6.5 TTP (Time Triggered Protocol)

TTP/C ist ein zeitgesteuertes Kommunikationsprotokoll, das speziell für den Einsatz in sicherheitskritischen Anwendungen, wie z.B. im Flugzeugbereich oder für X-by-Wire-Anwendungen im Automobil geeignet ist. Mit diesem Protokoll können die höchsten Sicherheitsanforderungen erfüllt werden. Zudem wird es mit diesem Protokoll möglich, Komponenten eines Systems einzeln zu entwickeln und zu testen, und dann zusammenzufügen.

6.5.1 Grundprinzipien der Kommunikation

Das Protokoll arbeitet nach dem TDMA-Verfahren. Die Kommunikation des Hostprozessors mit dem Bus findet ausschließlich über den TTP/C-Communication-Controller (Abb. 6-19 : Überblick über einen Knoten [4]) statt. Im Gegensatz zu klassischen, ereignisgesteuerten Bussystemen wie z.B. CAN, kommunizieren beim TTP-Protokoll alle angeschlossenen Knoten ununterbrochen in vordefinierten Zeitabständen.

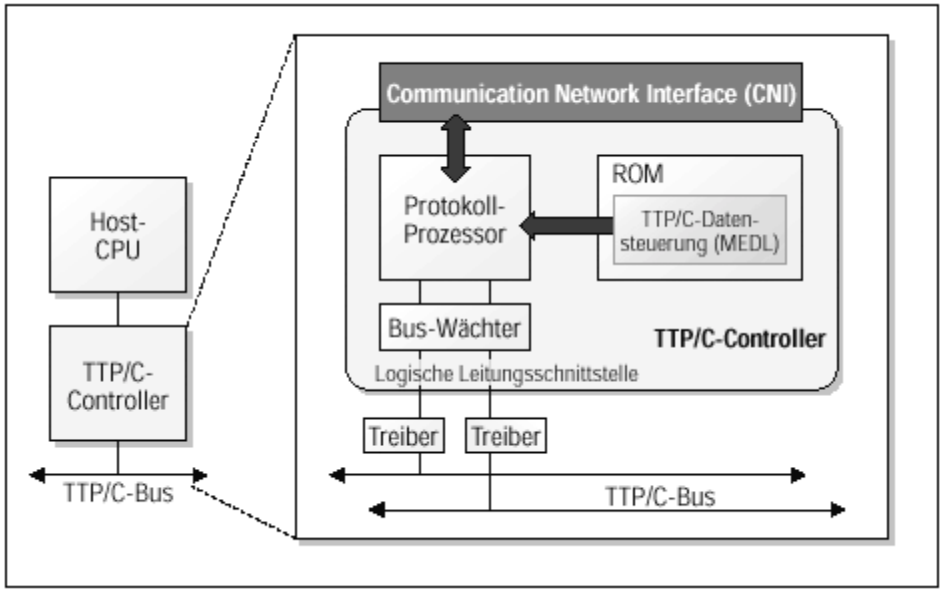

Abb. 6-19 : Überblick über einen Knoten [4]

6.5.2 MEDL

Während der Initialisierungsphase werden die notwendigen Kontrollinformationen in der lokalen *Message Descriptor List* (MEDL) des Controllers gespeichert [6]. Somit

kann der TTP/C-Controller nach der Initialisierungsphase völlig selbständig arbeiten, ohne Kontrollsignale vom Host zu erhalten.

In der MEDL sind die folgenden Daten enthalten:

- für zu sendende Nachrichten ist der Sendezeitpunkt (Sendeinstanz) und die Adresse in der CNI enthalten, wo die Daten abgeholt werden müssen.
- für zu empfangene Nachrichten ist der Empfangszeitpunkt (Empfangsinstanz) und die Adresse in der CNI enthalten, wo die Daten abgelegt werden müssen.
- zusätzliche Informationen für den Protokoll-Kontrollfluss.[2]

Alle Aktionen, die in dem System durchgeführt werden sollen, sind in der MEDL gespeichert. In der MEDL ist sogar gespeichert, wann welche Daten zu senden sind. Für unterschiedliche Modi gibt es dabei auch verschiedene MEDLs.

6.5.3 Clock-Synchronisation

Um den richtigen Sendezeitpunkt ermitteln zu können, ist eine verteilte Zeitbasis mit regelmäßiger Uhrensynchronisation notwendig. Statt eines Busmasters für die Zeitsynchronisation wird ein dezentraler Algorithmus verwendet. Der TTP/C-Controller führt diese Uhrensynchronisation autonom durch. Um die benötigte Genauigkeit der Zeitbasis zu erreichen, werden die bekannten Sendezeiten aus der MEDL verwendet. Da jeder Busteilnehmer weiß, wann er Nachrichten empfangen soll, kann er ein Empfangsfenster um diesen Zeitraum spannen. Aus der Differenz von erwartetem und tatsächlichem Empfangszeitpunkt kann die für die Generierung der Zeitbasis notwendige Korrektur errechnet werden.

