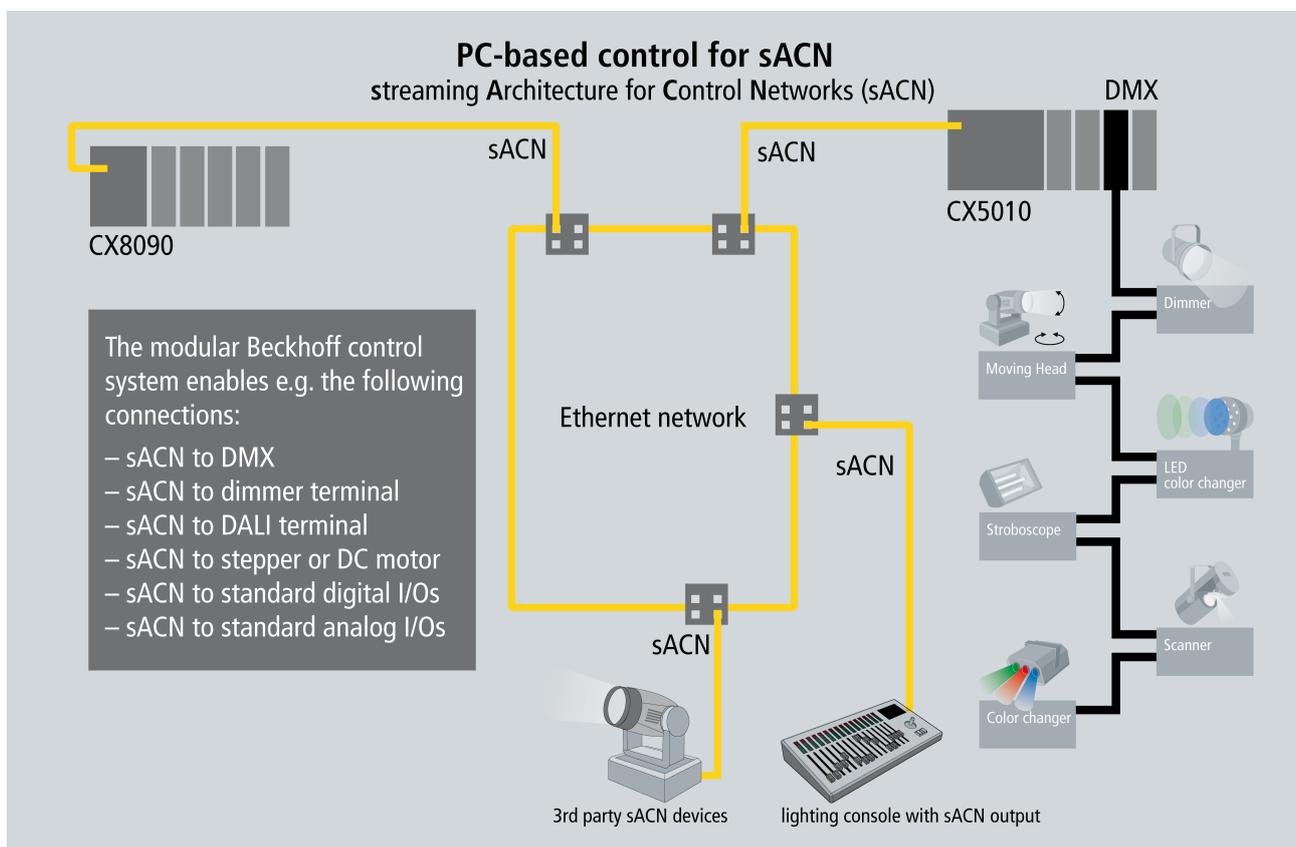
2 Übersicht

Hinweis Diese Dokumentation bezieht sich auf den ANSI Standard E1.31-2016 und die Verwendung von TwinCAT 3.

DMX ist in der Bühnen- und Gebäudetechnik ein weit verbreiteter Standard und dient zur Ansteuerung von Beleuchtungselementen. Beckhoff bietet hierzu die EL6851 DMX-Master-Klemme und die EL6851-0010 DMX-Slave-Klemme an.

Um die Vorteile von Ethernet für DMX zu nutzen, wurde von der ESTA (Entertainment Services and Technology Association) Streaming Architecture for Control Networks (im Folgenden sACN) entwickelt. Dieses überträgt die Daten des DMX-Universes über UDP/IP. Dabei werden die DMX-Daten als komplettes Universe in den sACN-Frame eingebettet und so durch UDP/IP getunnelt. sACN wird durch die ANSI-Norm E1.31-2009 (kurz sCN-2009) definiert und stellt eine Untermenge der ACN-Norm (ANSI-Norm E1.17) dar.

2016 hat die ESTA die Neufassung der sACN-Norm E1.31-2016 (kurz sACN-2016) eingeführt, mit der zwei neue Frame-Typen hinzugefügt wurden, der „Synchronization Frame“ und der „Universe Discovery Frame“. Der Synchronization Frame ist ein Paket, das nur Synchronisationsinformationen des Universe enthält, und wird zur Auslösung der Synchronisation verwendet. Der Universe Discovery Frame ist ein Paket, das eine gepackte Liste des Universe enthält, in dem eine Quelle aktiv ist. Dieser Frame ermöglicht anderen Geräten, die Interesse am Netzwerkverkehr haben, zu überwachen, welche Universes derzeit aktiv sind, ohne jeder Multicast-Gruppe beizutreten.



sACN-Daten können beliebigen Datentypen in TwinCAT zugeordnet werden.

sACN ist eine Protokollspezifikation für die Übertragung von DMX-Daten und wurde von der ESTA entwickelt.

sACN wird zu Art-Net™ vereinfacht und die Informationen über Art-Net™ befinden sich auch auf unserer Website.

Weitere Informationen über die Marktaktivitäten von Beckhoff im Marktsegment Stage and Show befinden sich auf unserer Website unter: PC-based Control für Stage- and Showtechnik.

Systemanforderungen:

