

## Application Note DK9221-0613-0063

### Oversampling

#### Keywords

SMPTE  
Timecode  
EL3702  
Entertainment  
Bühne  
Show  
Licht  
Technologie  
Lichtsteuerung  
Dekodieren  
Zeitinformation  
Synchronisieren  
LED

# SMPT-imecode-Daten auslesen

Dieses Application Example enthält die Grundlagen zum SMPTE-Timecode sowie einen TwinCAT-Sample-Code zum Auslesen der von einem SMPTE-Master gesendeten Zeitinformationen. Mit Hilfe einer EtherCAT-Oversampling-Klemme EL3702 kann das vom Master an die Slaves gesendete Analogsignal „abgehört“ werden und über TwinCAT zur Synchronisierung unterschiedlicher Gewerke und Systeme genutzt werden (siehe Sample-Code in DK9222-0213-0063\_TwinCAT\_sample.zip).

## Vorwort

In diesem Dokument wird der Begriff „Frame“ im doppelten Sinne verwendet:

- Teil oder Einheit eines Protokolls
- Standbild bzw. Bildsequenz als Bestandteil eines Films

## Was ist SMPTE-Timecode?

Der SMPTE-Timecode ist ein von der Society of Motion Picture and Television Engineers SMPTE (Gesellschaft der Film- und Fernsehingenieure) definierter Standard für absolute Zeitangaben, die mit Medienmetadaten übertragen werden. Der SMPTE-Timecode wird hauptsächlich zur Synchronisierung der verschiedenen Spuren (Ton/Geräusch, Sprache, Special-Effects, Texte usw.) in einer Medienproduktion verwendet. Ein Frame besteht aus 80 Bit und wird als Broadcast vom Master an die Slaves gesendet. Über ein Analogsignal ( $\pm 1 \dots 10$  V) liefert der SMPTE-Timecode eine absolute Zeitreferenz auf Basis eines binärkodierten dezimalen 24h-Formates (hh:mm:ss:ff). Es gibt je nach Signalkodierung unterschiedliche, in Frames pro Sekunde (fps) ausgedrückte Frameraten. Die geläufigsten Werte sind 24, 25 und 30 fps.

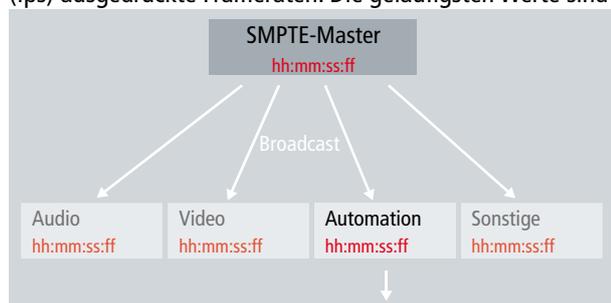


Abb. 1 SMPTE-Timecode: Senden einer absoluten Zeitreferenz an die Slaves (Broadcast)

## Application Note DK9221-0613-0063

### Oversampling

#### Protokolldetails | Dauer eines einzelnen Bits

Im ersten Schritt muss die Dauer eines einzelnen Bits im Verhältnis zur Framerate bestimmt werden. Aufgrund unterschiedlicher Frameraten (24, 25, 30) ändert sich die Dauer eines einzelnen Bits und wird folgendermaßen berechnet:

Frames pro Sekunde	Dauer eines Bits	Dauer eines Frames
30	$416\frac{2}{3} \mu\text{s}$	$80 \times 416\frac{2}{3} \mu\text{s} = 33\frac{1}{3} \text{ms}$
25	$500 \mu\text{s}$	$80 \times 500 \mu\text{s} = 40 \text{ms}$
24	$520 \mu\text{s}$	$80 \times 520 \mu\text{s} = 41\frac{2}{3} \text{ms}$

Tab. 1 Dauer eines Bits

#### Protokolldetails | Eins- oder nullwertiges Bit (engl. High or low bit)

Im zweiten Schritt müssen die Vorzeichenwechsel (engl. transitions) innerhalb der errechneten Bitdauer gezählt werden. Die Bits sind zweiphasig kodiert: Ein „0“-Bit hat einen einzigen Vorzeichenwechsel am Anfang der Periode. Ein „1“-Bit hat zwei Wechsel; zu Beginn und in der Mitte der Periode. Durch Zählung der Vorzeichenwechsel innerhalb der errechneten Zeitspanne wird ermittelt, ob das Bit eins- oder nullwertig ist. Ist das Signal bei einer Framerate von 30 fps über die Dauer von  $417 \mu\text{s}$  stabil, dann ist es 0. Werden zwei konstante Signale von  $\sim 208 \mu\text{s}$  erkannt, dann ist es 1.