6.5.4 Start der Kommunikation

Beim Hochfahren des Systems gibt es noch keine globale Zeit, da nicht alle Knoten zur selben Zeit aktiv sind. Jedem Knoten ist eine bestimmte Zeit bis zum Time-out in der MEDL zugewiesen worden. Auf dem Bus findet zunächst keine Kommunikation statt, bis bei dem ersten Knoten ein Time-out gemeldet wird. Dieser Knoten startet dann die Kommunikation durch Senden eines I-Frames. Die Startphase ist der einzige Zeitpunkt, in den Kollisionen auf dem Bus auftreten können [5].

6.5.5 Frames

Es gibt zwei verschiedene Arten von Frames: Den I-Frame, der zur Initialisierung und Resynchronisation von Controllern verwendet wird, und den N-Frame zum Senden der Daten.

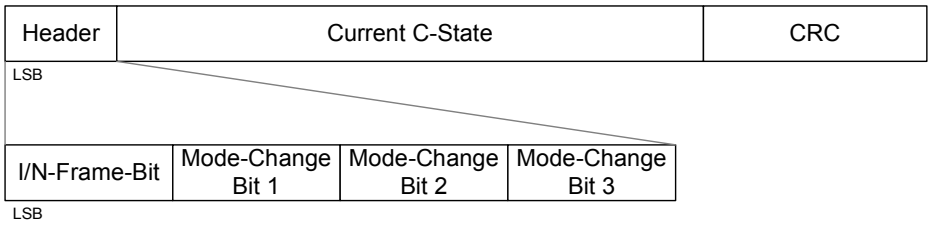

Abb. 6-20 : I-Frame nach [3]

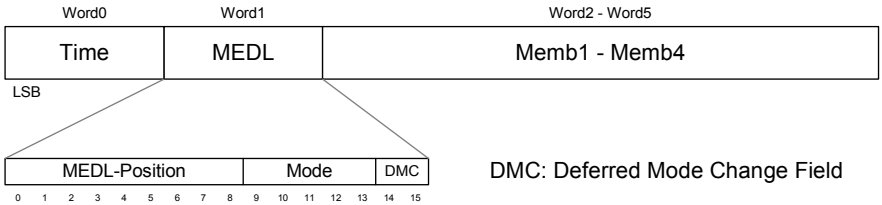

Abb. 6-21: C-State nach [3]

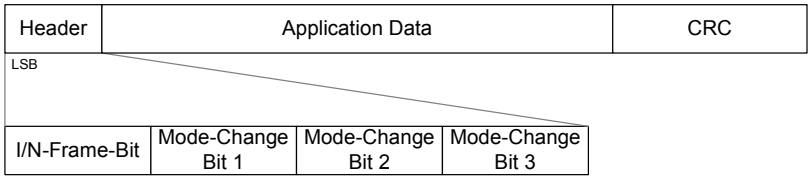

Abb. 6-22: N-Frame nach [3]

6.5.6 Media Access

Eine TDMA-Round besteht aus so vielen Zeitslots, wie SRUs im System vorhanden sind. Während des einer SRU zugewiesenen Zeitslots kann die SRU die geforderten Daten senden. Im System beinhaltet ein Cluster-Cycle alle zu sendenden Nachrichten.

