

EtherCAT timing

FAQ No.0005

Part	Version	Revision	Date	Status
de	6.3.0.0	006	2020-03-12	released

Inhalt

Einführung	2
Timing (Frame at task start=false).....	2
Allgemeiner Masterzyklus	2
Wo Sie diese Werte in einen Standard Projekt finden?	3
Sendezyklus, 2 Masterzyklen	4
Wo finde ich diese Werte in einem Standardprojekt?	6
Sync Offset.....	7
Verschiebezeit	8
Anwendungsbeispiele	8
Bestimmung und Einstellung der Telegrammposition	9
Berechnung.....	9
Messung.....	10
Sende-/Empfangszyklus, 4 Masterzyklen	11
Zusammenfassung: Frame bei Task start=false.....	11
Timing (Frame at task start=true)	12
Sendezyklus, 2 Masterzyklen	12
Berechnung	13
Messung.....	13
Zusammenfassung: Frame bei Task start=true	14
Wie kann ich die Einstellungen für das Frame-Sendeereignis ändern?	15
Anhang	16
Tabelle vom KEB PLC timing.....	16
Tabelle vom KEB DRIVE timing.....	16
Disclaimer	17

Einführung

Dieses Dokument gibt einen allgemeinen Überblick über das KEB EtherCAT Timing Verhalten für den Synchronbetrieb. Allgemeine Begriffe und mögliche Benutzereinstellungen werden erläutert.

Timing (Frame at task start=false)

Allgemeiner Masterzyklus

Die allgemeine Aufgabe der SPS ist es, einen technischen Prozess in Echtzeit mit deterministischem Verhalten zu steuern. Echtzeitfähigkeit ist die Fähigkeit, rechtzeitig auf Prozesseingaben (Sensoren) zu reagieren, indem Prozessausgaben (Aktoren) gesteuert werden, um ein stabiles Verhalten zu gewährleisten. Der Austausch der Prozessdaten erfolgt über eine Feldbus Master Hardware. Der Feldbus Stack übernimmt die Kommunikation zwischen Master und Slave Geräten. Dieses Dokument beschreibt den Prozessdatenaustausch auf Basis der EtherCAT-Feldbustechnologie.

Ein allgemeiner SPS Zyklus ist in bestimmte Ereignisse und Abläufe gegliedert. Der Zyklus wird in fest eingestellten Zeitintervallen (Zykluszeit) kontinuierlich wiederholt, wobei das Zyklusstartereignis (Ereignis a) in Abhängigkeit von der Regelqualität (Bereich A) jittert. Die Zyklussequenz umfasst hauptsächlich das Lesen des Prozesses in Daten (Bereich B), das Ausführen von Benutzercode (Bereich C+D) und das Schreiben von Prozessdaten (Bereich E).

Der Benutzercode ist nach IEC61131-3 mit KEB COMBIVIS studio 6 frei programmierbar. Abhängig von der Leistung von der Zielhardware benötigt jeder Befehlsaufruf eine gewisse Zeit. Je nach Anzahl und Art der Anweisungen variiert die Ausführung des Benutzercodes in dem Zyklus zwischen minimaler Zykluszeit und maximaler Zykluszeit (Bereich D).

Deterministisches Verhalten kann erreicht werden, wenn die maximale Zykluszeit immer kleiner als die eingestellte Zykluszeit ist.

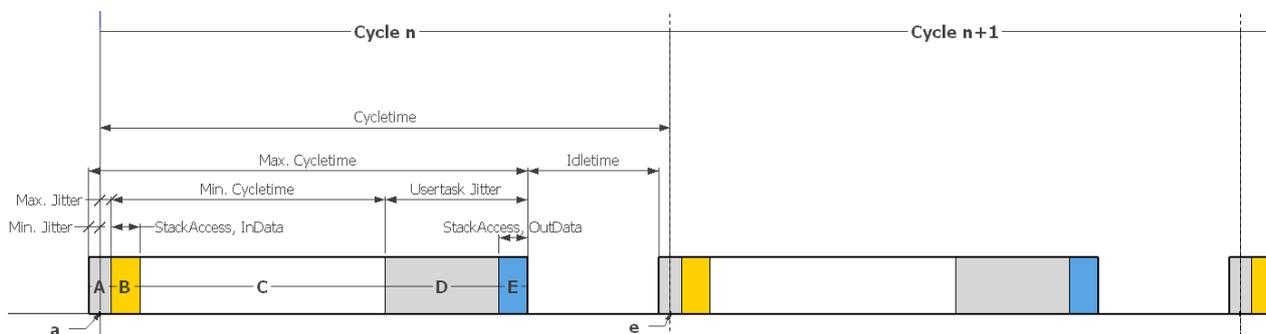


Diagramm 1 – Allgemeiner Masterzyklus (EtherCAT master task)

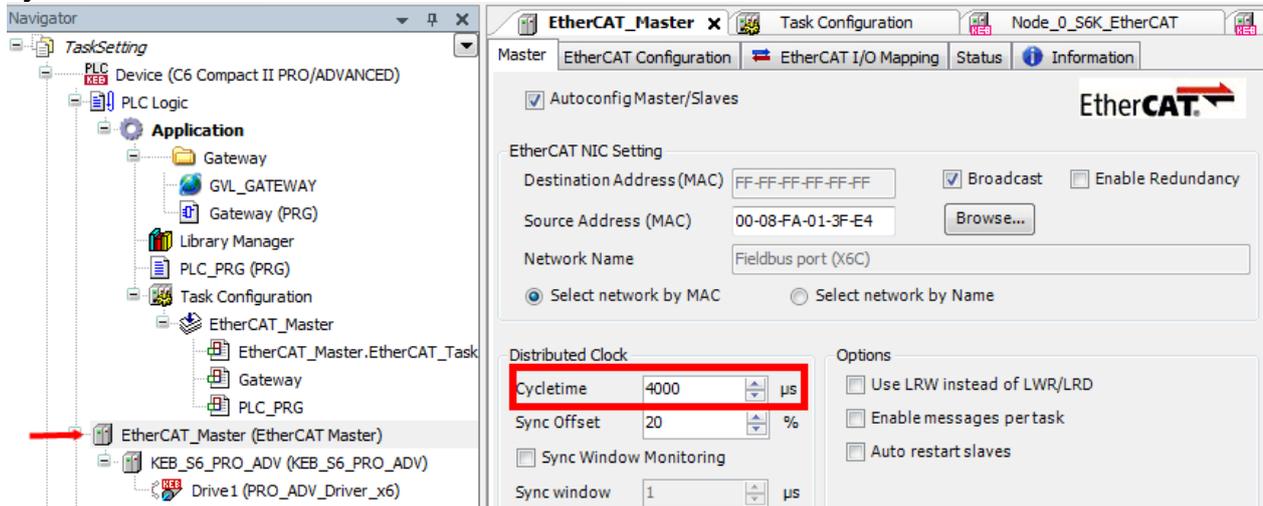
a	Durchschnittliche Zyklus-Startereignis, Ausführungsbeginn des SPS Zyklus n
e	Durchschnittliches Zyklusende n, Ausführungsbeginn des SPS Zyklus n+1
A	Jitter des SPS-Task Startereignis.
B	Stack Zugriff zum Einlesen von Daten.
C	Ausführung von Benutzercode. Interpretation der aktuellen Prozesswerte und Berechnen der Sollwerte zur Steuerung eines technischen Prozesses.
D	Jitter der Usercodeausführungszeit.
E	Stack Zugriff auf das Schreiben von Daten und das Senden des EtherCAT-Telegramms.
Cycletime	Konfigurierte EtherCAT-Bus Zykluszeit (Einstellung von EtherCAT-Master / EtherCAT-Task).
Max. Cycletime:	Maximal gemessene Laufzeit des SPS-Zyklus.
Min. Cycletime:	Minimal gemessene Laufzeit des SPS-Zyklus.
IdleTime:	Puffer-/Leerlaufzeit am Ende der SPS-Task zum Startereignis des nächsten Zyklus.
Max. Jitter	Jitter der SPS-Task, maximaler Wert.
Min. Jitter	Jitter der SPS-Task, minimaler Wert.