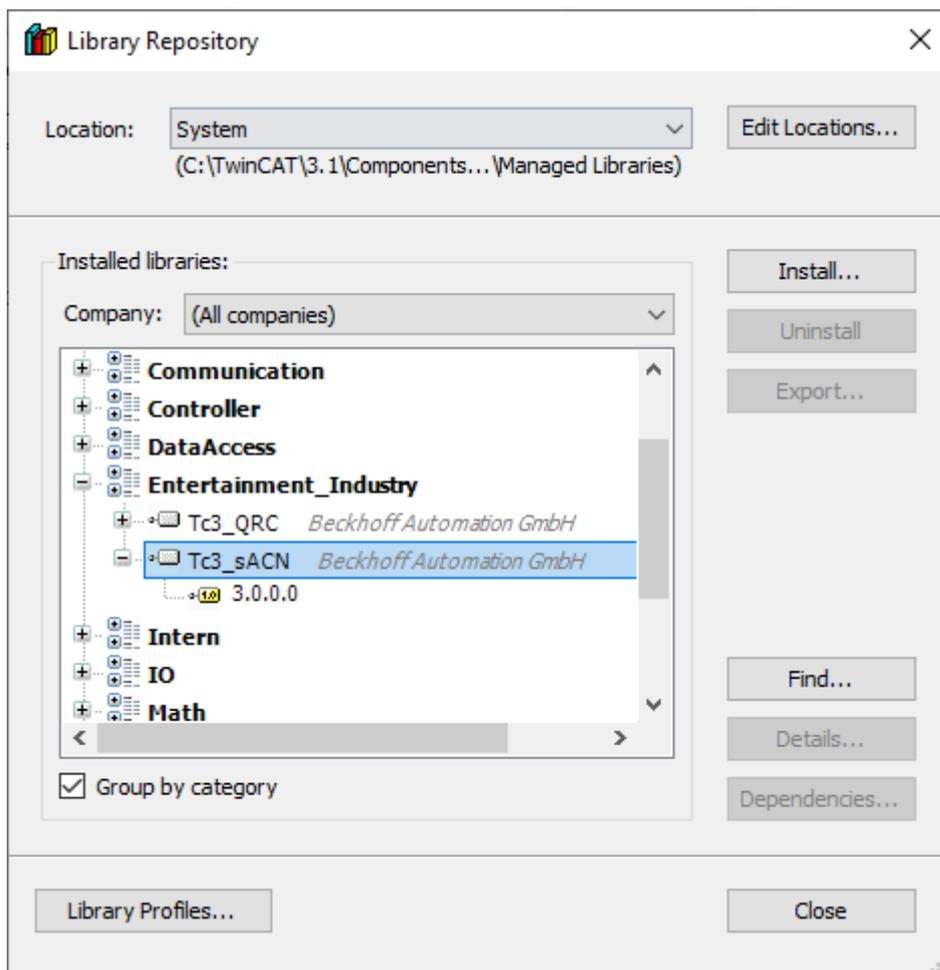
Technische Daten	Anforderung
TwinCAT-Version	TwinCAT 3.1 Build 4022.20 oder höher
Visual Studio-Version	Visual Studio 2013 oder höher
Erforderliche TwinCAT-Lizenz	TF6310-Lizenz

2.1 Update History

[Version 3.0.0.0] – 15.12.2020

Geändert:

- Hauptversionsnummer dieser Bibliothek wegen TwinCAT 3 in 3.x.x.x geändert.
- Verschiebung dieser Bibliotheksdatei in die Bibliothekskategorie „Entertainment_Industry“.



3 Programmierung

Dieses Beispiel besteht im Allgemeinen aus zwei Modulen, einem Codierungs-/Decodierungs-Modul und einem Kommunikationsmodul.

Beim Sendeprozess wird ein DMX-Frame vom Codierungs-/Decodierungs-Modul entsprechend der sACN-Struktur verpackt. Dann wird der verpackte sACN-Frame intern durch die Schnittstelle zum Kommunikationsmodul übertragen. Das Kommunikationsmodul sendet den sACN-Frame über UDP an den Ziel-Client.

Beim Empfangsprozess wird ein sACN-Frame empfangen und vom Kommunikationsmodul validiert und anschließend zum Codierungs-/Decodierungs-Baustein übertragen. Dieser Funktionsbaustein identifiziert den Typ des sACN-Frames und liest den DMX-Frame und die zugehörigen Informationen über den sendenden Server aus.

3.1 Funktionsbausteine

Die Funktionsbausteine `FB_sACN` und `FB_sACN_Ex` gehören zum Codierungs-/Decodierungsmodul. Das Kommunikationsmodul besteht aus einem Funktionsbaustein `FB_UDP_Connection`. Die zwei Schnittstellen sind `ICommunication` und `I_sACN`.

Der Funktionsbaustein `FB_sACN` implementiert sACN-2009 und der von `FB_sACN` abgeleitete Funktionsbaustein `FB_sACN_Ex` implementiert sACN-2016.

`FB_UDP_Connection` ermöglicht das Senden und Empfangen von Frames über UDP. Er implementiert die Schnittstelle `ICommunication` und sollte instanziiert und mit dem Funktionsbaustein `FB_sACN` oder `FB_sACN_Ex` verbunden werden.

i Gemäß der sACN-Spezifikation erfolgt die Übertragung des sACN-Frames in UDP-Paketen und in Zukunft kann der Frame vielleicht über andere Netzwerke übertragen werden, die von ACN unterstützt werden.

3.1.1 FB_sACN

Dieser Funktionsbaustein ermöglicht das Senden und Empfangen eines sACN-Daten-Frames. Mit Hilfe des Funktionsbausteins `FB_UDP_Communication` können Frames über UDP gesendet und empfangen werden.



i Da die Methoden `Send` und `Receive` asynchron sind, benötigen sie mehr als einen Zyklus, um abgeschlossen zu werden. Nur eine Methode könnte jeweils aufgerufen werden. Daher sollte beim Aufrufen dieser Methoden immer der Ausgabeparameter `bBusy` geprüft werden.

Syntax

```
FUNCTION_BLOCK FB_sACN IMPLEMENTS I_sACN
VAR_OUTPUT
    bBusy          : BOOL;
    bError         : BOOL;
    bCommunicated  : BOOL;
    ipResultMessage : I_TcMessage;
END_VAR
```

VAR_OUTPUT

bBusy: Wird gesetzt, wenn ein sACN-Frame gesendet/empfangen wird.

bError: Wird gesetzt, wenn ein Fehler aufgetreten ist. Fehlerdetails einschließlich der Fehler-ID befinden sich im Fenster „Error List“.

bCommunicated: Wird gesetzt, nachdem der erste sACN-Frame erfolgreich gesendet oder empfangen wurde. Weitere Informationen befinden sich im Abschnitt [Statusanzeige](#) [▶ 23].

ipResultMessage: Ermöglicht die Fehlerbehandlung mit dem Tc3_EventLogger.

METHODEN

FB_init: Initialisierungsmethode.

Send: Senden von sACN-Frames (sACN-2009).

Receive: Empfangen von sACN-Frames.

SCHNITTSTELLE

I_sACN: Definiert die Schnittstelle der Sende- und Empfangsmethoden.

3.1.1.1 FB_init

VAR_INPUT

```
Method FB_init : BOOL
VAR_INPUT
    ipCommunication      : ICommunication;
    aCID                 : ARRAY[1..16] OF BYTE;
END_VAR
```

ipCommunication: Kommunikationsbaustein, der die Schnittstelle `ICommunication` implementiert.

aCID: Komponentenidentifizierung. Eindeutige ID des Senders.

Beispiel:

Deklaration des Funktionsbausteins `FB_sACN`.

```
PROGRAM MAIN
VAR
    fbsACN      : FB_sACN(fbUDP, aCID);
    fbUDP       : FB_UDP_CONNECTION;
    aCID        : ARRAY[1..16] OF BYTE;
END_VAR
```

3.1.1.2 I_sACN

Der Funktionsbaustein `FB_sACN` und sein abgeleiteter Funktionsbaustein `FB_sACN_Ex` implementieren beide die Schnittstelle `I_sACN`. Mit ihrer Hilfe können diese zwei Funktionsbausteine nur online geschaltet werden, wenn `bBusy` gleich `FALSE` ist.

