

Public Support - Support Request #5912

Filter trace view vs. Streamtime

2019-01-28 11:06 - hidden

Status:	Closed	
Priority:	Normal	
Category:		
Customer:	BOSCH	Product Issue Numbers:
Department:	CC-DA/ESU	Affected Products: ADTF 2.13.3
Requester's Priority:	Normal	Platform: Windows 7 64bit
Support Level:	2nd Level	Topic: ADTF::Clock
Resolution:	Solved Issue	FAQ Links:

Description

Supportanfrage

Eine Frage zum Filter Trace View:

Berechnet der die Datenraten anhand der StreamTime oder gegen eine freilaufende System-Zeit?

Wir haben Funktionen aus einer Steuergeräte-SW in ADTF eingebunden, gehen damit ins Fahrzeug und machen auch ein Replay der Daten.

Bisher wurde ein cKernelThread genutzt, um die Funktionen im 10 ms Takt aufzurufen.

Das funktioniert im Fahrzeug sehr gut, im Replay jedoch nicht zu meiner Zufriedenheit. Zum Beispiel stimmen die Datenraten dann nicht mehr, wenn ich die Abspielgeschwindigkeit auf Faktor 4 setze. Zudem befürchte ich, dass ich mit einer Thread-Steuerung jedes mal andere Ergebnisse erhalte, wenn ich die gleiche Aufnahme mehrmals abspiele.

Daher habe ich eine zweite Option fürs Replay implementiert, die einen cKernelTimer zum Aufruf der Funktionen nutzt. Damit funktioniert das Abspielen mit Faktor 4 auch. Und die Reproduzierbarkeit sollte somit auch gegeben sein?

Die angezeigte Datenrate bzw. die SamplesPerSecond im Replay geht jedoch unter die Erwartung zurück. Meiner Meinung nach ist das aber okay, wenn eben mal nicht genügend Rechenleistung zur Verfügung steht...

Lösung

cKernelTimer ist für genau diese Anwendungsfälle gedacht (Um eben auch im Playback zu funktionieren und genauer zu sein als ein Sleep). Und ja das Filter Trace View nutzt immer "Echtzeit", also einfach die System Zeit zur Berechnung der Datenrate (und nicht die Stream Time).

Filter sollten und müssen eigentlich nie selbst eine Unterscheidung zwischen Live- und Playbackbetrieb machen. Der cKernelCyclicThread ist eher dafür gedacht, auf ein Event zu warten (Condition Variable) um dann Daten asynchron zu verarbeiten, als dass man einen Timer mittels Sleep simuliert.

History

#1 - 2019-01-29 13:50 - hidden

- Project changed from Public Support to 5
- Status changed from New to In Progress
- Topic set to ADTF::Clock

#2 - 2019-01-30 14:10 - hidden

Ja, alles ist so, wie von Dir beschrieben. cKernelTimer ist für genau diese Anwendungsfälle gedacht (Um eben auch im Playback zu funktionieren und genauer zu sein als ein Sleep). Und ja das Filter Trace View nutzt immer "Echtzeit", also einfach die System Zeit zur Berechnung der Datenrate (und nicht die Stream Time).

Grüße,

Martin

#3 - 2019-01-30 14:30 - hidden

Hallo Martin,

sollte für die Aufnahme im Fahrzeug bzw. die LiveDemos der Funktion im Fahrzeug der cKernelCyclicThread beibehalten werden oder ist da auch der cKernelTimer zu empfehlen?

Mit freundlichen Grüßen / Best regards

Stephan Stuehmer

#5 - 2019-01-30 14:45 - hidden

Hi Stephan,

Ja, auch da ist der cKernelTimer sicher besser. Filter sollten und müssen eigentlich nie selbst eine Unterscheidung zwischen Live- und Playbackbetrieb machen.

Der cKernelCyclicThread ist eher dafür gedacht, auf ein Event zu warten (Condition Variable) um dann Daten asynchron zu verarbeiten, als dass man einen Timer mittels Sleep simuliert.

Grüße,

Martin

#6 - 2019-01-31 15:53 - hidden

- Status changed from In Progress to Customer Feedback Required

#7 - 2019-02-04 09:37 - hidden

- Project changed from 5 to Public Support

- Description updated

- Status changed from Customer Feedback Required to To Be Closed

- Private changed from Yes to No

- Resolution set to Solved Issue

#8 - 2019-02-04 09:37 - hidden

- Status changed from To Be Closed to Closed