Stack Zugriff, InData: Zeit, die für die Stack Verarbeitung und das Lesen von Prozesseingangsdaten benötigt wird (einschließlich Stack bezogener Offsetzeit).

Stack Zugriff, OutData: Zeitbedarf für die Stack Verarbeitung und das Schreiben von Prozessausgangsdaten.

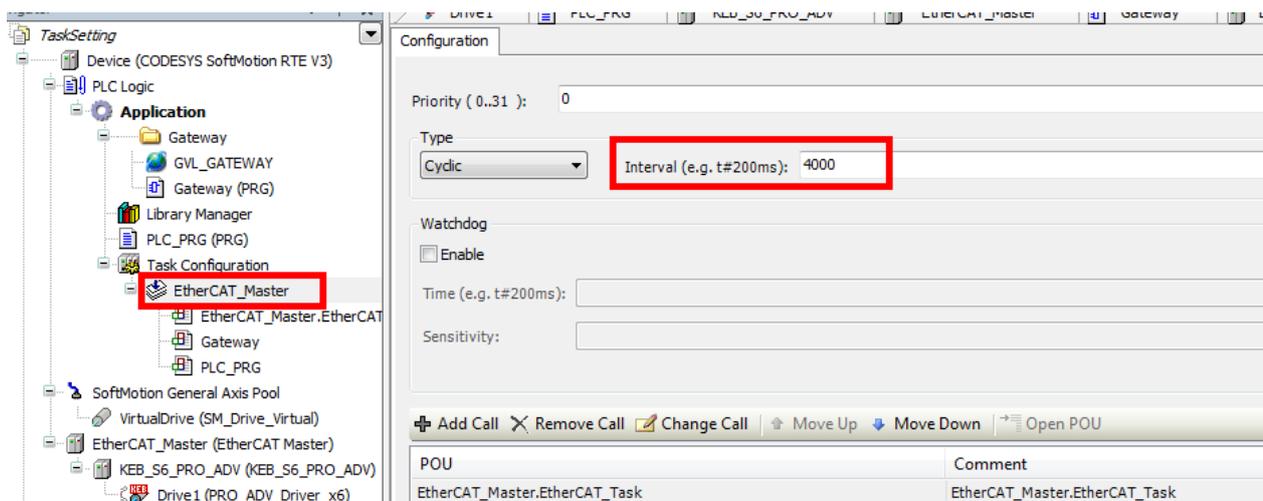
Wo Sie diese Werte in einen Standard Projekt finden?

Zykluszeit



The screenshot shows the 'EtherCAT Master' configuration window. The 'Cycletime' field is set to 4000 µs and is highlighted with a red box. Other fields include 'Destination Address (MAC)', 'Source Address (MAC)', 'Network Name', 'Distributed Clock', and 'Options'.

Zykluszeit in den EtherCAT Master Einstellungen.

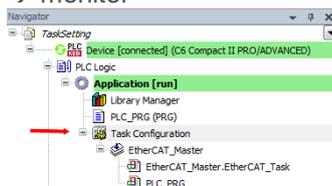


The screenshot shows the 'Task Configuration' window for the 'EtherCAT Master' task. The 'Interval (e.g. t#200ms):' field is set to 4000 and is highlighted with a red box. The 'Type' is set to 'Cyclic'.

Die EtherCAT-Master Task Einstellungen müssen mit den selben Werten eingestellt werden.

Jitter

Min. Jitter, Max. Jitter, Max Zykluszeit, Min. Zykluszeit werden gemessen und angezeigt in Taskconfiguration → monitor



The screenshot shows the 'Task Configuration' window with the 'Monitor' tab selected in the Navigator.

Task	Status	IEC-Cycle Count	Cycle Count	Last Cycle Time (µs)	Average Cycle Time (µs)	Max. Cycle Time (µs)	Min. Cycle Time (µs)	Jitter (µs)	Min. Jitter (µs)	Max. Jitter (µs)
EtherCA...	Valid	5471	5471	691	694	853	666	-2	-3	3

Sendezyklus, 2 Masterzyklen

Das Diagramm 2 zeigt, wie Daten vom EtherCAT-Mastergerät an ein Slave Gerät gesendet werden. Die dargestellte Sequenz gilt für synchrone Operationen und die Master Einstellung „FrameAtStart=false“. Beispielsweise werden die Daten im Zyklus n (Bereich C) berechnet, sie werden am Ende des Zyklus (Bereich E,F) per Telegramm gesendet und gelangen in die Slave Felbushardware (Bereich G). Nach dem nächsten Synchron Interrupt (Ereignis s) werden die verfügbaren Daten mit der Slave Firmware (Bereich H) ausgetauscht und in der Slave spezifischen Steuersequenz (Bereich I) berechnet. Die Position der Telegrammankunft (Bereich G) ist abhängig von der eingestellten Zykluszeit, dem eingestelltem Synchron-Offset, der min./max. Zykluszeit, weiteren Zeitkonstanten und dem Jitter in Abhängigkeit von Hardware und verwendeten Software Stack. Darüber hinaus wird die Position vom Stack Zugriff (Bereich E) von der SPS geregelt, so dass sie von den gemessenen Timings der vorherigen Zyklen abhängt.

Um ein deterministisches Verhalten zu erreichen, ist zu vermeiden, dass sich der Bereich der Telegrammankunft (Bereich G) mit dem Bereich des Telegrammzugriffs (Bereich H) überschneidet.

Da die Position der Telegrammankunft von der Benutzeranwendung und der verwendeten Hardware abhängt, wird dringend empfohlen, die Telegrammposition für jede Anwendung zu überprüfen!

Dies kann offline unter Verwendung der beigefügten Berechnungsformeln oder online unter Verwendung des KEB EtherCAT-Assistenten erfolgen.

Zur Verschiebung des Telegrammankunftsbereichs kann ein **Synchron Offset definiert** werden [-50...50% der Zykluszeit]. Eine positive und negative Verschiebung der SPS gegenüber dem Synchronereignis ist möglich.