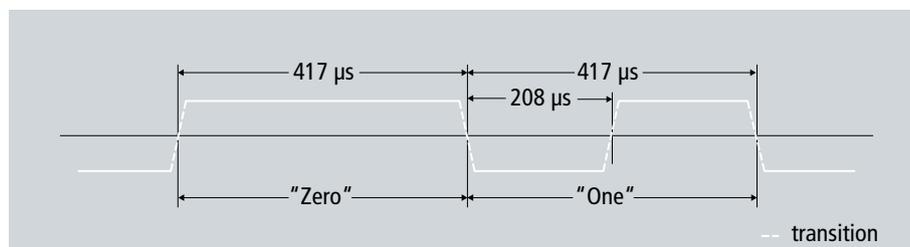


Abb. 2 Signalverlauf bei einer Framerate von 30 fps

#### Protokolldetails | Frame-Setup

Ein Timecode-Frame besteht immer aus 80 Bits und ist daher 10 Byte lang. Zur Markierung des Frameendes entsprechen die beiden letzten Bytes immer einer definierten Sequenz, dem sogenannten **SyncWord**: 0011.1111.1111.1101. In Tabelle 2 ist der Aufbau eines vollständigen Timecode-Frames inkl. SyncWord dargestellt. Die erste Spalte gibt einen Überblick über die Segmentierung der Zeitangaben.

## Application Note DK9221-0613-0063

### Oversampling

	Frame units	Frame tens	Second units	Second tens	Minute units	Minute tens	Hour units	Hour tens	Sync word (Feste Bitsequenz)	
Bit 0	1	10	1	10	1	10	1	10	binär 0	binär 1
Bit 1	2	20	2	20	2	20	2	20	binär 0	binär 1
Bit 2	4	x	4	40	4	40	4	x	binär 1	binär 1
Bit 3	8	x	8	x	8	x	8	x	binär 1	binär 1
Bit 4	x	x	x	x	x	x	x	x	binär 1	binär 1
Bit 5	x	x	x	x	x	x	x	x	binär 1	binär 1
Bit 6	x	x	x	x	x	x	x	x	binär 1	binär 0
Bit 7	x	x	x	x	x	x	x	x	binär 1	binär 1
	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10

Tab. 2 Übersicht der binärkodierte Zeitangaben im SMPTE-Frame. Bits, die keine Informationen enthalten, sind mit einem „x“ gekennzeichnet.

### Frameanalyse

In der folgenden Abbildung werden zwei Abschnitte eines Frames beispielhaft decodiert. Abbildung 3 zeigt den vollständigen SMPTE-Frame, Abbildung 4 zeigt anhand der Einer- und Zehner-Bereiche (Frame units und Frame tens), wie die binärkodierten Zeitinformationen analysiert und in dezimale Zählweise umgewandelt werden.