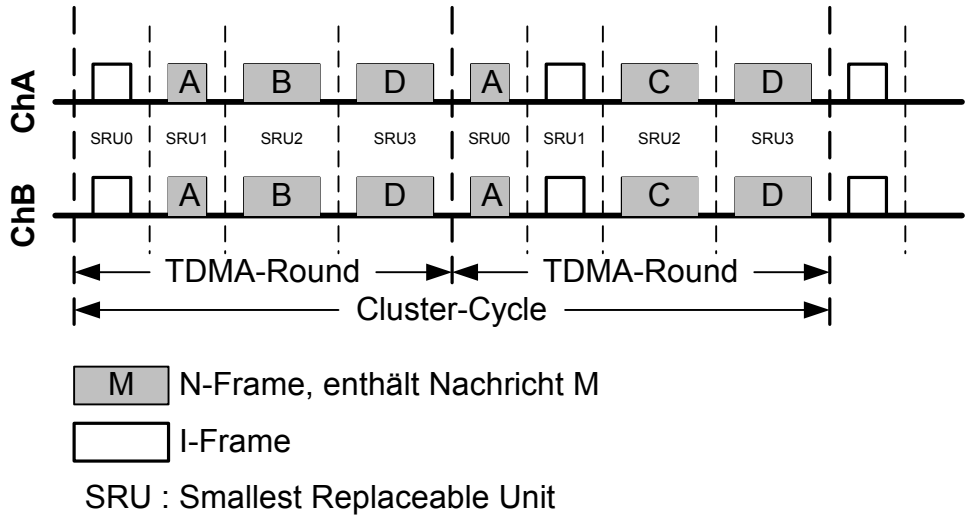

Abb. 6-23: Aufteilung eines Cluster Cycle in TDMA Rounds und SRU Slots nach [3]

6.5.7 Communication Network Interface (CNI)

Das CNI ist die Kommunikationsschnittstelle zwischen dem TTP/C-Controller und dem Hostprozessor. Es handelt sich dabei um eine reine Datenschnittstelle, über die keine Steuersignale gesendet werden: sie wirkt wie eine „temporäre Firewall“ [1]. Der Hostprozessor hat keine Möglichkeit, das zeitliche Verhalten des Kommunikationssystems zu beeinflussen.

6.5.8 Buswächter

Der Buszugriff erfolgt im TDMA-Verfahren. Somit ist sichergestellt, dass jeder TTP/C-Controller nur in dem ihm zugewiesenen Intervall Nachrichten auf den Bus senden kann. Ein im TTP/C-Controller enthaltener Bus-Wächter (Abb. 6-24 : Funktionsprinzip des Buswächters [1]) stellt diese Übertragung sicher. Somit wird auch eine Überlastung des Busses durch einen „Babbling Idiot“ vermieden.

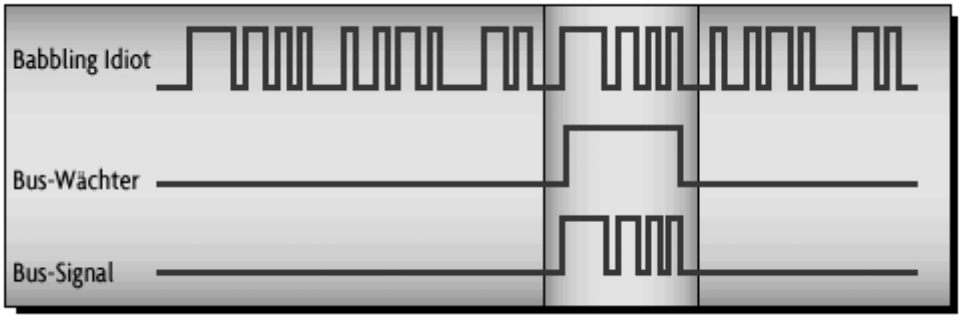

Abb. 6-24 : Funktionsprinzip des Buswächters [1]

6.5.9 Sicherheit und Fehlertoleranz

Um die Forderung nach Fehlersicherheit zu erfüllen, können zwei TTP/C-Controller, A und B parallel betrieben werden. Die beiden Controller bilden zusammen eine sogenannte fehlertolerante Einheit (FTU). Da die Nachrichtenübertragung zudem über zwei redundante Busse erfolgt, ist diese somit vierfach redundant vorhanden.

Jeder Busteilnehmer eines TTP/C-Netzwerks hat einen lokalen Membership-Vektor mit der *Membership*-Information für jeden anderen Busteilnehmer. Dadurch ist Fehlererkennung sowohl im Sender als auch im Empfänger möglich, weil jedem Knoten alle Empfangszeitpunkte bekannt sind. Kommt eine Nachricht nicht in dem angegebenen Zeitfenster an, so wird ein Fehler erkannt [3]. Alle Komponenten überprüfen kontinuierlich ihren Zeit- und Wertebereich selbst. Erkennt diese Komponente dabei eine Abweichung vom vorgegebenem Ablauf, so muss sich diese Komponente nach außen still verhalten (*Fail-Silent*) [2].

Fehlererkennung im Wertebereich wird durch die CRC-Kalkulation über den C-State, den Nachrichten-Header und die Nachricht selbst realisiert.

6.5.10 System Konfiguration

Es gibt zwei verschiedene mögliche Systemkonfigurationen:

6.5.11 Bus Konfiguration

Bei Verwendung der Bus Konfiguration hat jeder Knoten seinen eigenen Buswächter, so dass ein Schreibzugriff auf den Bus nur zu in der MEDL definierten Zeitpunkten möglich ist. Diese Konfiguration garantiert eine kollisionsfreie Kommunikation.