METHODEN

Send: Frames senden.

Receive: Frames empfangen.

Beispiel:

```
MAIN
VAR
    fbsACN      : FB_sACN(ipCommunication:= fbUDP, aCID := aCID);
    isACN       : I_sACN := fbsACN;
    fbsACNex    : FB_sACN_Ex(ipCommunication:= fbUDP, aCID:= aCID);
    fbUDP       : FB_UDP_Connection(sLocalHost:= '192.168.1.100', nLocalPort:= 200, sSrvNetId:=
'');
    aCID        : ARRAY[1..16] OF BYTE;
    bSwitch     : BOOL;
    bExtended   : BOOL;
```

```

    bBusy          : BOOL;
END_VAR

IF bSwitch AND NOT bBusy THEN
    bExtended := NOT bExtended;
    bSwitch := FALSE;
END_IF
IF bExtended THEN
    isACN := fbsACNex;
    bBusy := fbsACNex.bBusy OR fbUDP.bBusy;
ELSE
    isACN := fbsACN;
    bBusy := fbsACN.bBusy OR fbUDP.bBusy;
END_IF

```

3.1.1.2.1 Send

Diese Methode ermöglicht das Senden von UDP-Frames. Die Methode gibt TRUE zurück, wenn ein gültiger sACN-Frame gesendet wurde.

Syntax

```

Method Send: BOOL
VAR_INPUT
    sSourceName          : STRING;
    sRemoteHost          : T_IPv4Addr := '239.255.0.1';
    nDMXUniverse        : UINT := 1;
    nPriority             : USINT := 100;
    bPreviewData        : BOOL := FALSE;
    bStreamTerminated   : BOOL := FALSE;
    pDMXData            : POINTER TO BYTE;
    bForceSynchron      : BOOL := TRUE;
    nSynchronUniverse   : UINT := 0;
END_VAR

```

VAR_INPUT

sSourceName: Quellename.

sRemoteHost: IPv4-Zieladresse. Unicast- und Multicast-Adressen sind möglich.

nDMXUniverse: DMX-Ziel-Universum.

nPriority: Datenpriorität bei mehreren Quellen.

bPreviewData: Optionsbit: Preview_Data.

bStreamTerminated: Optionsbit: Stream_Terminated.

pDMXData: Hinweis auf DMX-Daten.

bForceSynchron: Neues Optionsbit von sACN-2016: Force_Synchronization.

nSynchronUniverse: Synchronisationsuniversum, ignoriert in sACN-2009.



1. Nach dem Auslösen der steigenden Flanke von `bStreamTerminated` sendet der Client weiterhin 3 sACN-Daten-Frames, die dieselben DMX-Daten, eine inkrementierte Sequenznummer und das Beendet-Flag enthalten. Nach dem Auftreten einer Zeitüberschreitung wird der geöffnete Socket geschlossen und anschließend der Sendeprozess vollständig beendet, sofern nicht die fallende Flanke von `bStreamTerminated` ausgelöst wird. (Weitere Einzelheiten siehe [hier \[► 23\]](#).)
2. `bForceSynchron` und `bSynchronUniverse` sind neue Funktionen, die in sACN-2016 definiert werden. Beim Funktionsbaustein `FB_sACN` werden diese beiden Variablen intern auf 0 gesetzt und andere Eingangswerte werden ignoriert.

3.1.1.2.2 Receive

Diese Methode ermöglicht das Empfangen eines UDP-Frames. Die Methode gibt TRUE zurück, wenn ein sACN-Frame empfangen wurde.

Syntax

```
Method Receive : BOOL
VAR_INPUT
    sTargetHost          : T_IPv4Addr := '239.255.0.1';
    nTargetUniverse      : UINT;
    sSourceName          : REFERENCE TO STRING(64);
    nDMXUniverse         : REFERENCE TO UINT;
    aDMXData             : REFERENCE TO ARRAY[1..512] OF BYTE;
    sSrcHost             : REFERENCE TO T_IPv4Addr;
    nSrcPort             : REFERENCE TO UDINT;
    bStreamTerminated   : REFERENCE TO BOOL;
    stFrameCounter       : REFERENCE TO ST_FrameCounter;
    nSynchronUniverse    : REFERENCE TO UINT;
    sDiscoveredUniverse  : REFERENCE TO T_MaxString;
END_VAR
```

VAR_INPUT

sTargetHost: IPv4-Zieladresse, von der sACN-Frames empfangen werden. Unicast- und Multicast-Adressen sind möglich.

nTargetUniverse: Ziel-Universe, von dem der sACN-Frame empfangen wird.

REFERENCE TO ... (OUTPUT)

sSourceName: Quellname der empfangenen sACN-Daten-Frames.

nDMXUniverse: DMX-Universe der empfangenen sACN-Daten-Frames.

aDMXData: Empfangene DMX-Daten.

sSrcHost: IPv4-Adresse des Remote-Geräts, von dem die sACN-Frames empfangen wurden.

nSrcPort: IPv4-Portnummer des Remote-Geräts, von dem die sACN-Frames empfangen wurden.

bStreamTerminated: Ist gleich TRUE, wenn die Datenquelle für das Universe die Übertragung beendet hat. In der Folge empfängt dieser Client keine sACN-Frames mehr.

stFrameCounter: Zähler der empfangenen sACN-Frames.

nSynchronUniverse: Synchronisations-Universe der empfangenen sACN-Daten-Frames.

sDiscoveredUniverse: Eine Liste der Universes, von denen sACN-Daten- und Synchronisations-Frames innerhalb von 10 s übertragen wurden.



1. **sDiscoveredUniverse** zeigt, welche Universes derzeit aktiv sind. Es wird aktualisiert, nachdem ein neuer Universe Discovery Frame angekommen ist. Es zeigt nichts (leerer String) bei **FB_sACN**.

2. Der Universe Discovery Frame und der Synchronization Frame sind neue Typen von sACN-Frames, die in sACN-2016 definiert werden. Diese beiden Frame-Typen können vom Zähler von **FB_sACN** (sACN-2009) nicht erkannt werden. Diese Frames können zwar empfangen werden, werden aber als „unbekannte“ Frames gezählt.

3.1.2 FB_sACN_Ex

Dieser Funktionsbaustein ist von **FB_sACN** abgeleitet und implementiert sACN-2016.



i Da die Methoden `Send`, `Receive` und `SendSynchronizationFrame` asynchron sind und mehr als einen Zyklus benötigen, um abgeschlossen zu werden, könnte nur jeweils eine Methode aufgerufen werden. Daher sollte beim Aufrufen dieser Methoden immer der Ausgabeparameter `bBusy` geprüft werden.