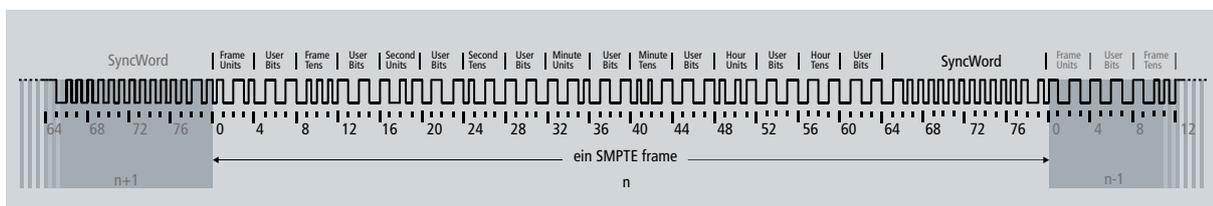


Abb. 3 Ein SMPTE-Beispiel-Frame

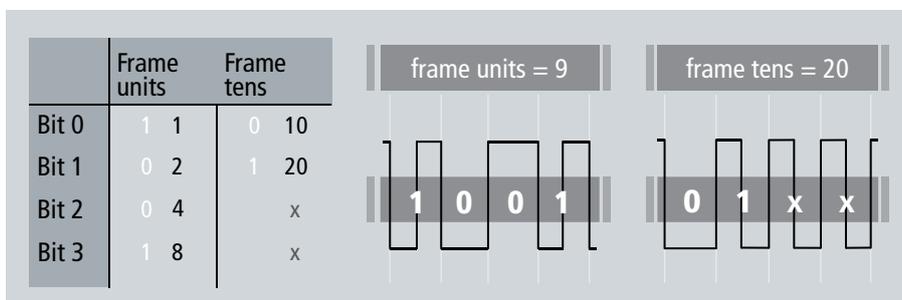


Abb. 4 Bitfolge 1001 | Frame-Einer (frame units) = 9; Bitfolge 01xx | Frame-Zehner (frame tens) = 20, da die letzten beiden Bits keine Zeitangaben enthalten

Der vollständige Wert für **Frame** addiert sich aus:  $20 + 9 = 29$ . Es ist der 29. Frame in dieser Sekunde. Das Auslesen weiterer Zeitinformationen aus dem vollständigen, in Abb. 3 gezeigten Beispiel-Frame ergibt: 05:38:14:29.

## Application Note DK9221-0613-0063

### Oversampling

Nach der Erläuterung der Frame-Kodierung gehen wir nun auf das geeignete Hardware-Setup für den Einsatz des TwinCAT-PLC-Sample-Codes ein. Das Setup beinhaltet einen externen SMPTE-Timecode-Master und eine EtherCAT-Klemme EL3702.

### TwinCAT-Sample-Program

Die Datei TwinCAT sample.zip beinhaltet ein Sample-Programm zum Auslesen der Zeitinformationen aus einem SMPTE-Frame.

Es wurde für das folgende Hardware-Setup programmiert:

- EL3702 | 2-Kanal-Analog-Eingangsklemme  $\pm 10$  V mit Oversampling
- EK1100 | EtherCAT-Koppler
- CX5010 | Embedded-PC mit Intel® Atom™ Prozessor – oder jeder andere Beckhoff IPC mit Prozessor höherer Leistung, siehe Kapitel **Systemanforderungen**
- Video Binloop | Externer SMPTE-Master von Alcorn McBride Inc.