6.5.12 Stern Konfiguration

Bei der Stern-Konfiguration wird ein zentraler Buswächter verwendet. Jeder der zwei Übertragungskanäle hat dabei einen zentralen Buswächter. Durch die Stern-Konfiguration kann eine Aufbereitung der Bussignale durchgeführt werden. Ein weiterer Vorteil der Stern-Konfiguration ist, dass bei Zerstörung eines Knotens, z.B. durch Feuer, der Buswächter nicht betroffen ist, da dieser sich dezentral an einem anderen Ort befindet und somit den defekten Knoten abschalten kann [6].

6.5.13 Zusammensetzbarkeit (Composability)

Ein weiterer wichtiger Aspekt ist die Zusammensetzbarkeit von verschiedenen Knoten. Ist Zusammensetzbarkeit gegeben, so können die einzelnen Komponenten des Systems unabhängig voneinander entwickelt und getestet werden. Zudem können an einem Steuergerät Änderungen durchgeführt werden, ohne dass diese einen Einfluss auf die anderen Komponenten des Systems haben.

Mit TTP/C wird es ermöglicht, jeden Knoten einzeln gegen das CNI zu testen. Zusammensetzbarkeit wird garantiert und ermöglicht es so, Zeitaufwand und Kosten für Test- und Systemintegration drastisch zu senken.

6.5.14 Asynchroner Modus

TTP/C unterstützt auch das Senden von ereignisgesteuerten Nachrichten. Dafür ist eine vorher spezifizierte Anzahl von Bytes in einer Nachricht für ereignisgesteuerte Nachrichten vorzusehen. Mit diesen reservierten Bytes kann ein ereignisgesteuertes Protokoll, wie z.B. CAN, so verwendet werden, dass der Knoten, der auf den Bus senden kann, Nachrichten nach CAN-Standard übermitteln kann. Die CAN-Software kann dabei mit nur geringen Änderungen verwendet werden. Dieser Modus stört dabei das TTP/C-Protokoll in keiner Weise [6].

6.5.15 Robustheit

Die Robustheit des TTP/C-Netzwerkes wird durch eine Verkabelung entsprechend der CAN-Spezifikation , durch Manchester Codierung und durch spezielle Hardware, wie z.B. High Speed CAN-Treiber mit integrierter Fehlertoleranz [3].

6.5.16 TTP/A

TTP/A ist die kostengünstige Variante des TTP/C, erfüllt aber auch nicht die harten Echtzeitbedingen der Class C. TTP/A ist ein Master-Slave-UART-basiertes Protokoll in dem der Master- der zugleich ein Teilnehmer am TTP/C-Bus ist- den Zeittakt vorgibt [1]. Der Master spricht über Polling die einzelnen Slaves an. Die Slaves antworten nur, wenn das von Ihnen gefordert wird. Die Kommunikation zwischen den Slaves kann nur über den Master stattfinden.

6.5.17 TTP/C vs. Flexray

Der FlexRay-Bus bietet, ebenso wie TTP eine deterministische Nachrichtenübertragung, wie auch eine Uhren-Synchronisation, aber der Controller bewirkt keine Nachrichten-Konsistenz, die dafür sorgt, dass alle Knoten wissen, ob Nachrichten angekommen sind oder nicht. Dieser sogenannte „Membership-Service“ muss bei FlexRay per Middleware vom Anwender programmiert werden [7]. Da davon aber die Sicherheit des gesamten Systems abhängt, ist es gefährlich, diesen Service für jede Anwendung neu zu entwickeln.

Literatur

[1] Poledna, S.; Stöger, G.; Schlatterbeck, R.; Niedersüß, M.: Sicherheit auf vier Rädern; Elektronik 10/2001

[2] Dilger, E.; Führer, T.; Müller, B.; Poledna, S.; Thurner, T.: X-by-Wire: Design von verteilten, fehlertoleranten und sicherheitskritischen Anwendungen in modernen Kraftfahrzeugen; www.vmars.tuwien.ac.at/projects/xbywire/projects/new-vdifinal.html

[3] Specification of the TTP/C Protocol; www.tttech.com/specrequest.shtml

[4] Poledna, S.; Kroiss, G.: TTP: “Drive by Wire” in greifbarer Nähe; Elektronik 14/99

[5] Poledna, S.; Ettlmayr, W.; Novak, M.: Communication Bus for Automotive Applications

[6] Kopetz, H.: A Comparison of TTP/C and FlexRay; TU Wien Research Report 2001/10

[7] Klasche, G: TTP und FlexRay: Wann kommt die Einigung? Elektronik Automotive September 2001