Syntax

```
FUNCTION_BLOCK FB_sACN_Ex EXTENDS FB_sACN
VAR_OUTPUT
    bBusy          : BOOL;
    bError         : BOOL;
    bCommunicated  : BOOL;
    ipResultMessage : I_TcMessage;
END_VAR
```

VAR_OUTPUT

bBusy: Wird gesetzt, wenn ein gültiger sACN-Frame gesendet/empfangen wird.

bError: Wird gesetzt, wenn ein Fehler aufgetreten ist. Fehlerdetails einschließlich der Fehler-ID befinden sich im protokollierten Ereignis.

bCommunicated: Wird gesetzt, nachdem der erste Streaming-sACN-Frame erfolgreich gesendet oder empfangen wurde. Weitere Informationen befinden sich im Abschnitt [Statusanzeige](#) | 23].

ipResultMessage: Ermöglicht die Fehlerbehandlung mit dem `Tc3_EventLogger`.

METHODEN

FB_init: Initialisierungsmethode

Send: Senden von sACN-Frames (sACN-2016).

Receive: Empfangen von sACN-Frames.

SendSynchronizationFrame: sACN-Synchronisations-Frame senden.

3.1.2.1 FB_init

```
Method FB_init : BOOL
VAR_INPUT
    ipCommunication : ICommunication;
    aCID             : ARRAY[1..16] OF BYTE;
END_VAR
```

VAR_INPUT

ipCommunication: Kommunikationsbaustein, der die Schnittstelle `ICommunication` implementiert.

aCID: Komponentenidentifizierung. Eindeutige ID des Senders.

Beispiel:

Deklaration des Funktionsbausteins `FB_sACN_Ex`:

```
PROGRAM MAIN
VAR
    fbsACNex : FB_sACN_Ex(fbUDP, aCID);
    fbUDP    : FB_UDP_CONNECTION;
    aCID     : ARRAY[1..16] OF BYTE;
END_VAR
```

3.1.2.2 SendSynchronizationFrame

Diese Methode ermöglicht das Senden eines sACN-Synchronisations-Frames, der zur Auslösung der Synchronisation verwendet wird.

- Diese Methode gibt TRUE aus, sobald die Anforderung zum Senden eines sACN-Synchronisations-Frames beendet ist.
- Diese Methode gibt FALSE aus, wenn die asynchrone Anforderung noch aktiv ist. Die Methode muss aufgerufen werden, bis sie erfolgreich ausgeführt wurde und der Rückgabewert gleich TRUE ist.

```
Method SendSynchronizationFrame : BOOL
VAR_INPUT
    nSynchronUniverse          : UINT := 0;
END_VAR
```

VAR_INPUT

nSynchronUniverse: Universe, von dem Synchronisations-Frames übertragen werden.



Diese Methode kann nur aufgerufen werden, wenn `bBusy` gleich FALSE ist.

3.1.2.3 I_sACN

Der Funktionsbaustein `FB_sACN` und sein abgeleiteter Funktionsbaustein `FB_sACN_Ex` implementieren beide die Schnittstelle `I_sACN`. Mit ihrer Hilfe können diese zwei Funktionsbausteine nur online geschaltet werden, wenn `bBusy` gleich FALSE ist.

METHODEN

Send: Frames senden.

Receive: Frames empfangen.

Beispiel:

```
MAIN
VAR
    fbsACN          : FB_sACN(ipCommunication:= fbUDP, aCID := aCID);
    isACN           : I_sACN := fbsACN;
    fbsACNex       : FB_sACN_Ex(ipCommunication:= fbUDP, aCID:= aCID);
    fbUDP          : FB_UDP_Connection(sLocalHost:= '192.168.1.100', nLocalPort:= 200, sSrvNetId:=
''');
    aCID           : ARRAY[1..16] OF BYTE;
    bSwitch        : BOOL;
    bExtended      : BOOL;
    bBusy          : BOOL;
END_VAR
```

```
IF bSwitch AND NOT bBusy THEN
    bExtended := NOT bExtended;
    bSwitch := FALSE;
END_IF
IF bExtended THEN
    isACN := fbsACNex;
    bBusy := fbsACNex.bBusy OR fbUDP.bBusy;
ELSE
    isACN := fbsACN;
    bBusy := fbsACN.bBusy OR fbUDP.bBusy;
END_IF
```

3.1.2.3.1 Send

Diese Methode ermöglicht das Senden von UDP-Frames. Die Methode gibt TRUE zurück, wenn ein gültiger sACN-Frame gesendet wurde.

Syntax

```

Method Send: BOOL
VAR_INPUT
    sSourceName          : STRING;
    sRemoteHost          : T_IPv4Addr := '239.255.0.1';
    nDMXUniverse         : UINT := 1;
    nPriority             : USINT := 100;
    bPreviewData         : BOOL := FALSE;
    bStreamTerminated    : BOOL := FALSE;
    pDMXData             : POINTER TO BYTE;
    bForceSynchron       : BOOL := TRUE;
    nSynchronUniverse    : UINT := 0;
END_VAR

```

VAR_INPUT

sSourceName: Quellename.

sRemoteHost: IPv4-Zieladresse. Unicast- und Multicast-Adressen sind möglich.

nDMXUniverse: DMX-Ziel-Universe.

nPriority: Datenpriorität bei mehreren Quellen.

bPreviewData: Optionsbit: Preview_Data.

bStreamTerminated: Optionsbit: Stream_Terminated.

pDMXData: Hinweis auf DMX-Daten.

bForceSynchron: Neues Optionsbit von sACN-2016: Force_Synchronization.

nSynchronUniverse: Synchronisationsuniverse, ignoriert in sACN-2009.



1. Nach dem Auslösen der steigenden Flanke von `bStreamTerminated` sendet der Client weiterhin 3 sACN-Daten-Frames, die dieselben DMX-Daten, eine inkrementierte Sequenznummer und das Beendet-Flag enthalten. Nach dem Auftreten einer Zeitüberschreitung wird der geöffnete Socket geschlossen und anschließend der Sendeprozess vollständig beendet, sofern nicht die fallende Flanke von `bStreamTerminated` ausgelöst wird. (Weitere Einzelheiten siehe [hier](#) [► 23].)

2. `bForceSynchron` und `bSynchronUniverse` sind neue Funktionen, die in sACN-2016 definiert werden. Beim Funktionsbaustein `FB_sACN` werden diese beiden Variablen intern auf 0 gesetzt und andere Eingangswerte werden ignoriert.

3.1.2.3.2 Receive

Diese Methode ermöglicht das Empfangen eines UDP-Frames. Die Methode gibt TRUE zurück, wenn ein sACN-Frame empfangen wurde.

Syntax

```

Method Receive : BOOL
VAR_INPUT
    sTargetHost          : T_IPv4Addr := '239.255.0.1';
    nTargetUniverse      : UINT;
    sSourceName          : REFERENCE TO STRING(64);
    nDMXUniverse         : REFERENCE TO UINT;
    aDMXData             : REFERENCE TO ARRAY[1..512] OF BYTE;
    sSrcHost             : REFERENCE TO T_IPv4Addr;
    nSrcPort             : REFERENCE TO UDINT;
    bStreamTerminated    : REFERENCE TO BOOL;
    stFrameCounter       : REFERENCE TO ST_FrameCounter;
    nSynchronUniverse    : REFERENCE TO UINT;
    sDiscoveredUniverse  : REFERENCE TO T_MaxString;
END_VAR

```

VAR_INPUT

sTargetHost: IPv4-Zieladresse, von der sACN-Frames empfangen werden. Unicast- und Multicast-Adressen sind möglich.

nTargetUniverse: Ziel-Universe, von dem der sACN-Frame empfangen wird.