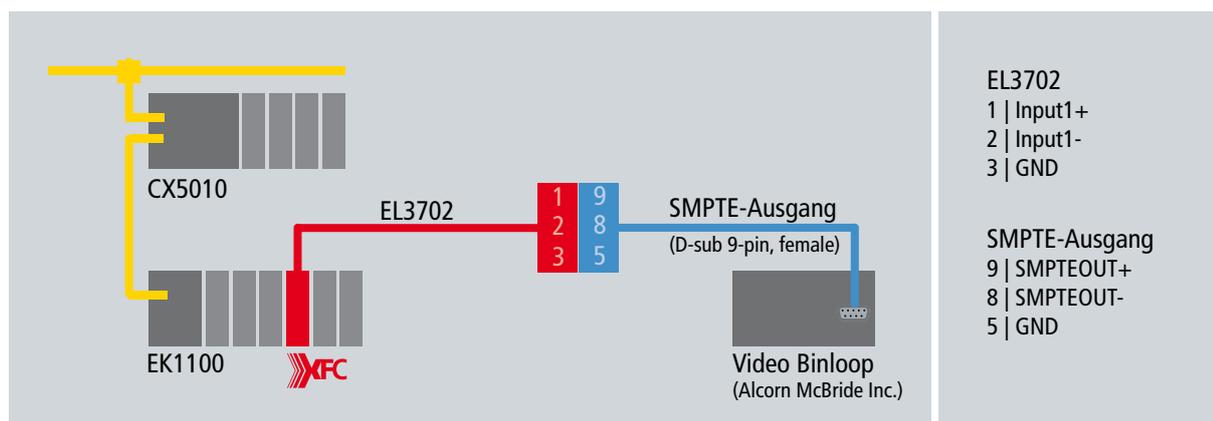


Abb. 5 Hardware-Setup und Verkabelung des Timecode-Masters mit der EL3702

Der SMPTE-Stecker bietet sowohl einen Ein- als auch einen Ausgang für die SMPTE-Schnittstelle des Binloop. Der abgeglichene 600-Ohm-Eingang ist galvanisch getrennt und eignet sich für Eingangsspannungen zwischen 1...5 V<sub>ss</sub>. Der abgeglichene Ausgang fällt standardmäßig auf 0 dBV bei 150 Ohm, kann aber auch zwischen 0...4 dBV am Bedienfeld des Binloop eingestellt werden.

## Application Note DK9221-0613-0063

### Oversampling

Folgende Tabelle zeigt einen Überblick über die Settings des Video Binloop, die nicht in TwinCAT, sondern extern am Gerät eingestellt werden.

Setting	Beschreibung	Wert
SMPTE	legt fest, ob die SMPTE-Schnittstelle den Timecode generiert oder einen externen Timecode einliest.	Gen
Frame rate	Framerate mit der der Timecode generiert wird.	23,976
Preroll Time	Anfangszeit des SMPTE-Generators, wenn er das erste Mal aktiviert wird.	00:00:00.00
Start Time	Zeitwert, den der SMPTE-Generator nach Erreichen der Endzeit wieder annimmt (Loop-Ende).	00:00:00.00
End Time	Zeitwert, an dem der SMPTE-Generator entweder stoppt oder zur Startzeit zurückkehrt.	00:00:00.00
Loop Mode	Legt fest, ob der SMPTE-Generator als Loop von der Start- zur Endzeit durchläuft.	enabled
Powerup Mode	Legt fest, ob der SMPTE-Generator beim Einschalten von Binloop aktiviert oder deaktiviert ist.	Start
Restart Mode	Legt fest, wie der SMPTE-Generator auf einen Neustart-Befehl reagiert. Er kann entweder neu starten oder den Startbefehl ignorieren.	enabled
Idle Mode	Legt fest, ob der SMPTE-Ausgang stumm geschaltet oder aktiv ist, wenn die SMPTE-Schnittstelle sich im Leerlauf befindet.	SMPTE Active
Self Gen Mode	Legt fest, wie lange der SMPTE-Leser freiläuft, nachdem das externe SMPTE-Signal verloren ging.	disabled
Output Level	Konfiguriert den Spannungspegel (in dB) des SMPTE-Ausgangs.	4 dBV

Tab. 3 Die Parameter des Binloop (Auszug aus der Bedienungsanleitung, siehe Webseite Alcorn McBride Inc.) werden nicht über TwinCAT konfiguriert.

Hinweis:

Theoretisch kann die Information bei allen Frameraten gelesen werden, aber das Sample-Programm ist für 24 fps konzipiert und nicht bei anderen Frameraten getestet. Darüber hinaus wird lediglich die Zeitinformation ausgelesen, nicht aber die Userbits, die Flags und andere mit dem Frame übermittelte Informationen.

## Application Note DK9221-0613-0063

### Oversampling

#### TwinCAT-Sample

In der Datei DK9222-0213-0063\_TwinCAT\_sample.zip befindet sich das TwinCAT-Sample-Programm unter **SMTPE\_Timecode.pro**.