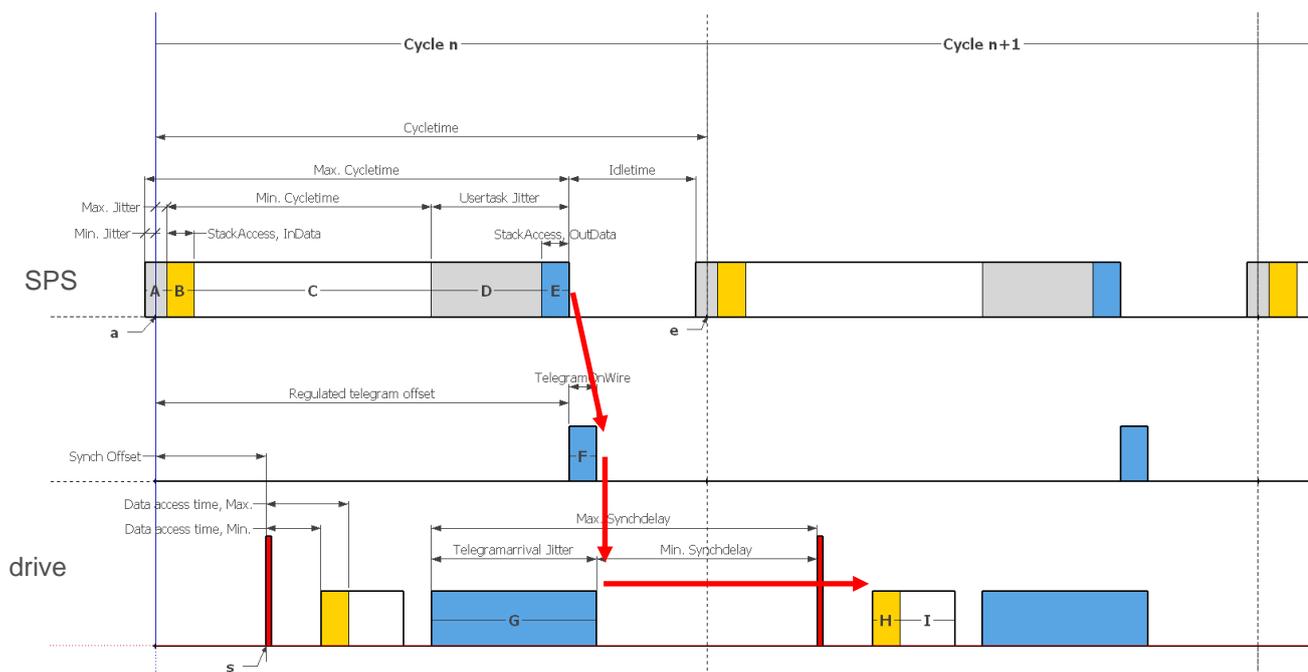


Diagramm 2 – Telegramm Sendezyklus

C: PDO-Daten n werden im SPS-Zyklus berechnet

F: Daten n auf der Leitung. Die Daten kommen beim Slave mit der fallenden Flanke dieses Bereichs an.

G: Telegramm-Eingangsbereich. Der Bereich jittert in Abhängigkeit von der Masterzyklusposition und dem Jitter.

H: Der PDO-Puffer wird vom Slave gelesen/geschrieben.

I: Die Daten n werden von der Slave-Firmware verarbeitet.

TelegramOnWire (range F)	Telegrammlaufzeit, typisch ca. 50µs
Max. Synchdelay :	Fallende Flanke von tTelegramOnWire. bezogen auf das Slave-Synchronisationsereignis
Min. Synchdelay:	Fallende Flanke von tTelegramOnWire. bezogen auf das Slave-Synchronisationsereignis
Telegramarrival Jitter, (range G):	Jitter bei Ankunft des PDOs am Slave-Gerät.
Telegramarrival Jitter :	Max. Zykluszeit - Minimale Zykluszeit (SPS-bezogene Berechnung.)
Telegramarrival Jitter:	Max. Synchroverzögerung - Minimale Zykluszeit (antriebsbezogene Berechnung)
Synch Offset [%]:	Offset des SPS-Zyklusstarts (relativ zur Synchronisation) in Prozent der Zykluszeit.
Data access time, Min:	Zeitpunkt des frühesten Zugriffs des EtherCAT-Slaves auf Prozessdaten (relativ zur Synchronisation). Nahezu konstanter Faktor abhängig vom Slave-Typ und der Firmware.
Data access time, Max:	Zeitpunkt des letzten Zugriffs des EtherCAT-Slaves auf Prozessdaten (relativ zur Synchronisation). Nahezu konstanter Faktor abhängig vom Slave-Typ und der Firmware.

Hinweis:

Zeitpunkte und Berechnungen gelten nur für stabile Zyklen. EtherCAT-Startsequenz und instabile Anwendungen mit Zykluszeitüberläufen sind ausgeschlossen.

Wo finde ich diese Werte in einem Standardprojekt?

Synchronisationsverzögerung, Telegramm-Eingangsjitter

Die auf das Synchronereignis bezogenen Telegrammankunftszeiten werden von KEB-Antrieben gemessen und im EtherCAT-Assistenten angezeigt.

Zykluszeit: Fb10
Min. synch Verzögerung: Fb27,
Max. synch Verzögerung: Fb28

Hinweis: Die Parameter Fb28, Fb27 zeigen einen positiven Wert an, aber der negative Abstand zum Synchronereignis wird gemessen. Um sie mit berechneten Werten zu vergleichen, muss sie mit -1 multipliziert werden.

Telegrankunft Jitter: Blauer Bereich:= Fb27-fb28

s Synch event: Rote → Puls Linie
Data Access: Gelb-orange → Zeitfenster des Datenzugriffs des Slaves.
Toleranz Zeit 1: Pufferzeit nach Slave-Datenzugriff, vor Telegrammeingang
Toleranz Zeit 2: Pufferzeit nach Telegrammeingang, vor dem nächsten Slave-Datenzugriff