REFERENCE TO ... (OUTPUT)

- sSourceName:** Quellname der empfangenen sACN-Daten-Frames.
- nDMXUniverse:** DMX-Universes der empfangenen sACN-Daten-Frames.
- aDMXData:** Empfangene DMX-Daten.
- sSrcHost:** IPv4-Adresse des Remote-Geräts, von dem die sACN-Frames empfangen wurden.
- nSrcPort:** IPv4-Portnummer des Remote-Geräts, von dem die sACN-Frames empfangen wurden.
- bStreamTerminated:** Ist gleich TRUE, wenn die Datenquelle für das Universe die Übertragung beendet hat. In der Folge empfängt dieser Client keine sACN-Frames mehr.
- stFrameCounter:** Zähler der empfangenen sACN-Frames.
- nSynchronUniverse:** Synchronisations-Universes der empfangenen sACN-Daten-Frames.
- sDiscoveredUniverse:** Eine Liste der Universes, von denen sACN-Daten- und Synchronisations-Frames innerhalb von 10 s übertragen wurden.

- i** 1. `sDiscoveredUniverse` zeigt, welche Universes derzeit aktiv sind. Es wird aktualisiert, nachdem ein neuer Universe Discovery Frame angekommen ist. Es zeigt nichts (leerer String) bei `FB_sACN`.
- 2. Der Universe Discovery Frame und der Synchronization Frame sind neue Typen von sACN-Frames, die in sACN-2016 definiert werden. Diese beiden Frame-Typen können vom Zähler von `FB_sACN` (sACN-2009) nicht erkannt werden. Diese Frames können zwar empfangen werden, werden aber als „unbekannte“ Frames gezählt.

3.1.3 FB_UDP_Connection

Dieser Funktionsbaustein ermöglicht die Herstellung oder Beendigung einer UDP-Verbindung.



- i** Dieser Funktionsbaustein kann nur als Anfangsvariable der Funktionsbausteine `FB_sACN` und `FB_sACN_Ex` verwendet werden. Er kann nicht direkt aufgerufen werden.

Syntax

```
FUNCTION_BLOCK FB_UDP_Connection IMPLEMENTS ICommunication
VAR_OUTPUT
bBusy           : BOOL;
bError          : BOOL;
bConnected      : BOOL;
ipResultMessage : I_TcMessage;
END_VAR
```

VAR_OUTPUT

- bBusy:** Ist gleich TRUE, solange die asynchrone Anforderung noch aktiv ist. Ist gleich FALSE, wenn die Anforderung abgeschlossen wurde. (Weitere Informationen befinden sich im Abschnitt [Statusanzeige](#) [► 23])
- bError:** Wird gesetzt, wenn bei der Ausführung des Funktionsbausteins ein Fehler auftritt. Fehlerdetails befinden sich im Fenster „Error List“.
- bConnected:** Ist gleich TRUE, solange ein Socket ohne Fehler geöffnet ist; ist gleich FALSE, wenn der Socket geschlossen ist.

ipResultMessage: Ermöglicht die Fehlerbehandlung mit dem `Tc3_EventLogger`.

METHODEN

FB_init: Initialisierungsmethode

SCHNITTSTELLE

ICommunication: Definiert die Schnittstelle der Kommunikationsmethoden.

3.1.3.1 FB_init

```
Method FB_init : BOOL
VAR_INPUT
    sLocalHost      : T_IPv4Addr;
    nLocalPort      : UINT := 200;
    sSrvNetId       : T_AmsNetId;
END_VAR
```

VAR_INPUT

sLocalHost: Die lokale IP-Adresse (IPv4) des UDP-Sockets als String

nLocalPort: Die lokale IP-Portnummer des UDP-Sockets. Der Standardwert ist 200.

sSrvNetId: AMS Net Id. Für den lokalen Rechner (Default) kann auch ein Leerstring angegeben werden.



Die Variable `nLocalPort` ist nur für den Socket, der für den Sendeprozess verwendet wird. Während des Empfangsprozesses muss die Portnummer 5568 sein, andere Werte werden ignoriert.

Beispiel:

Deklaration des Funktionsbausteins `FB_UDP_CONNECTION`:

```
PROGRAM MAIN
VAR
    fbUDP      : FB_UDP_CONNECTION (sLocalHost:= '192.168.1.100', nLocalPort:= 200, sSrvNetId:= '
');
END_VAR
```

3.1.3.2 ICommunication

Diese Schnittstelle definiert eine Kommunikationsschnittstelle, um den Datenaustausch zwischen zwei Funktionsbausteinen zu ermöglichen. Diese Schnittstelle wurde intern implementiert, d. h. die Benutzer müssen sie nicht implementieren.

3.2 Strukturen, Schnittstellen

3.2.1 Strukturen

3.2.1.1 ST_FrameCounter

```
TYPE ST_FrameCounter:
STRUCT
    nDMX_Frames      : UDINT;
    nSynchronization_Frames : UDINT;
    nUniverse_Discovery_Frames : UDINT;
    nUnknown_sACN_Frames : UDINT;
END_STRUCT
END_TYPE
```



Der Universe Discovery Frame und der Synchronization Frame sind neue Typen von sACN-Frames, die in sACN-2016 definiert werden. Sie können zwar auch von FB_sACN (sACN-2009) empfangen werden, sind aber als „unbekannte“ Frames gezählt.

3.2.2 Schnittstellen

3.2.2.1 ICommunication

Diese Schnittstelle definiert eine Kommunikationsschnittstelle, um den Datenaustausch zwischen zwei Funktionsbausteinen zu ermöglichen. Diese Schnittstelle wurde intern implementiert, d. h. die Benutzer müssen sie nicht implementieren.

3.2.2.2 I_sACN

Der Funktionsbaustein FB_sACN und sein abgeleiteter Funktionsbaustein FB_sACN_Ex implementieren beide die Schnittstelle I_sACN. Mit ihrer Hilfe können diese zwei Funktionsbausteine nur online geschaltet werden, wenn bBusy gleich FALSE ist.

METHODEN

Send: Frames senden.

Receive: Frames empfangen.

Beispiel:

```

MAIN
VAR
    fbsACN      : FB_sACN(ipCommunication:= fbUDP, aCID := aCID);
    isACN       : I_sACN := fbsACN;
    fbsACNex    : FB_sACN_Ex(ipCommunication:= fbUDP, aCID:= aCID);
    fbUDP       : FB_UDP_Connection(sLocalHost:= '192.168.1.100', nLocalPort:= 200, sSrvNetId:=
''');
    aCID        : ARRAY[1..16] OF BYTE;
    bSwitch     : BOOL;
    bExtended   : BOOL;
    bBusy       : BOOL;
END_VAR

IF bSwitch AND NOT bBusy THEN
    bExtended := NOT bExtended;
    bSwitch := FALSE;
END_IF
IF bExtended THEN
    isACN := fbsACNex;
    bBusy := fbsACNex.bBusy OR fbUDP.bBusy;
ELSE
    isACN := fbsACN;
    bBusy := fbsACN.bBusy OR fbUDP.bBusy;
END_IF
    
```

3.2.2.2.1 Send

Diese Methode ermöglicht das Senden von UDP-Frames. Die Methode gibt TRUE zurück, wenn ein gültiger sACN-Frame gesendet wurde.