Im Sample-Code werden die Zeitinformationen in sechs Schritten aus dem SMPTE-Frame ausgelesen:

- Konditionierung der EL3702
- Erfassen der Übergänge
- Analyse der Übergänge
- Identifizierung der eins- und nullwertigen Bits
- Analyse des Bitstroms
- Auslesen der Bits

#### 1 | Konfiguration der EL3702

Die EtherCAT-Oversampling-Klemme EL3702 erzeugt aus dem mitgehörten SMPTE-Ausgangssignal ein Werte-Array für die TwinCAT-SPS. Standardmäßig ist die Zykluszeit der SPS auf 1 ms und die Oversampling-Rate der EL3702 auf Faktor 100 eingestellt. Demzufolge wird das Signal in Fragmente von 10 µs aufgesplittet, sodass für jedes Timecode-Bit 40...50 Samples analysiert werden können.

#### 2 | Erfassung der Übergänge

Im array **Tab\_EL3702\_Doublecycles** sind die Samples von zwei Zyklen gespeichert: die des aktuellen Zyklus (n) und die des vorherigen Zyklus (n-1). In jedem SPS-Zyklus werden die Samples des aktuellen Zyklus (n) abgearbeitet. Die Samples von n-1 werden dazu verwendet, um den Zustand des Timecode-Eingangs in diesem Zyklus mit dem des vorherigen Zyklus zu vergleichen, um daraus die Vorzeichenänderungen abzuleiten. Anhand dieser Vorzeichenänderung wird ein Übergang (engl. transition) erkannt.

**IF Tab\_EL3702\_Doublecycles[ i-1 ] \* Tab\_EL3702\_Doublecycles [ i ] < 0**

**(\* The analog signal changes sign --> there is an edge \*)**

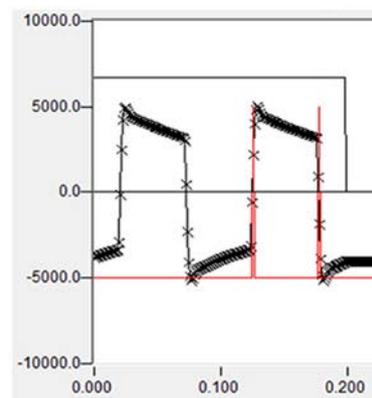
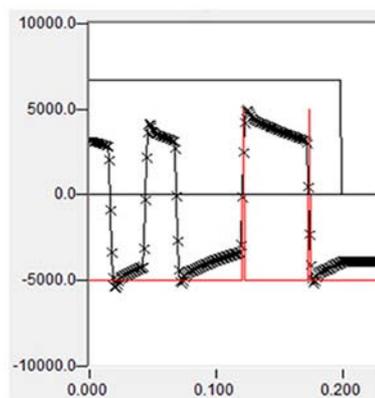
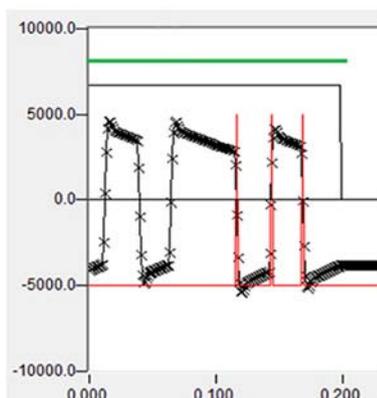
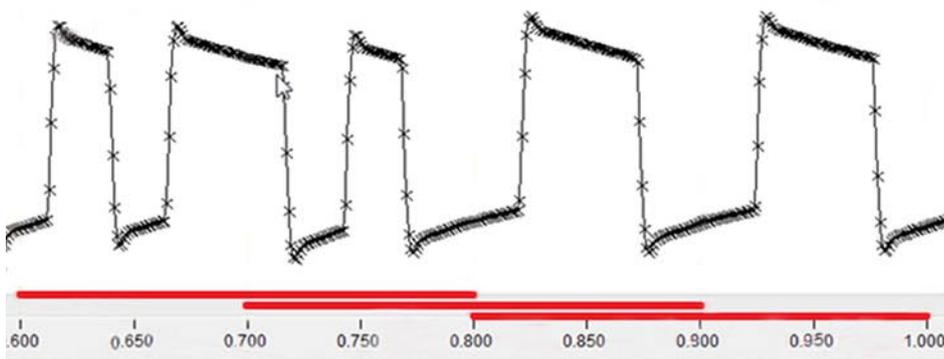
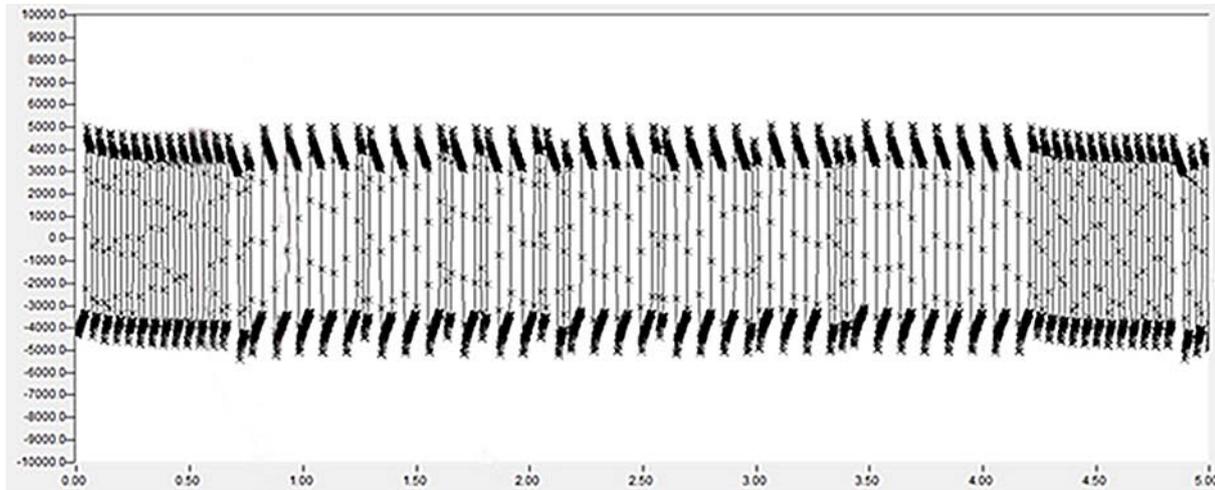
**OR Tab\_EL3702\_Doublecycles [ i-1 ] <> 0 AND Tab\_EL3702\_Doublecycles [ i ] = 0 THEN**

