

5.11.3.4 Die Kommunikationsquittierung der Alarmbenachrichtigung

Mit dem RTA_ACK quittiert das Device auf Protokollebene den Empfang eines RTA_Data (Alarm_Ack). Diese Quittierung sagt nur aus, dass Ressourcen für eine weitere Nachricht zur Verfügung stehen und dass keine Wiederholung mehr erforderlich ist.

Der Frame-Aufbau ist wie folgt (siehe hierzu auch Kapitel 18.7):

VLAN	Ether-type	Frame_ID	RTA Header	FCS
4	2	2	12	4

Erklärung der relevanten Daten im Frame

Siehe Alarm-Benachrichtigung (Alarm Notification).

5.12 Der azyklische Datenverkehr im Detail (Record Data-CR)

PROFINET IO bietet die Möglichkeit, neben dem zyklischen Datenverkehr zur Übertragung der Prozess-Daten bestimmte Daten auch azyklisch zwischen einem Initiator und Responder über die Record Data CR auszutauschen. Azyklische Daten werden als niederprioritäre Daten übertragen, die nur bei Bedarf ausgetauscht werden. Dies erfolgt über den NRT-Pfad mit dem RPC-Protokoll. Hierzu stellt PROFINET IO das Lesen (Read-Service) und das Schreiben (Write-Service) von Daten zur Verfügung.

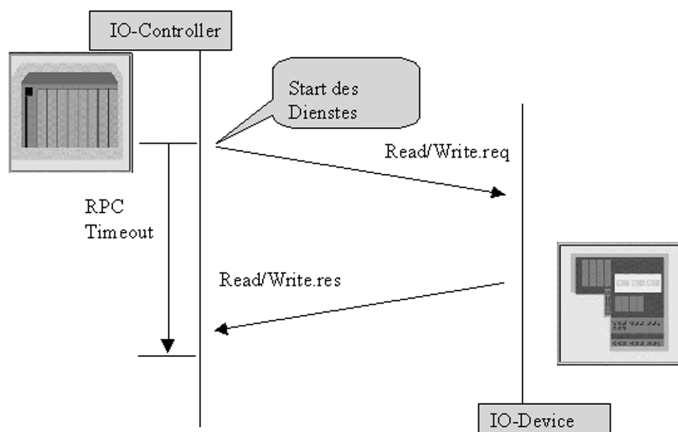


Bild 5.21: Der Ablauf der azyklischen Datenübertragung wird zeitlich über den RPC-Timeout überwacht.

Der Ablauf der azyklischen Kommunikation besteht aus zwei Frames, einem Aufruf (Request) und einer Antwort (Response).

Folgende Werte gelten bei der Adressierung von Record Data Services:

Slot 0 ... 0x7FFF

Subslot 1 ... 0x7FFF

Index 0 ... 0x7FFF.

Typische Anwendungsfälle beim azyklischen Datenverkehr sind beispielsweise

- Auslesen von Diagnosepuffern für Wartungszwecke,
- Lesen der Konfigurationsdaten zum Überprüfen des Ausbaugrads,
- Lesen der gerätespezifischen Daten (I&M-Funktionen),
- Lesen der Logbuch-Einträge für statistische Datenauswertungen,
- Lesen von Ein- und Ausgängen zur Prozesskontrolle,
- Lesen der PDev-Daten
- Schreiben von Baugruppen-Parametern zu Testzwecken.

Write-Aufrufe sind nur innerhalb einer aufgebauten CR erlaubt. Read-Aufrufe dürfen auch ohne vorherigen Verbindungsaufbau durchgeführt werden, da der Prozess dadurch nicht beeinflusst wird. Welche Daten letztendlich gelesen oder geschrieben werden sollen, spezifiziert das Indexfeld (siehe auch *Kap. 18.13, Tabelle 18.1*) im Read- oder Writeblock innerhalb der 'Data Unit'.

5.12.1 Read Request

Read-Aufrufe können während des zyklischen Datenaustauschs zu jedem Zeitpunkt durchgeführt werden. Mit dem 'Read.req' fordert ein Initiator Daten von einem Device an.

Der Frame-Aufbau ist wie folgt:

VLAN	EtherType	IP/UDP	RPC	NDR	Read Block	FCS
4	0x0800	28	80	20	64	4

Opt.
Feld

Erklärung der relevanten Daten im Frame

Im Read Block (Kennung 0x00, 0x09) spezifiziert der Initiator die zu lesenden Daten anhand der ARUUIID, API und Slot/Subslot und Index (siehe auch *Kap. 18.13, Tabelle 18.1*).

