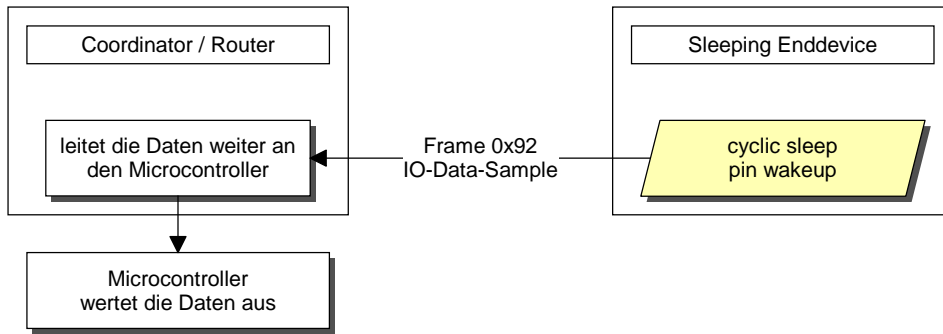


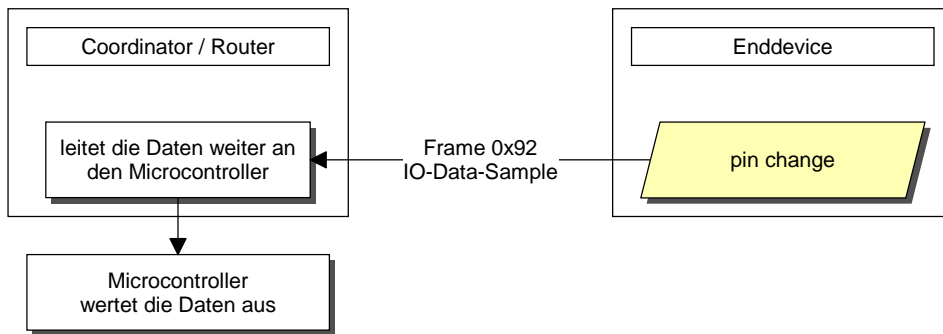
**Variante 1**

Sleeping Enddevice ohne Microcontroller.  
Ein cyclic sleep sendet IO-Sample an den Coordinator / Router.



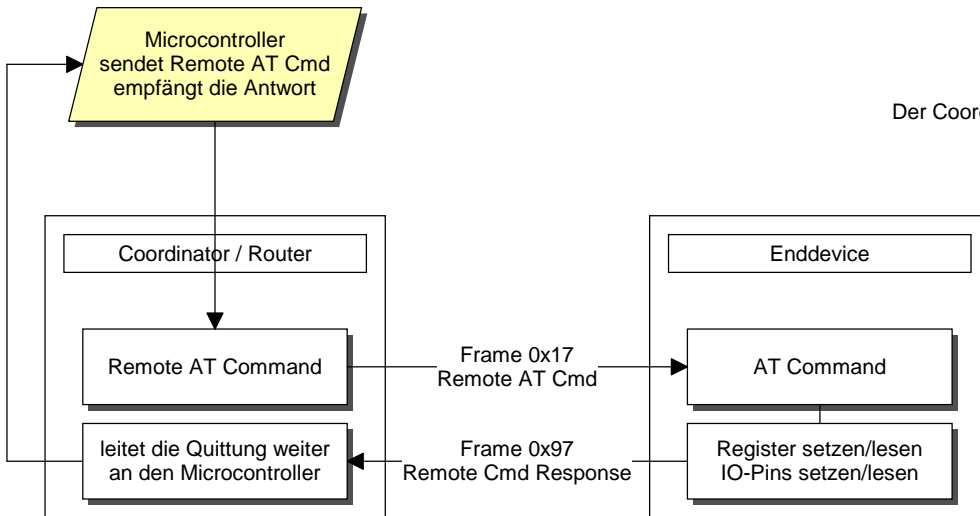
**Variante 2**

Enddevice ohne Microcontroller.  
Ein pinchange sendet IO-Sample an den Coordinator / Router.