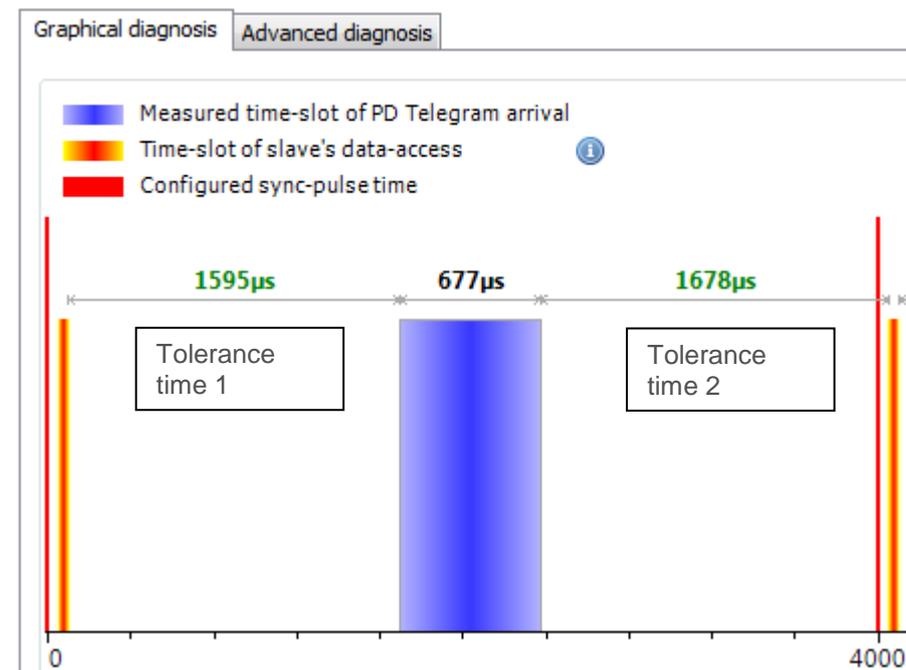
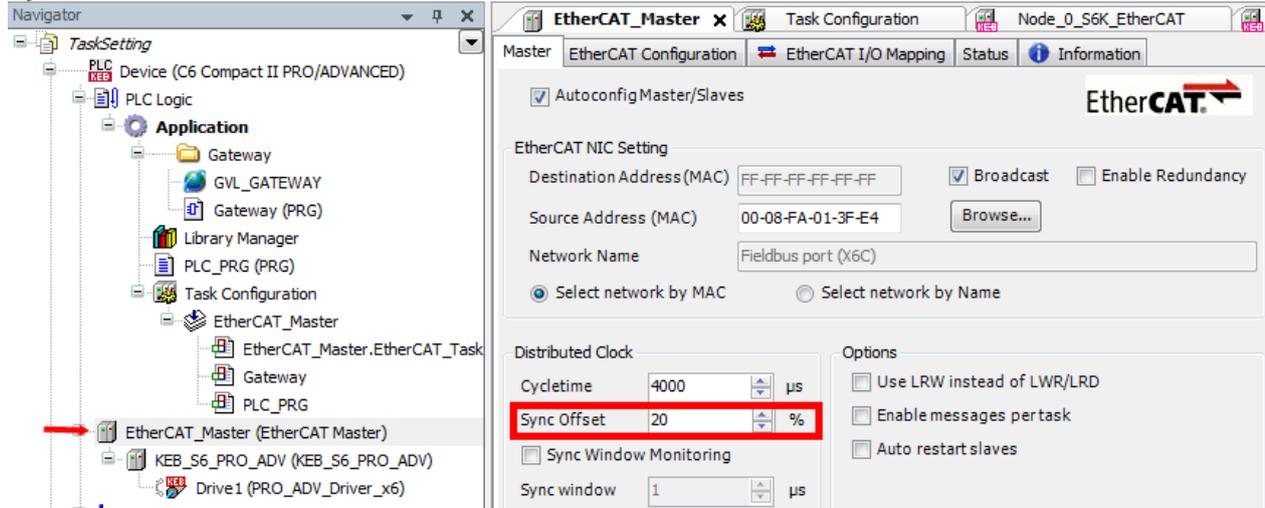


Diagramm 3 - Ethercat wizard

Name	Online value
st00: statusword	424: F + /QS + W + SY
fb 10: sync interval	4000
fb20: invalid frame count P0	0
fb21: RX error count P0	0
fb22: invalid frame count P1	0
fb23: RX error count P1	0
fb24: forwarded RX error count P0	0
fb25: forwarded RX error count P1	0
fb26: processing unit error count	255
fb27: min. sync delay	1628 µs
fb28: max. sync delay	2305 µs
fb29: no frame per sync cnt	0
fb30: mult. frames per sync cnt	0
fb31: no PDO data per sync cnt	0

Sync Offset



The screenshot displays the configuration interface for an EtherCAT Master. On the left, a 'Navigator' pane shows a project tree with 'EtherCAT_Master (EtherCAT Master)' selected. The main window is titled 'EtherCAT_Master' and contains several tabs: 'EtherCAT Configuration', 'EtherCAT I/O Mapping', 'Status', and 'Information'. The 'EtherCAT Configuration' tab is active, showing the following settings:

- Autoconfig Master/Slaves
- EtherCAT NIC Setting**
 - Destination Address (MAC): FF-FF-FF-FF-FF-FF Broadcast Enable Redundancy
 - Source Address (MAC): 00-08-FA-01-3F-E4
 - Network Name: Fieldbus port (X6C)
 - Select network by MAC Select network by Name
- Distributed Clock**
 - Cycletime: 4000 μs
 - Sync Offset: 20 %** (highlighted with a red box)
 - Sync Window Monitoring
 - Sync window: 1 μs
- Options**
 - Use LRW instead of LWR/LRD
 - Enable messages per task
 - Auto restart slaves

Der Synchron Offset kann am EtherCAT-Mastergerät in einem Bereich von -50...50% der Zykluszeit eingestellt werden.

Verschiebezeit

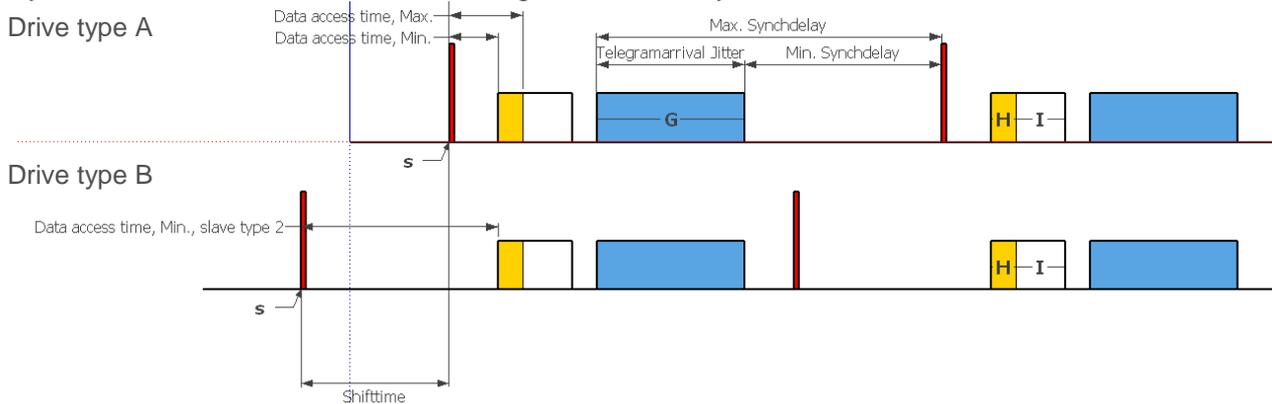
Im Gegensatz zum Synchron-Offset, das den Masterzyklus verschiebt, arbeitet die Slave-Schaltzeit auf den Zyklus jedes Slaves einzeln.

Durch die Änderung der Verschiebezeit wird das Synchronereignis eines einzelnen Slaves und damit alle slavebezogenen Bereiche wie der Zeitpunkt des Slave-Datenzugriffs verschoben. Im Allgemeinen sollte der Standardwert aus der Geräte Beschreibung des Herstellers verwendet werden. In besonderen Anwendungsfällen kann der Wert geändert werden.

Anwendungsbeispiele

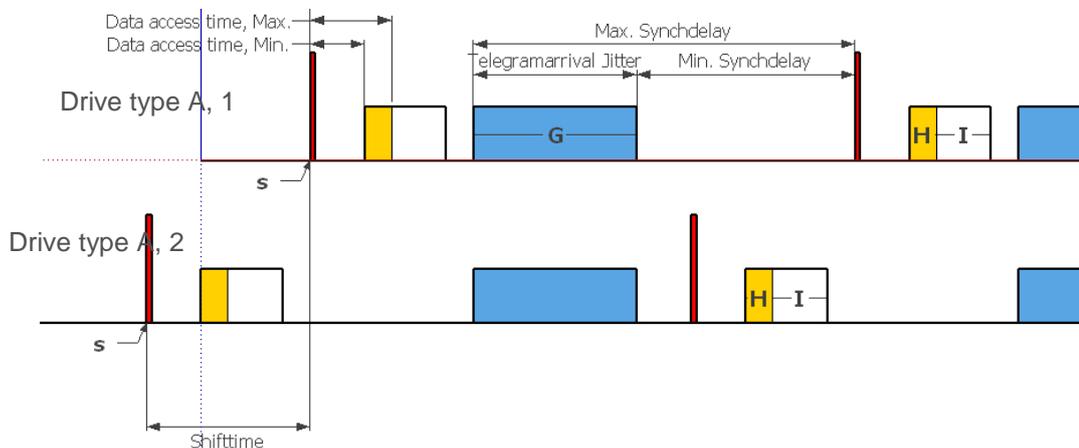
A) Synchronisieren Sie den Slave-Datenzugriff verschiedener Slave-Typen.