Syntax

```

Method Send: BOOL
VAR_INPUT
    sSourceName      : STRING;
    sRemoteHost      : T_IPv4Addr := '239.255.0.1';
    nDMXUniverse     : UINT := 1;
    nPriority         : USINT := 100;
    bPreviewData     : BOOL := FALSE;
    
```

```

bStreamTerminated      : BOOL := FALSE;
pDMXData               : POINTER TO BYTE;
bForceSynchron         : BOOL := TRUE;
nSynchronUniverse      : UINT := 0;
END_VAR

```

VAR_INPUT

sSourceName: Quellename.

sRemoteHost: IPv4-Zieladresse. Unicast- und Multicast-Adressen sind möglich.

nDMXUniverse: DMX-Ziel-Universum.

nPriority: Datenpriorität bei mehreren Quellen.

bPreviewData: Optionsbit: Preview_Data.

bStreamTerminated: Optionsbit: Stream_Terminated.

pDMXData: Hinweis auf DMX-Daten.

bForceSynchron: Neues Optionsbit von sACN-2016: Force_Synchronisation.

nSynchronUniverse: Synchronisationsuniversum, ignoriert in sACN-2009.



1. Nach dem Auslösen der steigenden Flanke von `bStreamTerminated` sendet der Client weiterhin 3 sACN-Daten-Frames, die dieselben DMX-Daten, eine inkrementierte Sequenznummer und das Beendet-Flag enthalten. Nach dem Auftreten einer Zeitüberschreitung wird der geöffnete Socket geschlossen und anschließend der Sendeprozess vollständig beendet, sofern nicht die fallende Flanke von `bStreamTerminated` ausgelöst wird. (Weitere Einzelheiten siehe [hier \[23\]](#).)

2. `bForceSynchron` und `bSynchronUniverse` sind neue Funktionen, die in sACN-2016 definiert werden. Beim Funktionsbaustein `FB_sACN` werden diese beiden Variablen intern auf 0 gesetzt und andere Eingangswerte werden ignoriert.

3.2.2.2.2 Receive

Diese Methode ermöglicht das Empfangen eines UDP-Frames. Die Methode gibt TRUE zurück, wenn ein sACN-Frame empfangen wurde.

Syntax

```

Method Receive : BOOL
VAR_INPUT
  sTargetHost          : T_IPv4Addr := '239.255.0.1';
  nTargetUniverse      : UINT;
  sSourceName          : REFERENCE TO STRING(64);
  nDMXUniverse         : REFERENCE TO UINT;
  aDMXData             : REFERENCE TO ARRAY[1..512] OF BYTE;
  sSrcHost             : REFERENCE TO T_IPv4Addr;
  nSrcPort             : REFERENCE TO UDINT;
  bStreamTerminated    : REFERENCE TO BOOL;
  stFrameCounter       : REFERENCE TO ST_FrameCounter;
  nSynchronUniverse    : REFERENCE TO UINT;
  sDiscoveredUniverse  : REFERENCE TO T_MaxString;
END_VAR

```

VAR_INPUT

sTargetHost: IPv4-Zieladresse, von der sACN-Frames empfangen werden. Unicast- und Multicast-Adressen sind möglich.

nTargetUniverse: Ziel-Universum, von dem der sACN-Frame empfangen wird.

REFERENCE TO ... (OUTPUT)

sSourceName: Quellname der empfangenen sACN-Daten-Frames.

nDMXUniverse: DMX-Universum der empfangenen sACN-Daten-Frames.

aDMXData: Empfangene DMX-Daten.

sSrcHost: IPv4-Adresse des Remote-Geräts, von dem die sACN-Frames empfangen wurden.

nSrcPort: IPv4-Portnummer des Remote-Geräts, von dem die sACN-Frames empfangen wurden.

bStreamTerminated: Ist gleich TRUE, wenn die Datenquelle für das Universe die Übertragung beendet hat. In der Folge empfängt dieser Client keine sACN-Frames mehr.

stFrameCounter: Zähler der empfangenen sACN-Frames.

nSynchronUniverse: Synchronisations-Universe der empfangenen sACN-Daten-Frames.

sDiscoveredUniverse: Eine Liste der Universes, von denen sACN-Daten- und Synchronisations-Frames innerhalb von 10 s übertragen wurden.



1. `sDiscoveredUniverse` zeigt, welche Universes derzeit aktiv sind. Es wird aktualisiert, nachdem ein neuer Universe Discovery Frame angekommen ist. Es zeigt nichts (leerer String) bei `FB_sACN`.

2. Der Universe Discovery Frame und der Synchronization Frame sind neue Typen von sACN-Frames, die in sACN-2016 definiert werden. Diese beiden Frame-Typen können vom Zähler von `FB_sACN` (sACN-2009) nicht erkannt werden. Diese Frames können zwar empfangen werden, werden aber als „unbekannte“ Frames gezählt.

4 Beispiel

Dieses <https://infosys.beckhoff.com/content/1031/sACN/Resources/zip/9007207188597515.zip> beinhaltet ein SPS-Programm für TwinCAT 3, das das allgemeine Senden und Empfangen von sACN-Frames zeigt. In jedem Anwendungsfall können die Benutzerdaten des sACN-Frames mit beliebigen SPS-Variablen verbunden werden. Dies ermöglicht eine sehr flexible Nutzung dieses Beispiels. Die Benutzer können ihre eigenen Programme entsprechend ihren Bedürfnissen erstellen. Die einzelnen Funktionsbausteine des Beispiels werden im Abschnitt Programmierung erläutert.

Beispielsweise kann eine Zustandsmaschine für die Realisierung einer Logik verwendet werden, bei der grundsätzlich das „Zuhören“ in der Methode Empfangen verwendet wird, gefolgt von der Methode Senden bei einem speziellen Ereignis oder Timer. Alternativ könnten, wenn die Methode Senden und die Methode Empfangen gleichzeitig ausgeführt werden sollten, zwei Instanzen deklariert werden: eine Instanz zum Senden und die andere zum Empfangen.

Zusätzlich ist ein TwinCAT 2-Beispiel verfügbar, das nur die sACN-Version 2009 unterstützt.

In diesem TwinCAT 3-Beispiel sind beide sACN-Versionen – „sACN 2009“ und „sACN 2016“ – implementiert, damit Benutzer die Version „sACN 2009“ auf der TwinCAT 3-Plattform verwenden können, wenn sie dies möchten. Grundsätzlich geht es bei diesem Beispiel um sACN-Anwendungen, die die Version sACN 2016 verwenden. Es wird jedoch auch darin gezeigt, wie beide sACN-Versionen online umgeschaltet werden können. Durch Instanzieren der Schnittstelle „isACN“ und Auslösen der Variable „bSwitch“ ist es möglich, zwischen diesen beiden sACN-Versionen online umzuschalten.

