Public Support - Support Request #6834

cKernelcyclicThread and external clock

2019-04-10 12:51 - hidden

Status:	Closed		
Priority:	Normal		
Category:			
Customer:	AUDI	**Product Issue Numbers:**	
Department:	EFS	**Affected Products:**	ADTF 2.14.0
Requester's Priority:	Normal	**Platform:**	Ubuntu 16.04 64bit
Support Level:	2nd Level	**Topic:**	ADTF::Clock
Resolution:	Solved Issue	**FAQ Links:**	

Description

Supportanfrage

Ich habe eine Frage zur Verhaltensweise von CyclicThreads in Kombination mit einer externen ClockSorce im ADTF.

Mein Setup ist wie folgt:

- VTD
- ADTF Instanz 1 (Timemaster) mit Demo_VTD_Trigger_RDB (ADTF VTD Toolbox 4.1.0) zur Zeitsteuerung/Triggerung vom VTD
- ADTF Instanz 2 mit VTD_ADTF_Time_Sync (ADTF VTD Toolbox 4.1.0) zur Synchronisation der ADTF-Streamtime mit VTD

In ADTF Instanz 2 habe weitere Filter und Services zur Verarbeitung von Daten aus VTD.

Ein Filter erbt dabei von der adtf:CKernelSyclicThread und beinhaltet eine Funktionalität, welche zyklisch Outputdaten generieren soll.
Das Intervall wird dabei über eine Property übergeben und mittels SetCycletime gesetzt.

Wenn ich nun als Beispiel das Intervall so einstelle, dass ich alle 1000ms Daten generieren möchte, funktioniert alles so lange ich den TimeMaster so einstelle, dass VTD und damit auch ADT Instanz 2 in Echtzeit laufen (genauer - alle 20ms einen 20ms Zeitschritt mache). Die CyclicFunc() wird erwartungsgemäß einmal pro Sekunde (nach ADTF-Streamtime) aufgerufen.

Wenn ich nun aber alles langsamer laufen lasse, bekomme ich eine abweichende Frequenz. Lasse ich z.B die Simulation mit halber Geschwindigkeit laufen(alle 40ms einen 20ms Zeitschritt mache), bekomme ich den zyklischen Aufruf (CyclicFunc()) alle 760ms nach StreamTime (Realzeit ca alle 1500ms)
Meine Erwartung an dieser Stelle war, dass ich weiterhin alle 1000ms in Simulationszeit/Streamtime ADTF-Instanz 2 einen Funktionsaufruf bekomme.

Die tatsächliche Rate ist anders , aber konstant - nach welcher Clock richtet sich dieser Aufruf tatsächlich? Wie ist dsa Soll-Verhalten in diesem Fall?
Welche ADTF-Mechanismen kann ich nutzen um das gewünschte Verhalten auf bei langsamerer Uhr zu erhalten?

Lösung

cKernelCyclicThread schläft einfach mit cSystem::Sleep und kann nicht wirklich mit langsameren oder schnelleren Uhren umgehen.

Es ist eigentlich immer besser einen cKernelTimer zu nutzen, da der korrekt auf die Reference Clock hört (sowohl im Playback als im Live Betrieb).

History

#1 - 2019-04-10 13:20 - hidden

- Project changed from Public Support to 11

- Status changed from New to In Progress

- Topic set to ADTF::Clock

#2 - 2019-04-10 14:17 - hidden

cKernelCyclicThread schläft einfach mit cSystem::Sleep und kann nicht wirklich mit langsameren oder schnelleren Uhren umgehen.

Es ist eigentlich immer besser einen cKernelTimer zu nutzen, da der korrekt auf die Reference Clock hört (sowohl im Playback als im Live Betrieb).

Grüße,

Martin

#3 - 2019-04-10 17:29 - hidden

Hallo,
Danke für die schnelle Rückmeldung.
Unter Verwendung von cKernelTimer konnte ich das gewünschte verhalten herstellen0

Grüße,
Alex

#4 - 2019-04-11 09:01 - hidden

- Project changed from 11 to Public Support

- Subject changed from cKernelcyclicThread und externe Clock to cKernelcyclicThread and external clock

- Description updated

- Status changed from In Progress to To Be Closed

- Private changed from Yes to No

- Resolution set to Solved Issue

#5 - 2020-07-07 12:44 - hidden

- Status changed from To Be Closed to Closed