Verschiedene Slave-Typen können unterschiedliche Datenzugriffszeiten haben, so dass sich der Einfluss auf den technischen Prozess verschieben kann. Der technische Prozess kann mit höherer Genauigkeit synchronisiert werden, indem der Datenzugriff der Slaves synchronisiert wird.



B) Desynchronisieren Sie den Slave-Datenzugriff desselben Slave-Typs..

Der synchronisierte Zugriff mehrerer Slaves auf das technische System kann zu negativen Nebenwirkungen führen (z.B. PWM bezogene EMV, Spannungsabfall des DC-Kreises). Die Nebenwirkungen können durch Desynchronisation des Datenzugriffs der Slaves reduziert werden.



Wo finde ich die Slave-Verschiebezeit

The screenshot shows the 'Task Configuration' window in the KEB software. On the left, a tree view shows the project structure, including 'EtherCAT_Master', 'EtherCAT_Master.EtherCAT_Task', 'Gateway', 'PLC_PRG', and 'EtherCAT_Master (EtherCAT Master)'. Under 'EtherCAT_Master (EtherCAT Master)', the 'KEB_S6_PRO_ADV (KEB_S6_PRO_ADV)' and 'Drive1 (PRO_ADV_Driver_x6)' are listed. On the right, the 'Sync0' configuration is shown. The 'Enable Sync 0' checkbox is checked. The 'Sync Unit Cycle' is set to 'x 1' and '4000' (Cycle Time (µs)). The 'Shift Time (µs)' is set to '0' and is highlighted with a red box.

Bestimmung und Einstellung der Telegrammposition

Berechnung

Die Berechnung zeigt, wie man die Telegrammankunftsposition **ohne** Einsatz des EtherCAT-Assistenten errechnen kann.

Dieser Schritt kann übersprungen werden, wenn Messdaten aus dem EtherCAT-Assistenten verfügbar sind.

Um einen synchronen Betrieb zu gewährleisten, ist ein Puffer/Toleranz zwischen der Position des Telegrammeingangs, der so genannten Synchronisationsverzögerung und der Position des Slave-Datenzugriffs, erforderlich. Die Toleranzzeit vor und nach der Telegrammankunft muss größer als 0 sein, die Toleranzzeit kann durch Vergleich der Position der Telegrammankunft mit der Position des Slave-Datenzugriffs berechnet werden.

time 1 (siehe Diagramm 3, ethercat wizard):= Cycletime - Max. Synchdelay- Data access time, Max Tolerance time 2 (see diagram 3, ethercat wizard):= Data access time, Min - Min. Synchdelay

Die Synchronisationsverzögerungen können mit folgender Formel geschätzt werden (gemäß Diagramm 2 - Telegramm-Sendezyklus):

Min. Synchdelay = CycleStart + Min.Cycletime + Min. Jitter + TelegramOnWire - Shifttime

Max. Synchdelay = CycleStart + Max.Cycletime + Max. Jitter + TelegramOnWire - Shifttime

Der geschätzte CycleStart Offset (Event a, Diagramm 2) kann mit der folgenden Formel berechnet werden:

*CycleStart = - SynchOffset *Cycletime*

Zusätzlich kann die Breite der Telegrammankunft (siehe Diagramm 3, EtherCAT-Assistent) nach der Formel berechnet werden:

Telegramarrival Jitter= |Max. Synchdelay – Min. Synchdelay|

Beispiel:

SynchOffset:	20%:
Cycletime:	4ms
Data access time, Max (S6K):	150µs
Data access time, Min (S6K):	50µs
TelegramOnWire:	50µs
Max. Cycletime:	581µs
Min. Cycletime:	529µs
Min. Jitter:	-4µs
Max. Jitter:	+4µs
Shifttime (default)	0µs

*CycleStart = - SynchOffset *Cycletime*

*CycleStart := - 0.2*4000µs:= -800 µs(before synch):= 3200µs (after previous synch)*

Max. Synchdelay = CycleStart + Min.Cycletime + Min. Jitter + TelegramOnWire -Shifttime

Max. Synchdelay = -800 µs + 529 µs + (-4) µs + 50 µs + 0 µs = -225µs

Min. Synchdelay = CycleStart + Max.Cycletime + Max. Jitter + TelegramOnWire - Shifttime

Min. Synchdelay = -800 µs + 581 µs + 4 µs + 50 µs + 0 µs = -165 µs

Telegramarrival Jitter= |Max. Synchdelay – Min. Synchdelay|= |-225 µs – (-165 µs)| = 60 µs

Tolerance time 1 (see diagram 3, ethercat wizard):= Cycletime - Max. Synchdelay- Data access time, Max

Tolerance time 1 = 4000µs – (-150µs) – 225µs= 3625µs

Tolerance time 2 (see diagram 3, ethercat wizard):= Data access time, Min - Min. Synchdelay

Tolerance time 2 = 50µs – -165µs= 310 µs

Interpretation/ Überschneidungsgefahr:

Die Telegrammannahme und der Bereich des Slave-PDO-Zugriffs überschneiden sich nicht.

Ein synchroner Betrieb ist möglich, wenn der Anwendungsjitter stabil bleibt. Die Toleranzzeit 2 ist gering und sollte durch Verschieben des Telegrammannahmebereichs optimiert werden.

Messung

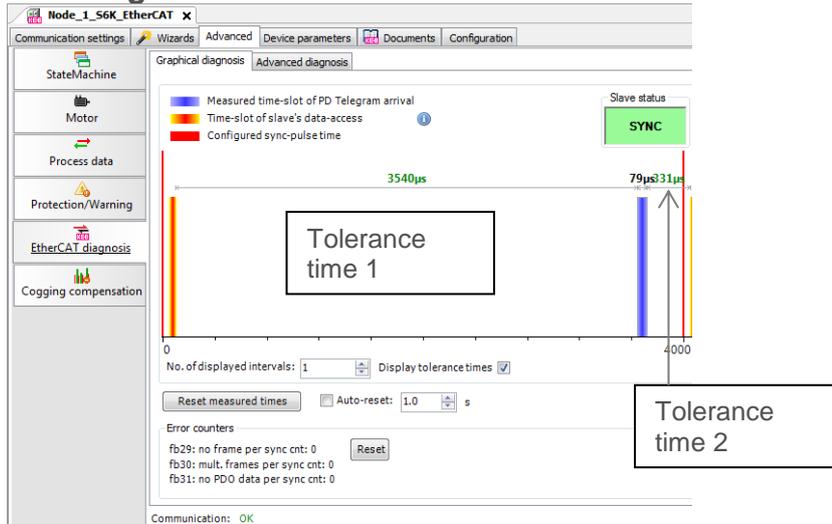


Diagramm: 4 Gemessenen Toleranz Zeit

Fb27: synch offset, Min: 0µs (bezogen auf die Datenzugriffszeit)

Fb28: synch offset, Max: 3672µs (bezogen auf die Datenzugriffszeit)

Toleranz Zeit 1, actual (see diagram 4) = **3540 µs**

Toleranz Zeit 2, actual (see diagram 4):= **331 µs**

Telegramarrival jitter:= 79µs

Interpretation/ Überschneidungsgefahr: Der Bereich der Telegrammannahme und der Bereich des Slave-PDO-Zugriffs überschneiden sich nicht, die Fehlerzähler sind stabil bei 0, ein synchroner Betrieb ist möglich, wenn der Applikations-Jitter stabil bleibt. Die Toleranzzeit 2 ist gering und sollte durch Verschieben des Telegrammannahmebereichs optimiert werden.