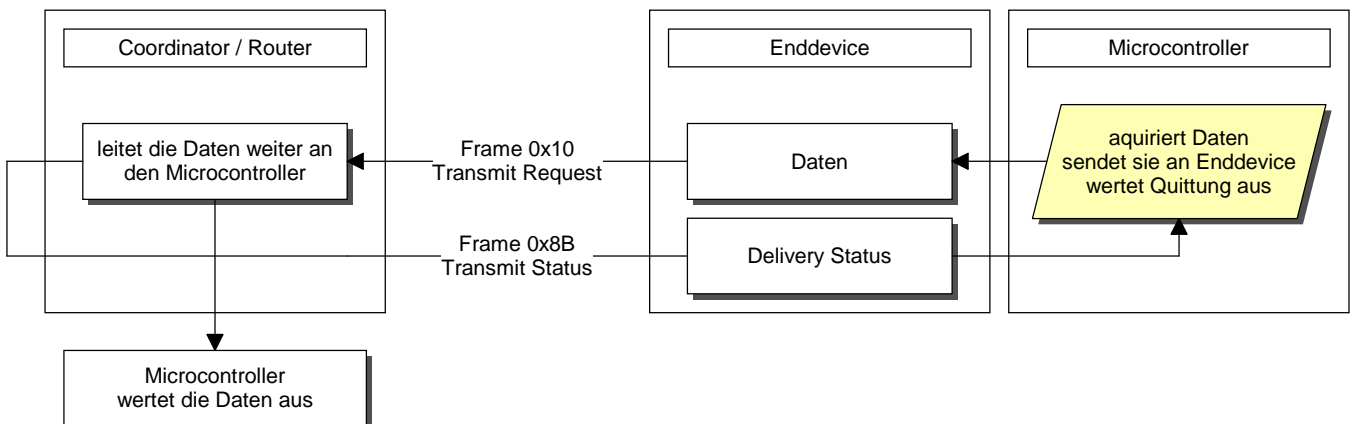
**Variante 3**

Enddevice ohne Microcontroller.  
Der Coordinator / Router sendet AT Commands und wertet die Antwort aus.



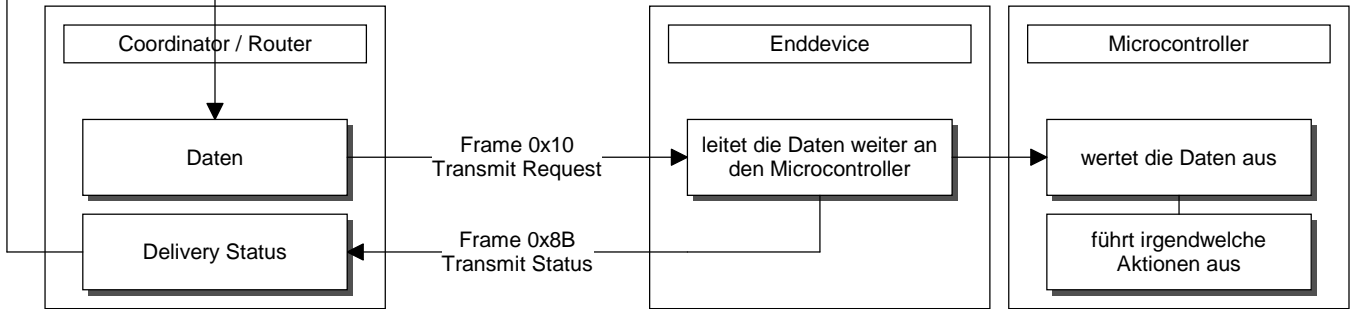
**Variante 4**

Enddevice mit zusätzlichem Microcontroller.  
Das Enddevice liefert eigenständig Daten an Coordinator / Router.