#### 3 | Analyse der Übergänge

Im Array **Tab\_EL3702\_DoublecyclesTrig** sind die Übergänge des Timecode-Signals hinterlegt. Es werden immer nur die Übergänge des aktuellen Zyklus (n) berücksichtigt. **Tab\_EL3702\_Doublecycles** enthält die von der EL3702 generierten Samples aus den beiden letzten SPS-Zyklen. Es werden nur die Übergänge des vergangenen Zyklus geprüft (in untenstehender Abbildung rot markiert).

## Application Note DK9221-0613-0063

### Oversampling



Im Pointer **Index\_Tab\_EL3702** wird die Position der Übergänge innerhalb des letzten Zyklus gespeichert. Für den folgenden Zyklus wird dieser Pointer verschoben und die weitere Analyse nach Übergängen an der Position des zuletzt erkannten Vorzeichenwechsels fortgesetzt. Wird kein weiterer Übergang erfasst, wird ein Fehler in **Index\_Tab\_EL3702\_Error** gemeldet.

## Application Note DK9221-0613-0063

### Oversampling

#### 4 | Identifizierung der eins- und nullwertigen Bits

Anhand des Abstandes zwischen den Übergängen kann der Zustand des Bits (high = 1 oder low = 0) bestimmt werden. Als Referenzabstand **ConstantTime** wird der Wert 330 µs verwendet:

Bei der maximalen Framerate von 30 fps ist die Mindestdauer eines low-Bits, also der kleinstmögliche Zeitabstand zwischen zwei Übergängen, am geringsten und liegt bei 417 µs. Da bei einem high-Bit zwei Übergänge vorliegen, wird nur die Hälfte des Abstand als Referenz angenommen.