Optimierung der Toleranzzeiten:

Der Bereich der Telegrammankunft kann optimiert werden, indem er in die Mitte von 2 zeitlichen Ausschnitten des Datenzugriffs des Slaves verschoben wird.

Toleranz Zeit, opt. = (Tolerance time 1 + Tolerance time 2)/2

Toleranz Zeit, opt. = (3540+331)/2 = 1935 µs

SynchOffset, opt.: = SynchOffset, actual + (Toleranzzeit, opt - Toleranzzeit, aktuell)/Zykluszeit

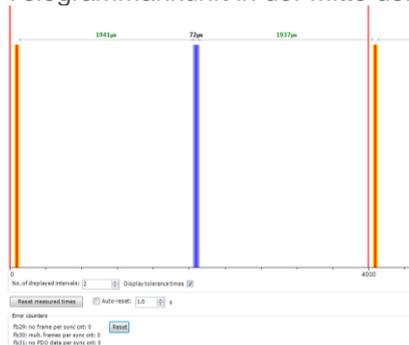
SynchOffset, opt.: =20% + (1935µs - 331µs)/ 4000µs =20%+40%= 60% (nach synch)

Der Bereich von SynchOffset. ist [-50..+50], daher muss der Wert 60% in einen negativen Offset umgewandelt werden.

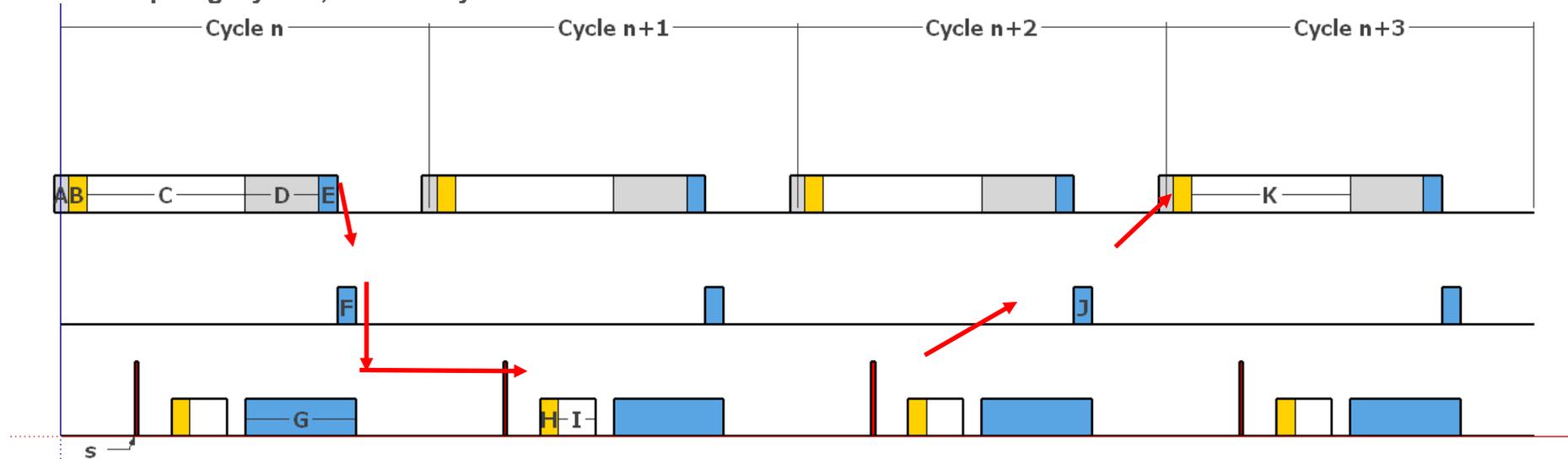
SynchOffset, opt.: = 60%-100%= -40% (vor synch)

Messungsergebnis

Telegrammankunft in der Mitte der Zugriffsbereiche des Slave-Pdo.



Sende-/Empfangszyklus, 4 Masterzyklen



C: PDO-Daten n werden im SPS-Zyklus berechnet.

F: Daten n auf der Leitung. Die Daten kommen beim Slave mit der fallenden Flanke dieses Bereichs an.

G: Telegramm-Eingangsbereich. Der Bereich jittert in Abhängigkeit von der Masterzyklusposition und dem Jitter.

H: Der PDO-Puffer wird vom Slave gelesen/geschrieben.

I: Die Daten n werden von der Slave-Firmware verarbeitet.

J: Daten n auf Draht in Richtung SPS

K: Daten werden in der SPS neu gelesen.

Zusammenfassung: Frame bei Task start=false

Vorteil:

Ein Schreib-/Lesezyklus wird auf 4 Zyklen reduziert. Bei optimiertem Synchronoffset ist eine schnelle Reaktion auf berechnete Werte auf dem Slave möglich.

Nachteil:

Die Telegrammankunft wird direkt von der Laufzeit des Anwenderprogramms beeinflusst (Zykluszeit, min, Zykluszeit, max., Jitter, Min, Jitter, Max). Dies kann den Synchronbetrieb gefährden, wenn die Ausführungszeit des Anwenderprogramms nicht konstant ist.

Es wird empfohlen, die Telegrammankunftszeit relativ zur Slave-Telegrammzugriffszeit für jede Anwendung separat zu berechnen oder zu messen! Mit dem Master SynchOffset sollte die Telegrammankunft in einen Speicherbereich verschoben werden.

Timing (Frame at task start=true)

Sendezyklus, 2 Masterzyklen

Der IoDrvEtherCAT-Stack ab Version 3.5.3.50 ermöglicht das Senden des Frames am Anfang des SPS-Zyklus (vor dem IEC-Anwenderprogramm). Dadurch ändert sich die Reihenfolge des Sendens und Empfangens von Telegrammen. Die Abhängigkeit der Telegrammposition des IEC-Anwenderprogramms kann vermieden werden.