The screenshot displays the Beckhoff TwinCAT IDE interface. At the top, the menu bar includes File, Edit, View, Project, Build, Debug, TwinCAT, TwinCAT HMI, TwinSAFE, PLC, Team, Scope, Tools, Window, and Help. The toolbar shows various icons for file operations and execution. The main window is titled 'TC3_sACN - TcXaeShell' and shows the 'MAIN [Online]' project.

Below the toolbar, there is a table of variable declarations for the 'TC3_sACN.TC3_sACN.MAIN' program:

Expression	Type	Value	Prepared value	Address	Comment
fbsACN	FB_sACN				
isACN	I_sACN	16#FFFFB08F81...			
fbsACNex	FB_sACN_Ex				
fbUDP	FB_UDP_Connection				
aCID	ARRAY [1..16] OF B...				
bExtended	BOOL	TRUE			
aDMXData	ARRAY [1..512] OF ...				
sSourceName	STRING	"			Reference t
nDMXUniverse	UINT	1			Reference t
nSynchronUniverse	UINT	0			Reference t
sSrcHost	T_IPv4Addr	"			Reference t
nSrcPort	UDINT	0			Reference t
stFrameCounter	CT_FrameCounter				

Below the table, the ladder logic code is displayed. The code includes comments and logic for switching sACN version, sending and receiving data, and synchronization. Key lines include:

```

1 //-----Switch sACN version-----
2 IF bChange FALSE AND NOT fbUDP.bBusy FALSE THEN // bBusy = FALSE => allowed to switch sACN FBs
3   bExtended TRUE := NOT bExtended TRUE;
4   bChange FALSE := FALSE;
5 END_IF
6 IF bExtended TRUE THEN
7   isACN := fbsACNex;
8   bError FALSE := fbsACNex.bError FALSE OR fbUDP.bError FALSE;
9   bCommunicated TRUE := fbsACNex.bCommunicated TRUE AND fbUDP.bConnected TRUE;
10 ELSE
11   isACN := fbsACN;
12   bError FALSE := fbsACN.bError FALSE OR fbUDP.bError FALSE;
13   bCommunicated TRUE := fbsACN.bCommunicated FALSE AND fbUDP.bConnected TRUE;
14 END_IF
15 //-----Send-----
16 IF bSend TRUE AND NOT bReceive FALSE THEN
17   IF isACN.Send(sSourceName:= 'BeckhoffTEst110', sRemoteHost:= '239.255.0.1',
18     nDMXUniverse:= 2, nPriority:= 100, bPreviewData:= FALSE,
19     bStreamTerminated:= bEnd FALSE, bForceSynchron:= TRUE,
20     nSynchronUniverse:= nSynchronUniverse 0, pDMXData:= ADR(aDMXData)) AND NOT bError FALSE THEN
21     aDMXData[1][233] := aDMXData[1][233]+1;
22   END_IF
23 END_IF
24 //-----Receive-----
25 IF NOT bSend TRUE AND bReceive FALSE THEN
26   isACN.Receive(sTargetHost:= '239.255.0.2', nTargetUniverse:= nTargetUniverse 0,
27     sSourceName:= sSourceName , nDMXUniverse:= nDMXUniverse 1,
28     nSynchronUniverse:= nSynchronUniverse 0, aDMXData:= aDMXData,
29     sSrcHost:= sSrcHost , nSrcPort:= nSrcPort 0, bStreamTerminated := bEnd FALSE,
30     stFrameCounter:= stFrameCounter, sDiscoveredUniverse:= sDiscoveredString );
31 END_IF
32 //-----Send Synchronization Frame-----
33 IF bSynchronSend FALSE AND bExtended TRUE THEN
34   IF fbsACNex.SendSynchronizationFrame(6751) THEN
35     bSynchronSend FALSE := FALSE;
36   END_IF;
37 END_IF RETURN
    
```

The bottom status bar shows 'Ready', 'Ln 16', 'Col 9', 'Ch 9', 'INS', and 'Add to Source Control'.

5 Anhang

5.1 Eigenschaften in Bezug auf Zeit

In diesem Projekt werden einige Zeiteigenschaften intern definiert und können von den Benutzern nicht geändert werden. Im Folgenden werden diese Zeiteigenschaften und die zugehörige Ablauflogik erläutert.

Sendeprozess:

- Gemäß sACN-2016 wird der Universe Discovery Frame alle **10 s** gesendet.
- Nachdem Daten-Frames mit dem **Beenden-Flag** gesendet wurden, bleibt der Socket **2,5 s** geöffnet, um während dieser Zeit weiter sACN-Frames von anderen Quellen zu empfangen. Nach Ablauf der Zeit wird dieser Socket geschlossen und `bCommunicated` und `bConnected` werden wieder auf FALSE gesetzt. Der Socket kann nur wieder geöffnet werden, wenn eine fallende Flanke der Beenden-Variable `bStreamTerminated` auftritt.

Empfangsprozess:

- Gemäß der sACN-Spezifikation wird der Empfangsprozess beendet und wechselt der Client in den Ruhezustand, wenn in einer Zeit von **2,5 s** keine neuen sACN-Daten-Frames empfangen wurden. Der Socket bleibt geöffnet, wenn während der Ruhezeit kein Fehler auftritt.
- Wenn während des Ruhezustands kein Fehler auftritt, beginnt der Empfangsprozess nach **5 s** von neuem.

5.2 Statusanzeige

`bCommunicated`

Dies ist eine Ausgangsvariable von `FB_sACN` und `FB_sACN_ex`. Sie zeigt den Status des Sende-/Empfangsprozesses an.

- Wird auf TRUE gesetzt, wenn ein sACN-Frame erfolgreich gesendet oder empfangen wurde.
- Wird auf FALSE gesetzt, wenn das Beendet-Flag abgesendet wurde. Der Client geht in den Ruhestatus über, wenn kein Fehler auftritt.

`bConnected`

Dies ist eine Ausgangsvariable des Funktionsbausteins `FB_UDP_Connection`. Sie zeigt den Verbindungsstatus an.

- Wird auf TRUE gesetzt, wenn ein UDP-Socket ohne einen Fehler erfolgreich geöffnet wurde.
- Wird auf FALSE gesetzt, wenn ein Socket geschlossen wurde oder ein Fehler aufgetreten ist.

`bBusy`

Dies ist eine Ausgangsvariable aller Funktionsbausteine.