5.12.2 Read Response

Mit der Read Response werden die angeforderten Daten zeitnah übergeben. Die Überwachungszeit, bis wann ein Read.req beantwortet werden muss, entspricht der RPC-Überwachungszeit.

Der Frame-Aufbau ist wie folgt:

VLAN	Ethertype	IP/UDP	RPC	NDR	Read Block	Read Data	FCS
4	0x0800	28	80	20	64	...	4

Opt.
Feld

Erklärung der relevanten Daten im Frame

Der Read Block (Kennung 0x80, 0x09) enthält im Wesentlichen die Slot und Subslotnummer sowie den Index, der die zu lesenden Daten spezifiziert. Im nachfolgenden Read Data-Feld überträgt das Feldgerät die angeforderten Daten.

5.12.3 Write Request

Mit einem *'Write.req'*-Dienst kann ein IO-Controller Daten in ein Feldgerät schreiben. Dies ist allerdings nur möglich, wenn eine *'Record Data-CR'* vorher aufgebaut wurde.

VLAN	Ethertype	IP/UDP	RPC	NDR	Write Block	Write Data	FCS
4	0x0800	28	80	20	64	...	4

Opt.
Feld

Erklärung der relevanten Daten im Frame

Write Block (Kennung 0x00, 0x08) enthält im Wesentlichen die eindeutige Identifikation für die übermittelten Daten oder unterschiedliche Datenbereiche dieser Verbindung (z.B. Angabe eines Application Profile Identifier (API) wie beispielsweise PROFIsafe oder PROFIdrive) sowie die Adressierung des Objektes und dessen Länge. Die Kodierung erfolgt im Big-Endian-Format.

Write Data Im *'Write Data'*-Bereich spezifiziert der Initiator die zu schreibenden Daten anhand der ARUUIID, API und Slot/Subslot und Index (siehe auch *Kap. 18.13 und Tabelle 18.1*).

5.12.4 Write Response

Jeder *'Write.req'-Frame* wird seitens eines IO-Devices mit einem *'Write.res'-Frame* quittiert. Fehlermeldungen können darin übermittelt werden.

VLAN	Ethertype	IP UDP	RPC	NDR	Write Block	FCS
4	0x0800	28	80	20	64	4

opt
Feld

Erklärung der relevanten Daten im Frame

Write Block (Kennung 0x80, 0x08) quittiert anhand der API, Slot, Subslot und Index (siehe auch *Kap. 18.13 und Tabelle 18.1*) die geschriebenen Daten.

5.13 Der PROFINET IO-Controller

Ein PROFINET IO-Controller ist die Station innerhalb einer Automatisierungsanlage, in der das Steuerungsprogramm abläuft. Er fordert die Prozess-Daten (Eingänge) von den projektierten IO-Devices im Hochlauf an, bearbeitet sein Steuerungsprogramm und überträgt die auszugebenden Prozess-Daten (Ausgänge) an die jeweiligen IO-Devices. Um diesen Datenaustausch durchführen zu können, benötigt er die Daten der Anlagenprojektion, die alle für die Kommunikation benötigten Daten enthält. Bei einer Anlagenprojektion werden folgende Daten festgelegt:

- Ausbaugrad eines IO-Devices,
- Parametrierungen für ein IO-Device,
- Übertragungshäufigkeit,
- der Ausbaugrad der Automatisierungsanlage,
- Informationen zu Alarmen und Diagnosen.

In einem PROFINET-System können mehrere IO-Controller eingesetzt werden. Sollen diese IO-Controller auf dieselben Daten in den IO-Devices zugreifen können, so ist das bereits beim Projektieren anzugeben.

Nachdem einem IO-Controller ein Name innerhalb der Automatisierungs-Anlage zugewiesen wurde, läuft die *'Start up Procedure'* im IO-Controller folgendermaßen ab:

- Starten des MAC-Interfaces,
- Starten LLDP zur Ermittlung der Nachbarschaftsinformationen,
- Starten des DCP-Senders und DHCP-Client (wenn vorhanden),
- Warten auf eine IP-Adresse (oder benutzen der intern vorhandenen IP-Adresse),
- Überprüfung der eigenen IP-Adresse und des Namens durch DCP und ARP zur Feststellung, ob bereits andere Teilnehmer dieselben Einstellungen besitzen.

Anschließend überprüft ein IO-Controller zuerst die Verfügbarkeit der projektierten IO-Devices und gibt ihnen den erforderlichen Namen (siehe auch Kapitel *'Die Adresszuordnung zu den IO-Devices'*). Dann baut er die benötigten AR's und CR's zu den IO-Devices auf und schreibt gegebenenfalls die in der GSD-Datei angegebenen Parameter in die IO-Devices.