Das Diagramm 5 zeigt, wie Daten vom EtherCAT-Mastergerät an ein Slave-Gerät gesendet werden. Die dargestellte Sequenz gilt für synchrone Operationen und die Master-Einstellung "FrameAtStart=true". So werden beispielsweise Daten n im Zyklus n (Bereich C) berechnet, am Anfang des nächsten Zyklus n+1 (Bereich B, F) per Telegramm gesendet und gelangen in die Slave-Feldbushardware (Bereich G). Nach dem nächsten Synchronsprung (Ereignis s) werden die verfügbaren Daten mit der Slave-Firmware (Bereich H) ausgetauscht und in der slavespezifischen Steuersequenz (Bereich I) berechnet. Die Position der Telegrammankunft (Bereich G) ist abhängig von der eingestellten Zykluszeit, dem eingestellten Synchron-Offset, weiteren Zeitkonstanten und Jitter in Abhängigkeit von Hardware und verwendeter Software-Stack.

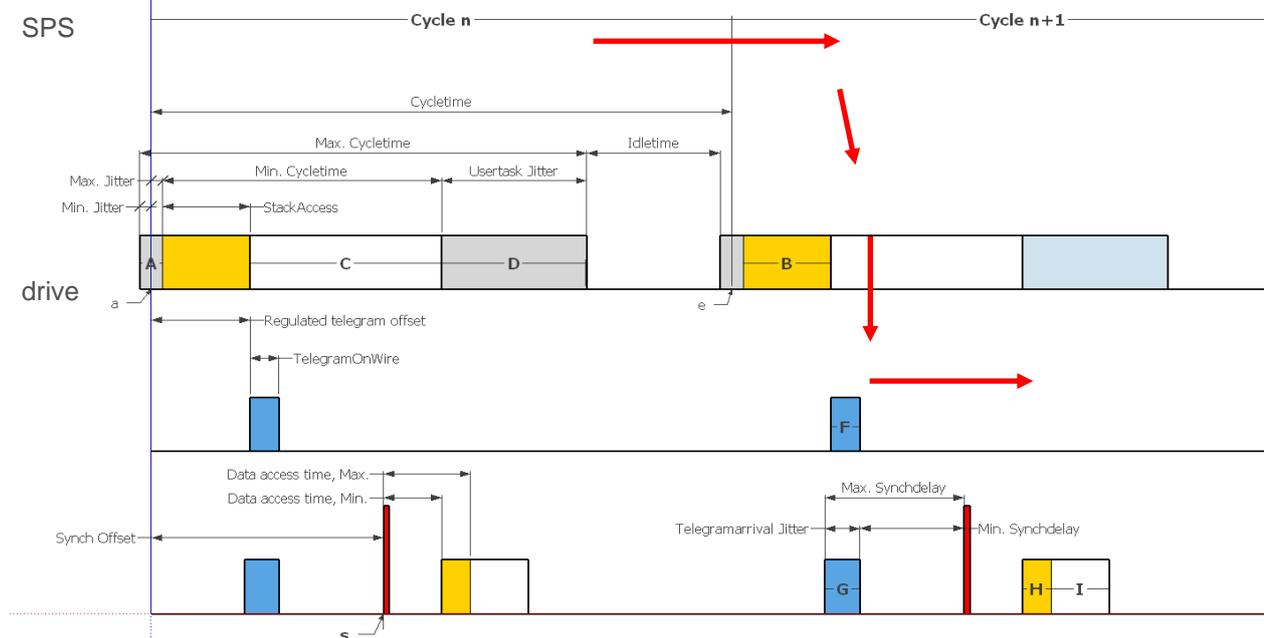


Diagramm 5 - Sendezyklus, frame beim Start

- a** ein durchschnittliches Zyklus-Startereignis, Ausführungsbeginn des SPS-Zyklus n
- e** Durchschnittliches Zyklusende n, Ausführungsbeginn des SPS-Zyklus n+1
- A** Ein Jitter des SPS-Task Startereignis.
- B** Stack zugriff zum Lesen der Daten
- C** Ausführung des Benutzercodes. Interpretation der aktuellen Prozesswerte und Berechnung der Sollwerte zur Steuerung eines technischen Prozesses.
- D** Jitter der Benutzercodeausführungszeit
- E** Stack-Zugriff zum Schreiben von Daten und Senden des Ethercat-Telegramms.

- Cycletime** Konfigurierte EtherCAT-Bus Zykluszeit (Einstellung von EtherCAT-Master / EtherCAT-Task).
- Max. Cycletime:** Maximale gemessene Laufzeit des SPS-Zyklus.
- Min. Cycletime:** Minimale gemessene Laufzeit des SPS-Zyklus.
- Idletime:** Puffer-/Leerlaufzeit am Ende der SPS-Task bis zum Start des nächsten Zyklus.
- Max. Jitter** Max. Jitter Jitter der SPS-Task, Maximalwert.
- Min. Jitter** Min. Jitter Jitter der SPS-Task, Minimalwert.
- Stack Access** Stack-Zugriffszeit, die für die Stack-Verarbeitung und das Lesen von Prozesseingangsdaten (einschließlich Stapel) benötigt wird. related offset time)

Berechnung

Die Berechnung zeigt, wie man die Telegrammankunftsposition **ohne** Einsatz des EtherCAT-Assistenten schätzen kann.

Dieser Schritt kann übersprungen werden, wenn Messdaten aus dem EtherCAT-Assistenten zur Verfügung stehen.

Beispiel:

SynchOffset:	20%:
Cycletime:	4ms
Data access time, Max (S6K):	150µs
Data access time, Min (S6K):	50µs
TelegramOnWire:	50µs
Min. Jitter:	-40µs
Max. Jitter:	+40µs
StackAccess C6C2	100µs
Shifttime	0µs (default)

$$\text{CycleStart} = - \text{SynchOffset} * \text{Cycletime}$$

$$\text{CycleStart} := - 0.2 * 4000 \mu\text{s} := -800 \mu\text{s (before synch)} := 3200 \mu\text{s (after previous synch)}$$

$$\text{Max. Synchronisationsverzögerung} = \text{CycleStart} + \text{StackAccess} + \text{Min. Jitter} + \text{TelegramOnWire} - \text{Shifttime}$$

$$\text{Max. Synchronisationsverzögerung} = -800 \mu\text{s} + 100 \mu\text{s} + (-40) \mu\text{s} + 50 \mu\text{s} - 0 = -690 \mu\text{s}$$

$$\text{Min. Synchronisationsverzögerung} = \text{CycleStart} + \text{Max. Jitter} + \text{TelegramOnWire}$$

$$\text{Min. Synchronisationsverzögerung} = -800 \mu\text{s} + 100 \mu\text{s} + 40 \mu\text{s} + 50 \mu\text{s} - 0 = -610 \mu\text{s}$$

$$\text{Telegrammankunfts Jitter} = |\text{Max. Synchdelay} - \text{Min. Synchdelay}| = |-690 \mu\text{s} - (-610 \mu\text{s})| = 80 \mu\text{s}$$

$$\text{Tolerance time 1 (see diagram 4, ethercat wizard)} := \text{Cycletime} - \text{Max. Synchdelay} - \text{Data access time, Max}$$

$$\text{Tolerance time 1} = 4000 \mu\text{s} - (690 \mu\text{s}) - 150 \mu\text{s} = \mathbf{3210 \mu\text{s}}$$

$$\text{Tolerance time 2 (see diagram 4, ethercat wizard)} := \text{Data access time, Min} - \text{Min. Synchdelay}$$

