

テストレポートのご確認方法

テストレポートが取得できたところで、
正常なデータが取得できているかを確認してみましょう。

誤ったレポートデータとは？

テストレポートは、ターゲット機器が出力したテストポイントの通過情報に対して、DT10プロジェクトで管理するソースファイル名・関数名・ステップ種別の情報を紐付けして表示します。

しかし、以下のような問題があった場合、
取得したデータと識別を管理する情報の不整合により、
誤ったレポートデータが取得される可能性があります。

- ・ターゲットへ組み込むドライバの問題
- ・ターゲットとの物理的な接続・ノイズの問題
- ・DT10プロジェクト作成時の誤まった手順 など。

No.	コア	ソース	関数	ステップ	説明
22 (22)		DisplayRenderer.c	DRenderer_BeginScene	01 FuncOut	
23 (23)		DisplayRenderer.c	DRenderer_DrawDeviceFormatImage	00 FuncIn	
24 (24)		DisplayRenderer.c	DRenderer_DrawDeviceFormatImage	03 if	
25 (25)		DisplayRenderer.c	DRenderer_DrawDeviceFormatImage	05 FuncOut	
26 (26)		DisplayRenderer.c	DRenderer_DrawDeviceFormatImage	00 FuncIn	
27 (27)		DisplayRenderer.c	DRenderer_DrawDeviceFormatImage	03 if	
28 (28)		DisplayRenderer.c	DRenderer_DrawDeviceFormatImage	05 FuncOut	
29 (29)		DisplayRenderer.c	DRenderer_DrawDeviceFormatImage	00 FuncIn	
30 (30)		DisplayRenderer.c	DRenderer_DrawDeviceFormatImage	03 if & FuncOut	
31 (31)		DisplayRenderer.c	DRenderer_DrawDeviceFormatImage		
32 (32)		DisplayRenderer.c	DRenderer_DrawDeviceFormatImage		
33 (33)		DisplayRenderer.c	DRenderer_EndScene		
34 (34)		DisplayRenderer.c	DRenderer_EndScene		
35 (35)		tasks.c	***** Event Trigger		
36 (36)		tasks.c	***** Event Trigger		
37 (37)		Task_RGB-LED.c	Task_RGBLED		
38 (38)		Task_Hardware.c	getColorVolume		

誤ったレポートデータの具体例

- ・ソース名のカラムに「***** Idle *****」と表示されるデータがある。
- ・関数名のカラムに「End of Dump」と表示されるデータがある。
- ・EVENTトリガー用テストポイントを手動挿入していないのに、「***** Event Trigger *****」と表示されるデータがある。
- ・変数値出力用テストポイントを手動挿入していないのに、「***** Dump Memory *****」と表示されるデータがある。
- ・EVENTトリガー用、変数値出力用テストポイントが挿入された処理を通過していないのに、上記の誤ったデータが表示される。
- ・ソース名、関数名、ステップ名のカラムに、「0x01」などの16進数の数値データが表示される。

No.	コア	ソース	関数	ステップ	説明
1 (1)		***** Idle *****			0x0000
2 (2)		***** Idle *****			0x0000
3 (3)		***** Idle *****			0x0000
4 (4)		***** Idle *****			0x0000
5 (5)		***** Idle *****			0x0000
6 (6)		ata_drv.c	AtaDrvInterval	00 FuncIn	
7 (7)		ata_drv.c	AtaDrvInterval	03 FuncOut	
8 (8)		ata_drv.c	AtaDrvInterval	00 FuncIn	
9 (9)		ata_drv.c	AtaDrvInterval	03 FuncOut	

No.	コア	ソース	関数	ステップ	説明
1 (1)		DisplayRenderer.c			0x006b
2 (2)		DisplayRenderer.c			0x002b
3 (3)		DisplayRenderer.c	DRenderer_EndScene	01 FuncOut	
4 (4)		DisplayRenderer.c	DRenderer_DrawDeviceFormatImage	03 if	
5 (5)		DisplayRenderer.c	DRenderer_DrawDeviceFormatImage	04 if & FuncOut	
6 (6)		DisplayRenderer.c	DRenderer_DrawDeviceFormatImage	04 if & FuncOut	
7 (7)		DisplayRenderer.c	DRenderer_DrawDeviceFormatImage	04 if & FuncOut	
8 (8)		DisplayRenderer.c	DRenderer_DrawDeviceFormatImage	04 if & FuncOut	
9 (9)		DisplayRenderer.c	DRenderer_DrawDeviceFormatImage	04 if & FuncOut	