Legt man mit 24 fps die niedrigste Framerate (= Längste Dauer eines Bits) für die Berechnung des Referenzabstandes zugrunde, beträgt der maximale Wert für eine „Halbzeit“ 260 µs (0,5 x 520 µs). Dies ergibt eine Referenz von < 330 µs, da diese feste Grenze **ConstantTime** zu allen Frameraten bei der Suche nach eins- und nullwertigen Bits passt:

- Wenn **ConstantTime** > 330 µs, dann beinhalten die Übergänge ein nullwertiges Bit (0). („duration long“)
- Wenn **ConstantTime** < 330 µs, dann können die beiden gemessenen Übergänge Teil einer dreifachen Identifizierung eines einwertigen Bits (1) sein. Wenn die nächste Periode ebenfalls < 330 µs, dann ist ein einwertiges Bit (1) identifiziert. („duration short“).

#### 5 | Analyse des Bitstroms

Um ein Bit als 1 zu identifizieren, wird die Variable **Even** bei der ersten Erfassung von „duration short“ auf **TRUE** gesetzt.

Wenn **Even** bei der Erfassung einer „duration long“ immer noch **TRUE** ist, wird ein Fehler ausgegeben (**FrameError**), denn die Anzahl der aufeinanderfolgenden „duration short“ muss immer gerade sein. Demzufolge kann ein einwertiges Bit (1) erst dann identifiziert werden, wenn vorher ein niedrigwertiges Bit (0) aufgetreten ist. Mit **First\_BitZero\_Reach** wird die Erfassung eines einwertigen Bits (1) erst dann gestartet, wenn zuvor ein niedrigwertiges Bit (0) bereits erfasst wurde.

Die erfassten Bits werden in das Array **Tab\_TimeCodeFrame** gespeichert, das die Sequenz der im Timecode-Signal erfassten Bits beinhaltet.

#### 6 | Auslesen der Bits

Um die 80 Bits eines gültigen Timecode-Frames zu identifizieren, muss das Array **Tab\_TimeCodeFrame** im nächsten Schritt, nach der Bitfolge des SyncWord durchsucht werden: Wird die SyncWord-Sequenz hinter Position 80 im Array erkannt, kann ein ganzer gültiger Frame in **LastTimeCodeFrame** ausgelesen werden. Nun kann der Frame gemäß Tabelle 2 in **TimeCodeFrame** dekodiert werden, um die darin enthaltenen Informationen zu extrahieren.

Um ein Overflow beim Array **TimeCodeFrame** zu verhindern, wird ein Fehler in **Index\_Tab\_TimeCodeFrame\_Error** ausgegeben, wenn mehr als 300 Timecode-Bits ohne die SyncWord-Sequenz erfasst werden.

#### Systemanforderungen/Einschränkungen

Das Sample-Programm lastet die CPU stark aus, da große Datenmengen kopiert und viele Berechnungen ausgeführt werden. Die Ausführung des Programms auf einem CX5010 bei einer Taskzeit von **T#1ms** nimmt 50 % der CPU-Kapazität in Anspruch.

## Application Note DK9221-0613-0063

### Oversampling

Bei Erweiterung der Taskzeit auf **T#2ms** bei einer Oversampling-Rate von 100 ist die CPU immer noch zu 25 % ausgelastet. Unter diesen Bedingungen können jedoch 20...25 Samples pro Timecode-Bit verarbeitet werden, was letztendlich ein akzeptabler Wert ist.

Zu beachten ist: Wenn die Taskzeit geändert wird, müssen auch andere Konstanten im Programm geändert werden:

- **10** steht für eine 10-µs-Oversampling-Rate (Taskzeit/Oversampling-Rate: 1 ms/1000 = 10 µs). Wird die Taskzeit auf 2 ms erweitert, muss die Oversampling-Rate auf **20** gesetzt werden.
- Wird die Oversampling-Rate im Quellcode geändert, dann muss der Faktor auch im System Manager geändert werden.

```

0033 LastEdge := i;
0034 ELSE (* Let's go ... *)
0035 ConstantTime := (i - LastEdge) * 10; (* if EL3702 Oversampling time = 10 µs*)
0036 LastEdge := i;
0037 IF ConstantTime < 330 AND ConstantTime <> 0 AND First_BitZero_Reach THEN (* It's part of a 1 signal *)
0038 (* 330µs is a fix limit that match all frame rates : Minimum period is 400 µs at 30 fr/sec, maximum half period is 260 µs at 24 fr/sec *)

0042 VAR CONSTANT
0043 OversamplingFactor: WORD := 100; (* EL3702 Oversampling rate = 100 samples per cycle *)
0044 DoubleOversamplingFactor: WORD := 200; (* = OversamplingFactor x 2 *)
0045