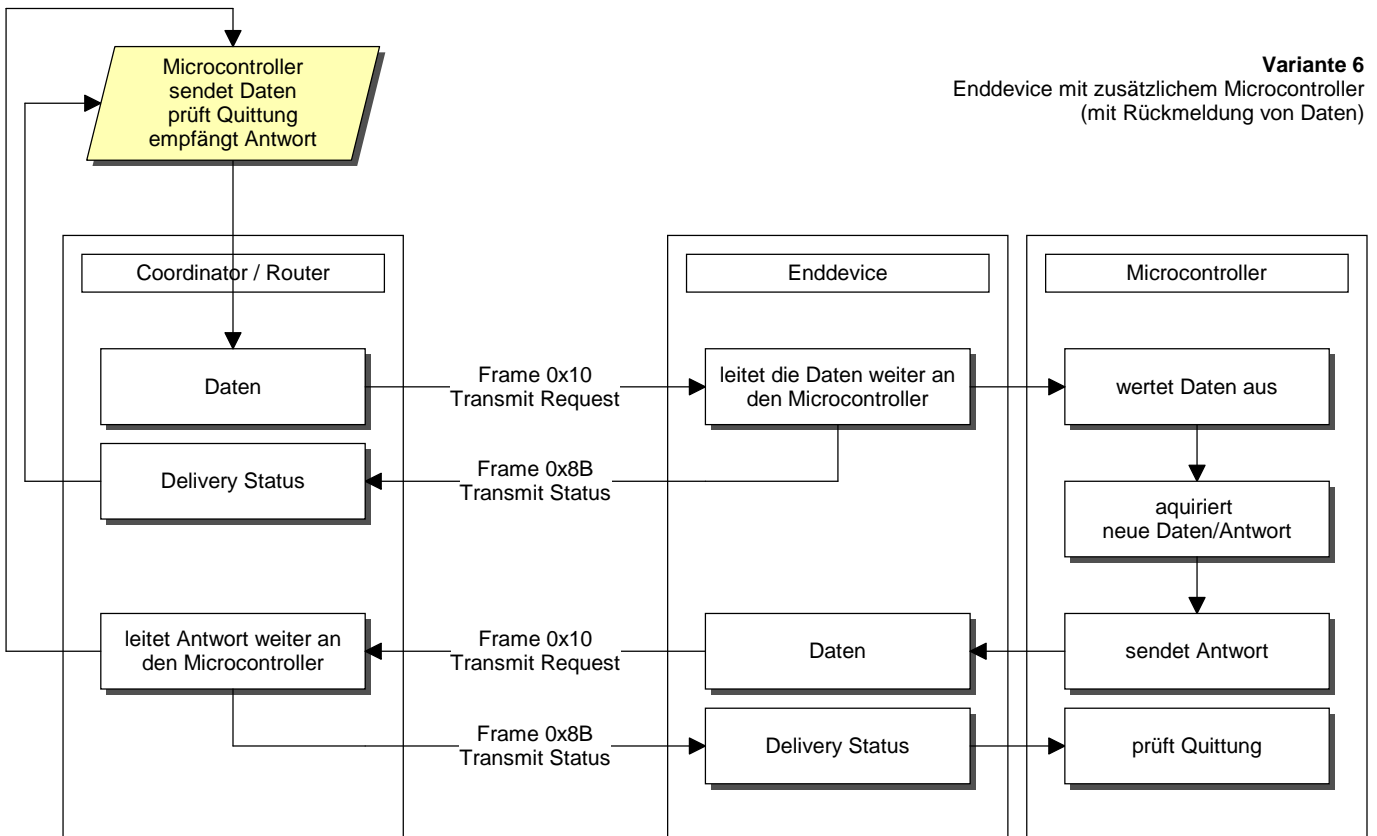
Microcontroller  
sendet Daten  
prüft Quittung

**Variante 5**  
Enddevice mit zusätzlichem Microcontroller  
(ohne Rückmeldung von Daten)



Microcontroller  
sendet Daten  
prüft Quittung  
empfängt Antwort

**Variante 6**  
Enddevice mit zusätzlichem Microcontroller  
(mit Rückmeldung von Daten)



## **Varianten der Kommunikation zwischen Router/Coordinator und Enddevice**

Der Coordinator/Router gibt in allen Varianten eingehende Daten via serieller Schnittstelle an einen Microcontroller oder an einen PC weiter.

