

Public Support - Support Request #7654

Problem manipulating and trasmitting data

2019-07-03 14:45 - hidden

Status:	Closed	
Priority:	Normal	
Category:		
Customer:	PORSCHE	Product Issue Numbers:
Department:		Affected Products: ADTF 2.14.1
Requester's Priority:	Normal	Platform:
Support Level:	2nd Level	Topic: DeviceTB::Flexray
Resolution:	Solved Issue	FAQ Links:

Description

Supportanfrage

Ich arbeite gerade daran ihren Flexray Signal Decoder in ADTF 2.14.1 für unsere Bedürfnisse anzupassen.

Nun habe ich folgendes Problem:

Ich habe eine Vielzahl von Signalen und möchte diese zum Teil 1 zu 1 übertragen und zum anderen durch konstante Werte überschreiben.

Ich habe hierfür eine while-Schleife geschrieben, aber leider wird die Simulation dadurch extrem langsam.

Durch den Befehl "m_pCoder->TransmitData(pData, &nDataSize);" werden nur modifizierte Werte überschrieben. Werte die aber konstant bleiben sollen, werden leider nicht übertragen.

Als Lösung habe ich mir folgende Möglichkeiten überlegt:

- 1.) Wir benötigen einen Befehl, der alle Werte künstlich auf modified setzt. Also eine Art: //m_pCoder->SetModifiedFlag(); // sets all modified flags
- 2.) Wir benötigen einen Befehl, der alle Werte ausgibt, egal ob überarbeitet oder nicht.

Wir können gerne am Donnerstagmittag ab 13:30 Uhr Skypen, dabei kann ich die die offenen Punkte aus der Mail #7646 erklären.

Nachfrage

ganz verstehe ich den Use Case noch nicht. Ein paar Dinge sind von den Begrifflichkeiten her unklar:

Was bedeutet bei euch

"Werte werden nicht übertragen": Wohin übertragen? Übertragen ist bei uns immer mit IPC/Netzwerk konotiert, aber darum gehts hier glaub ich nicht.

"Wir benötigen einen Befehl, der alle Werte ausgibt, egal ob überarbeitet oder nicht." Was heißt hier ausgeben? In ein Log schreiben, oder irgendwo ablegen? Ich vermute hier im ADTF Sample zu aktualisieren?

Wohin sollen die Flexray-Frames denn geschrieben/übertragen werden? Im Moment kenne ich folgende Kette:

ADTF DAT File -> Player -> Flexray Signal Filter (von Euch, mit Flexray Media Coder) -> ???

Soweit ich das erkenne wollt ihr an eurem Flexray Signal Filter wieder ADTF Samples mit Flexray Frames rausschreiben. Sollen die wieder auf einen "echten" Flexray geschrieben werden?

Mein naiver Ansatz wäre man legt sich in einer Hashmap Cycle und Slot IDs der zu modifizierenden Frames ab und leitet alle anderen Samples/Frames 1:1 am Ausgangspin wieder weiter.

Bei den zu modifizierenden Frames kopiert man die Samples in neue Samples und iteriert dann mit dem Coder über alle zu veränderten Signale und setzt die wie gewünscht um.

Bitte noch eine Erläuterung warum ihr bei den nicht modifizierten Signalen auch ein gesetztes Modified Bit haben wollt? Braucht ihr das im Fall auch für Frames in denen gar keine Signale verändert wurden?

Lösung

Ich habe einen eigenen Filter geschrieben, welcher im Low-Level die ganzen PDU's überschreibt und weitergibt.

Durch das Low-Level ist die Performance deutlich verbessert als beim High-Level.

History

#1 - 2019-07-03 15:43 - hidden

- Project changed from Public Support to 16
- Status changed from New to In Progress
- Topic set to DeviceTB::Flexray
- Customer set to PORSCHE
- Affected Products ADF 2.14.1 added

Hi Martin,

hier etwas mehr zum Use Case... kannst du das übernehmen ?

#2 - 2019-07-04 08:46 - hidden

Hallo,

ganz verstehe ich den Use Case noch nicht. Ein paar Dinge sind von den Begrifflichkeiten her unklar:
Was bedeutet bei euch

- "Werte werden nicht übertragen": Wohin übertragen? Übertragen ist bei uns immer mit IPC/Netzwerk konotiert, aber darum gehts hier glaub ich nicht.
- "Wir benötigen einen Befehl, der alle Werte ausgibt, egal ob überarbeitet oder nicht." Was heißt hier ausgeben? In ein Log schreiben, oder irgendwo ablegen? Ich vermute hier im ADF Sample zu aktualisieren?

Wohin sollen die Flexray-Frames denn geschrieben/übertragen werden? Im Moment kenne ich folgende Kette:

ADF DAT File -> Player -> Flexray Signal Filter (von Euch, mit Flexray Media Coder) -> ???

Soweit ich das erkenne wollt ihr an eurem Flexray Signal Filter wieder ADF Samples mit Flexray Frames rausschreiben. Sollen die wieder auf einen "echten" Flexray geschrieben werden?

Mein naiver Ansatz wäre man legt sich in einer Hashmap Cycle und Slot IDs der zu modifizierenden Frames ab und leitet alle anderen Samples/Frames 1:1 am Ausgangspin wieder weiter.

Bei den zu modifizierenden Frames kopiert man die Samples in neue Samples und iteriert dann mit dem Coder über alle zu veränderten Signale und setzt die wie gewünscht um.

Bitte noch eine Erläuterung warum ihr bei den nicht modifizierten Signalen auch ein gesetztes Modified Bit haben wollt? Braucht ihr das im Fall auch für Frames in denen gar keine Signale verändert wurden?

Grüße,

Martin

PS: Ein Online Meeting können wir gerne nächste Woche machen, aber es ist wichtig, dass die grundlegenden Dinge hier in Tickets schriftlich festgehalten werden, weil es sonst für jemand anderen (so wie jetzt für mich) sehr schwierig ist zu übernehmen.

#4 - 2019-07-17 08:24 - hidden

- Status changed from In Progress to Customer Feedback Required

[hidden](#): Ist das hier noch aktuell ? Wie ist hierzu die Erwartungshaltung und das weitere Vorgehen angedacht ?

#5 - 2019-07-17 10:20 - hidden

- Resolution set to Solved Issue

Hallo Alexander,

danke fürs Feedback.

Kannst du uns noch deine Lösung kundtun ?

Ansonsten ist das Ticket hier so halbgar...

Danke !

#6 - 2019-07-17 10:30 - hidden

Kann geschlossen werden, habe das Problem anderweitig gelöst.

Mit freundlichen Grüßen

Alexander Clauss

#7 - 2019-07-17 10:45 - hidden

Ich habe einen eigenen Filter geschrieben, welcher im Low-Level die ganzen PDU's überschreibt und weitergibt. Durch das Low-Level ist die Performance deutlich verbessert als beim High-Level.

Mit freundlichen Grüßen

Alexander Clauss

#8 - 2019-07-18 13:58 - hidden

- *Private changed from Yes to No*
- *Project changed from 16 to Public Support*
- *Subject changed from ADTF 2.14.1: Problem mit TransmitData to Problem manipulating and trasmitting data*
- *Description updated*
- *Status changed from Customer Feedback Required to To Be Closed*

#9 - 2020-07-07 12:44 - hidden

- *Status changed from To Be Closed to Closed*