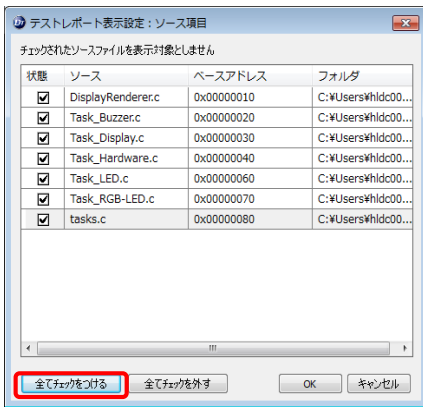
16進数の数値が表示される。

テストレポートのご確認方法

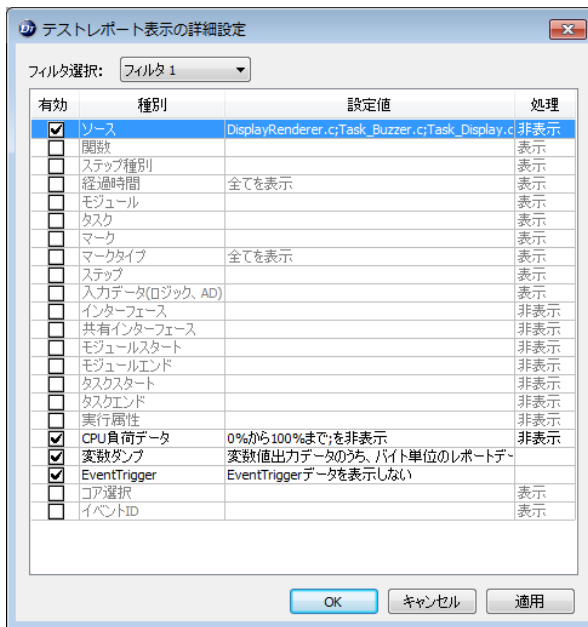
フィルタ機能を使った確認方法

DT10のフィルタ機能を使用することで、テストレポート内の誤ったデータを発見することができます。

- ①ソース名をフィルタの対象にする。
- ②ソース名のフィルタ処理を「非表示」で「有効」にチェックする。



・チェックしたソース名のデータがフィルタの対象となります。



- ・選択したソース名を対象に非表示のフィルタとして、有効化します。
- ・CPU負荷測定、変数値出力、イベント出力のデータを無視する場合は、非表示にします。

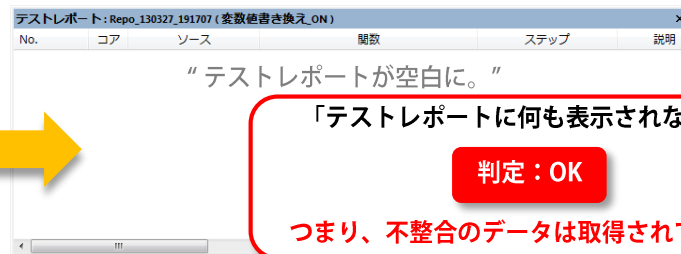
- ③フィルタ後のテストレポートで判断。

▼フィルタ設定前

No.	コア	ソース	関数	ステップ	説明
12 (15)		Task_RGB-LED.c	Task_RGBLED	06 switch	
13 (16)		Task_Hardware.c	setRGBLEDCOLOR	00 FuncIn	
14 (17)		Task_Hardware.c	setRGBLEDCOLOR	01 FuncOut	
15 (19)		DisplayRenderer.c	DRenderer_Present	04 FuncIn	
16 (20)		Task_Hardware.c	procHardware_LED_AD	25 FuncOut	
17 (22)		DisplayRenderer.c	DRenderer_BeginScene	01 FuncOut	
18 (23)		DisplayRenderer.c	DRenderer_DrawDeviceFormatImage	00 FuncIn	
19 (24)		DisplayRenderer.c	DRenderer_DrawDeviceFormatImage	03 if	
20 (25)		DisplayRenderer.c	DRenderer_DrawDeviceFormatImage	05 FuncOut	



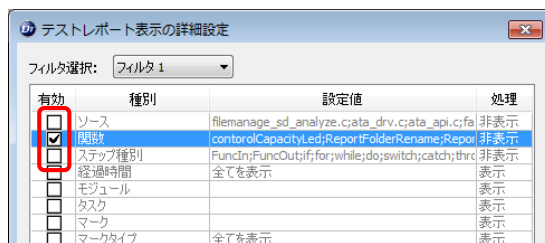
▼非表示フィルタを有効化



テストレポートのご確認方法

フィルタ機能を使った確認方法 つづき

ソース名で非表示のフィルタを設定した時と同様に、関数名・ステップ種別についても、非表示フィルタを有効にして、誤ったデータがテストレポート上に表示されないことを確認していきます。



ソース・関数・ステップ種別のいずれかを有効にします。
ソース → 関数 → ステップ種別の順にご確認ください。

トラブルシューティング

もしも、非表示フィルタ後に、誤ったデータが表示されていたら・・・

No.	コア	ソース	関数	ステップ	説明
1 (1)		***** Idle *****		0x0000	
2 (2)		***** Idle *****		0x0000	
3 (3)		***** Idle *****		0x0000	
4 (4)		***** Idle *****		0x0000	
5 (5)		***** Idle *****		0x0000	

業種・分野・システムを問わず導入実績多数。
まずはDT10技術サポートまでお問合せください。

ハートランド・データ株式会社
DT10技術サポート

user@hldc.co.jp TEL 0284-22-8791