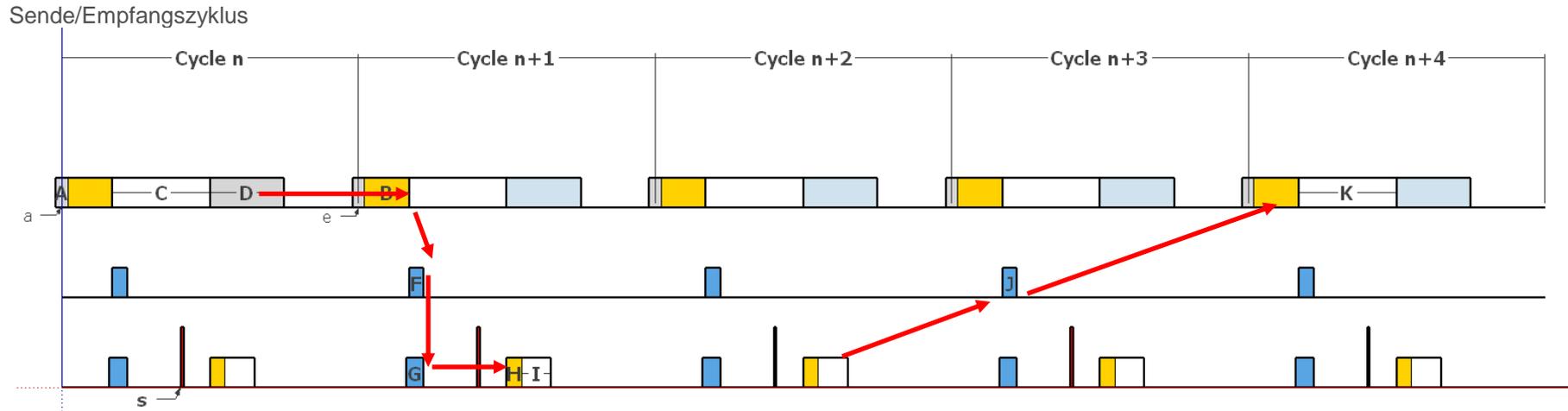
$$\text{Tolerance time 2} = 610 \mu\text{s} - (-50 \mu\text{s}) = \mathbf{660 \mu\text{s}}$$

Messung



Interpretation:

Die Telegrammankunft ist ein sicherer Bereich und hat einen akzeptablen Jitter. Da die Ankunft nicht von der Laufzeit der User-Task beeinflusst wird, kann diese Einstellung für synchrone Operationen verwendet werden.



C: PDO-Daten n werden im SPS Zyklus berechnet.

F: Daten n auf der Leitung. Die Daten kommen beim Slave mit der fallenden Flanke dieses Bereichs an.

G: Telegram Ankunftsbereich.

H: Der PDO-Buffer wird vom Slave gelesen/ geschrieben.

I: Die Daten n werden von der Slave Firmware verarbeitet.

J: Daten n auf der Leitung in Richtung SPS.

K: Daten werden in der SPS neu gelesen.

Zusammenfassung: Frame bei Task start=true

Vorteil:

Die Telegrammankunft wird durch die Dauer des Anwenderprogramms nicht beeinflusst. Der Start der Stack-Ausführung wird streng kontrolliert, so dass ein kleiner Jitter zu erwarten ist.

Ist der hardwareabhängige Offset bekannt, kann die Position der Telegrammankunft ohne Messung der Anwenderprogrammlaufzeit vorberechnet werden. Die Laufzeit des Anwenderprogramms kann wesentlich flexibler verkürzt/erhöht werden, ohne die Telegrammannahme erneut zu überprüfen. Auf jeden Fall sollte die Laufzeit des Anwenderprogramms überprüft werden, um Zykluszeitüberläufe zu vermeiden.

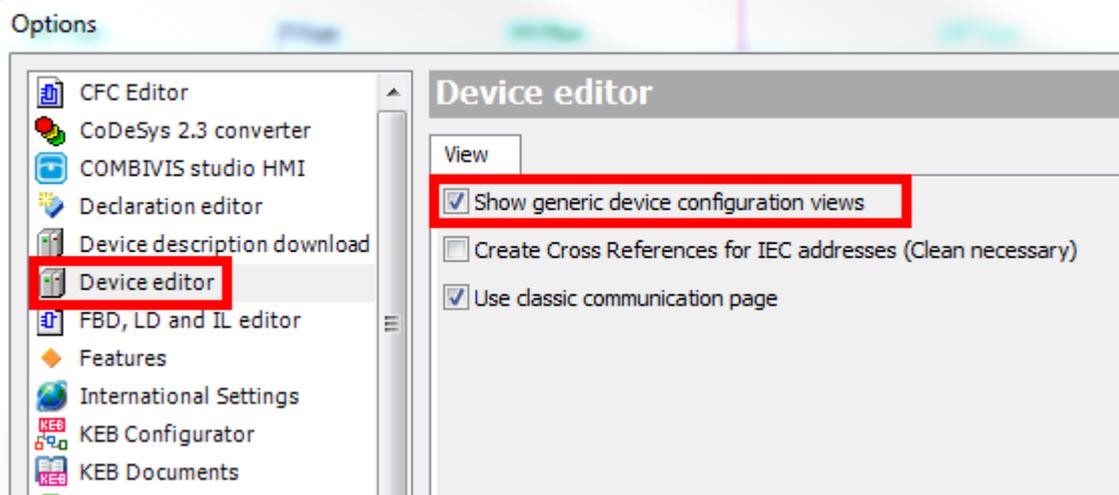
Nachteil:

PDO werden um einen Zyklus verzögert gesendet und empfangen. Eine Schreib-/Leseschleife dauert 5 Zyklen.

Wie kann ich die Einstellungen für das Frame-Sendeereignis ändern?

Zuerst muss die generic device Editoransicht aktiviert werden.

Tools→Options→Device Editor→Show generic device configuration views



Nun kann die Eigenschaft mit dem EtherCAT Mastergerät geändert werden.

EtherCAT master→ EtherCAT Configuration → FrameAtTaskStart

Parameter	Type	Value	Default Value	Unit	Description
Autoconfig	DWORD	1	1		Automatic configuration
MasterCycleTime	DWORD	4000	4000		Master Cycle Time
MasterUseLRW	BOOL	FALSE	FALSE		Master uses LRW command
SlaveAutorestart	BOOL	FALSE	FALSE		Slave restarts automatically
SlaveCheckMode	USINT	0	0		Mode for vendor product check
NetworkName	STRING(100)	'Fieldbus port (X6C)'			Name of the network card
SelectNetworkByName	BOOL	FALSE	FALSE		Select network by name
EnableTaskMessage	BOOL	FALSE	FALSE		Enable transmission per task
DisableTaskGeneration	BOOL	FALSE	FALSE		Disable automatic task generation
FrameAtTaskStart	BOOL	TRUE	FALSE		Send frame at task cycle start
ScanForAliasAddress	BOOL	TRUE	TRUE		Enables scan for alias address
SyncOffset	SINT	20	20		Master synchronisation offset



Anhang

Tabelle vom KEB PLC timing

Name	Version	Stack	Stack Access time [µs]
C6C2	3.4.1.7	3.5.3.50	100
C6 SMART	3.5.6.60	3.5.6.40	100

Tabelle vom KEB DRIVE timing

Name	Version	Data access, Min [µs]	Data access, Max [µs]
H6	1210,1211,1220,1221,1230,1231,1240,1241,1250,1251	50	100
S6K	9241,9242	50	100



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