```

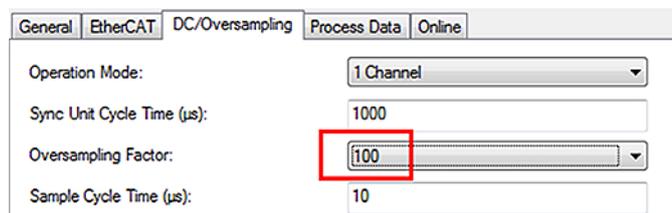


Abb. 6 Bei Änderungen der Zykluszeit müssen weitere Konstanten geändert werden.

Mit einer Taskzeit von **T#1ms** und einer Oversampling-Rate von 50 ist die CPU immer noch mit 40 % ausgelastet. Eine weitere Verringerung ist nicht möglich, da in dem Falle die Anzahl der Samples nicht mehr ausreicht, um das Signal korrekt zu generieren. **Aus diesem Grund wird der Einsatz von einem CX5020 oder einem Gerät mit noch höherer Leistung empfohlen.**

Zykluszeit	Oversampling-Rate	CPU-Auslastung
1 ms	50	40 %
1 ms	100	50 %
2 ms	100	25 %

Tab. 4 Performance-Einschränkungen bei einem CX5010

## Application Note DK9221-0613-0063

### Oversampling

–  $\pm 10$  V analoge Oversampling-Eingangsklemmen [www.beckhoff.de/EL3702](http://www.beckhoff.de/EL3702)

– Download-Link des TwinCAT-Samples

[http://download.beckhoff.com/download/document/Application\\_Notes/DK9222-0213-0063\\_TwinCAT\\_sample.zip](http://download.beckhoff.com/download/document/Application_Notes/DK9222-0213-0063_TwinCAT_sample.zip)

– Download-Link Application Example zum DMX-Protokoll

[http://download.beckhoff.com/download/document/Application\\_Notes/DK9222-0311-0029.pdf](http://download.beckhoff.com/download/document/Application_Notes/DK9222-0311-0029.pdf)

– PC-based Control für Bühnen- und Showtechnologie [www.beckhoff.de/stage](http://www.beckhoff.de/stage)

– SPS und Motion Control auf dem PC [www.beckhoff.de/TwinCAT](http://www.beckhoff.de/TwinCAT)

– SMPTE-Timecode-Master 'Video Binloop' von Alcorn McBride Inc. [www.alcorn.com](http://www.alcorn.com)

Dieses Dokument enthält exemplarische Anwendungen unserer Produkte für bestimmte Einsatzbereiche. Die hier dargestellten Anwendungshinweise beruhen auf den typischen Eigenschaften unserer Produkte und haben ausschließlich Beispielcharakter. Die mit diesem Dokument vermittelten Hinweise beziehen sich ausdrücklich nicht auf spezifische Anwendungsfälle, daher liegt es in der Verantwortung des Kunden zu prüfen und zu entscheiden, ob das Produkt für den Einsatz in einem bestimmten Anwendungsbereich geeignet ist. Wir übernehmen keine Gewährleistung, dass der in diesem Dokument enthaltene Quellcode vollständig und richtig ist. Wir behalten uns jederzeit eine Änderung der Inhalte dieses Dokuments vor und übernehmen keine Haftung für Irrtümer und fehlenden Angaben. Eine detaillierte Beschreibung unserer Produkte enthalten unsere Datenblätter und Dokumentationen, die darin enthaltenen produktspezifischen Warnhinweise sind unbedingt zu beachten. Die aktuelle Version der Datenblätter und Dokumentationen finden Sie auf unserer Homepage ([www.beckhoff.de](http://www.beckhoff.de)).

© Beckhoff Automation GmbH, Juni 2013

Die Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.