### **1. Enddevice im Sleep-Modus**

Das Enddevice arbeitet im Sleep-Modus ohne zusätzlichen Microcontroller, ein Batteriebetrieb ist möglich.

Bei jedem Aufwachen werden IO-Daten versandt und vom Coordinator/Router via Frame 0x92 empfangen. Alternativ kann auch ein Pin wakeup genutzt werden, um den Sleep-Modus zu beenden (der Wakeup Pin wird auf low gezogen).

Es kann der Pinstatus (high/low) von bis zu 12 Portpins und/oder bis zu 4 10Bit-ADC Werte (Messbereich von 0 - 1200mV) übertragen werden.

### **2. Enddevice überträgt IO-Daten bei einem PinChange**

Das Enddevice arbeitet nicht im Sleep-Modus, es wird kein zusätzlicher Microcontroller benötigt.

Bei einem PinChange an den überwachten Pins werden IO-Daten versandt und vom Coordinator/Router via Frame 0x92 empfangen.

Es kann der Pinstatus (high/low) von bis zu 12 Portpins und/oder bis zu 4 10Bit-ADC Werte (Messbereich von 0 - 1200mV) übertragen werden.

### **3. Router/Coordinator setzt IO-Ports/Register am Enddevice oder fordert IO-Daten vom Enddevice an**

Das Enddevice arbeitet nicht im Sleep-Modus, es wird kein zusätzlicher Microcontroller benötigt.

Der Router/Coordinator schickt ein Remote AT-Command Frame 0x17 an den Empfänger um dort Portpins oder Register zu setzen oder den IO-Status oder Register auszulesen.

Die Daten werden vom Coordinator/Router via Frame 0x97 empfangen.

### **4. Enddevice mit zusätzlichem Microcontroller sendet Daten**

Das Enddevice wird durch einen Microcontroller unterstützt.

Dieser Microcontroller auf dem Enddevice initiiert die Datenübertragung, er versendet Daten via Frame 0x10 und erhält vom Coordinator/Router den Übertragungsstatus via Frame 0x8B zurück.

### **5. Enddevice mit zusätzlichem Microcontroller (ohne Rückmeldung von Daten)**

Das Enddevice wird durch einen Microcontroller unterstützt.

Der Coordinator initiiert die Übertragung. Er versendet Daten via Frame 0x10 und erhält den Übertragungsstatus via Frame 0x8B vom Enddevice zurückgeliefert.

Der Microcontroller am Enddevice empfängt die Daten via serieller Schnittstelle, decodiert sie und führt entsprechende Aktionen (Schaltaktionen etc.) aus.

Eine Rückmeldung an den Coordinator/Router über den Erfolg dieser Aktion erfolgt nicht.

### **6. Enddevice mit zusätzlichem Microcontroller (mit Rückmeldung von Daten)**

Das Enddevice wird durch einen Microcontroller unterstützt.

Der Coordinator initiiert die Übertragung. Er versendet Daten via Frame 0x10 und erhält den Übertragungsstatus via Frame 0x8B vom Enddevice zurückgeliefert.

Der Microcontroller am Enddevice empfängt die Daten via serieller Schnittstelle, decodiert sie und führt entsprechende Aktionen (Messungen) aus.

Eine Rückmeldung der akquirierten Daten an den Coordinator/Router erfolgt via Frame 0x10.

Der Coordinator/Router sendet eine Quittung via Frame 0x8B an das Enddevice.