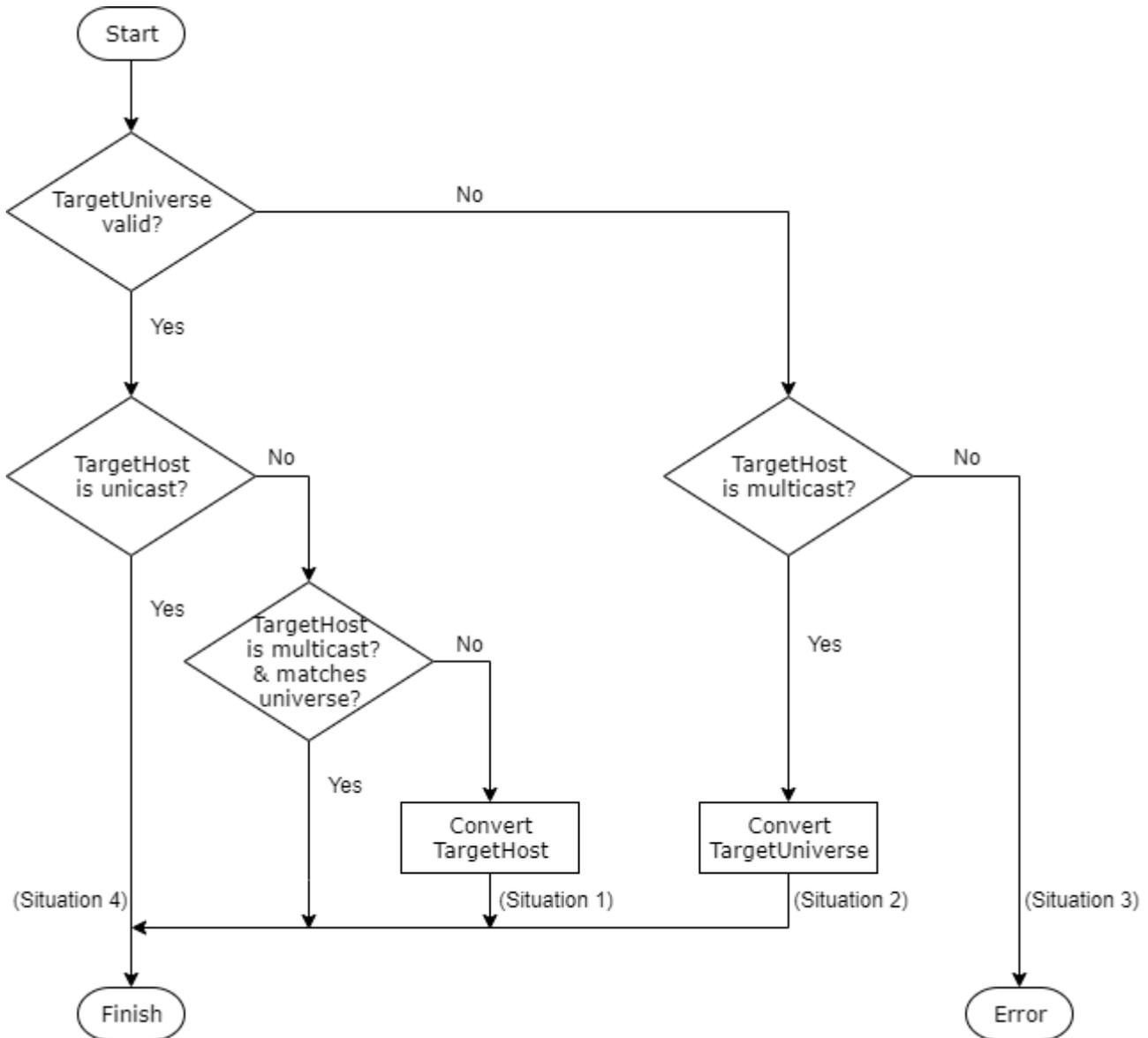
- Wird auf TRUE gesetzt, solange die Verarbeitung nicht abgeschlossen ist. Jede neue Eingangsvariable von Methoden oder das Wechseln von sACN-Funktionsbausteinen kann NUR akzeptiert werden, wenn `bBusy` gleich FALSE ist.
- Wird auf FALSE gesetzt, solange die Verarbeitung ohne Fehler abgeschlossen ist (`bError` ist gleich FALSE) oder eine Verarbeitung nicht abgeschlossen ist, aber ein Fehler auftritt (`bError` ist gleich TRUE).

`bError`

Wird auf TRUE gesetzt, wenn ein Fehler auftritt.

5.3 Zielhost und Ziel-Universse

Die Zielhost-Variablen `sTargetHost` und die Ziel-Universse-Variablen `nTargetUniverse` sind die IPv4-Adresse und der Port, an die die sACN-Frames gesendet werden und von denen sie kommen.



Text

Gemäß der sACN-Spezifikation werden im Multicast-Modus die letzten zwei Bytes einer Multicast-Adresse durch das High-Byte und Low-Byte des Universe definiert. Passt der Zielhost nicht zum Ziel-Universse, wird das Universse dominiert. In diesem Fall muss die Multicast-Adresse korrigiert werden. In der Folge wird die ursprüngliche Multicast-Adresse ignoriert (Situation 1).

Wenn das Ziel-Universse jedoch ungültig ist ($\text{Universe} > 63999$ oder $\text{Universe} = 0$), aber der Zielhost eine Multicast-Adresse ist, sollte das Universse ausgehend vom Zielhost konvertiert werden, woraufhin das ungültige Ziel-Universse ignoriert wird (Situation 2).

Insbesondere wenn beide Variablen ungültig sind oder der Zielhost eine Unicast-Adresse ist, während das Ziel-Universse ungültig ist, tritt ein Fehler auf, und die Fehlerinformationen befinden sich im Fenster „Error List“ (Situation 3).

Wenn das Ziel-Universse gültig ist und der Zielhost eine Unicast-Adresse ist, sind diese Regeln nicht von Belang (Situation 4).

Hier folgen einige Beispiele:

Beispiel für Situation 1: Der Zielhost ist **239.255.0.1** und das Ziel-Universum ist **2**. Da das Ziel-Universum gültig und der Zielhost eine Multicast-Adresse ist, wird der Zielhost in **239.255.0.2** geändert.

Beispiel für Situation 2: Der Zielhost ist **239.255.0.1** und das Ziel-Universum ist **0**. Da das Ziel-Universum ungültig und der Zielhost eine Multicast-Adresse ist, wird das Ziel-Universum in **1** geändert.

Beispiel für Situation 3: Der Zielhost ist **0.0.0.0** oder **192.168.0.1** und das Ziel-Universum ist **0**. Ein Fehler tritt auf.

Beispiel für Situation 4: Der Zielhost ist **192.168.0.100** und das Ziel-Universum ist **1**. Beide Werte werden ohne Konvertierung akzeptiert.

5.4 Multicast-Gruppe

Um sACN-Frames von verschiedenen Multicast-Gruppen (kurz MCGs) zu empfangen, wird in diesem Projekt der Beitritt zu mehreren MCGs unterstützt. Die Benutzer könnten entweder die neue MCG-Adresse am Eingang `sTargetHost` angeben oder ein zugehöriges neues Universum am Eingang `nTargetUniverse` schreiben. Alle beigetretenen MCGs werden gespeichert und die Benutzer müssen die neue MCG-Adresse kein zweites Mal angeben. Die Array-Kapazität der gespeicherten MCGs beträgt 100. Wenn das Array überlastet ist, werden die ersten beigetretenen MCG überschrieben.

Mehr Informationen:
www.beckhoff.de/stage

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Deutschland
Telefon: +49 5246 9630
info@beckhoff.de
www.beckhoff.de

