1.質問

変数をループ監視するようなタスクは、どのようになるのでしょうか?

2.回答

変数をループ監視するようなタスクを設計するには、下記に示す設定を行います。

- 1. STM のイベント種別を、フラグ型イベントに設定します。
- 2. メッセージ受信のシステムコールを行わないことを設定します。
- 2.1 フラグ型イベント

本章では、フラグ型イベントについて説明します。

変数をループ監視するタスクの場合は、メッセージを受信するまで待つことはありません。 よって、STM のイベント種別を、メッセージ型でなくフラグ型にします。

フラグ型イベントとは、変数を参照して発生させるイベントであり、下図のように表示します。



fig フラグ型イベント例





イベントセルを選択した状態で、[イベント] - [イベント属性] - [フラグ・条件型]メニュー を選択することで、フラグ型イベントを設定します。

ሺ ZIPC Developer Studio – [[Binary]ボーリング機構.stm]					
Ⅱ ファイル(E) 編集(E) 書式(O)	11210	遷移先自動調整(<u>A</u>)	ツール(II)	プロジェクト(<u>P</u>)	表示(⊻)
D 🚅 🖬 🗐 🗉 Desig MS ゴシック	イヘント属 開始アクラ 終了アクラ	性(<u>E)</u> れどうれ(I) れどうれ(<u>L</u>)	•	メッセージ型() ✓ フラゲ・条件型 SWITCH条(A) ≝(J) ≇(S)
· □··□ ZIPCトギキュメント-[オ □·□ ZIPC環境ファイ	分割(<u>D</u>) ゲルーフ៕	ŁG	Ctrl+D Ctrl+G	割り込み(1) インメイル(<u>N</u>) 関数(E)	
- 📄 システム情報 🎽 🎽	たの登録・編集(<u>M</u>)…		関数グル-ブ(<u>G</u>) 自動遷移(<u>T</u>)		
🔛 🔛 🖓 🛄 🖂 🛁 👘	わ.背呂色の途1つぶし(R)				

2.2 メッセージ受信

本章では、メッセージ受信のシステムコールを行わない方法について説明します。

通常 ZIPC では、タスクのメイン関数にてメッセージ受信を行います。

しかし変数をループ監視するタスクでは、メッセージ受信を行いません。

- よって、下記手順の操作を行い、メッセージ受信のシステムコールを行わないことを設定します。 手順
- 1. シミュレーションフェーズに移行します。
- 2. [プロジェクト] [プロジェクト設定] [ジェネレータ設定] [C コード生成設定] メニューを選択して、[Cコード生成設定]ダイアログを起動します。

😢 ZIPC Developer Studio				
ファイル(E) 書式(Q) ツール(T)	プロジェクト(<u>P</u>) 表示(<u>V</u>)	∿ルフ°(<u>H</u>)	_	
	プロジェクトフェースヾ(<u>P</u>) プロンジェクト認定(S)	•	トドキュアルチェック語会定(C)	
				> 本) トナート(三九一〇/14)
🖭 🚞 ZIPC環境ファイ	美行琼境生成(<u>L</u>)	F /	ジェネレー%!覚定(G) ▶	ジェネレータモート設定(<u>M</u>)…
	トッキュ メント目白 (ボギテック(口)		33-15-35)主行コード設定(S)	



- 3. [Cコード生成設定]ダイアログにて、[タスク・ハンドラ設定]タブを選択します。
- 4. [Cコード生成設定]ダイアログ [タスク・ハンドラ設定]タブにて、指定のタスク("ポ ーリング機構"タスク)を選択し、[イベント受信(REV)メインコール]項目のチェックを 外します。
- 5. [Cコード生成設定]ダイアログ、[OK]ボタンを選択します。

C1-ト*生成設定	X
生成先設定 タスク・ハンドラ設定)レートSTM設定 STM設定 全体設定	
タスク・ハント ⁵ ラー覧(B):	
タスク・ハントラ名称 「ロシステム 共通	
■ポーリング機構	
	<i>ν</i>

2.3 シミュレーション

以上の手順だけでも、ZIPC シミュレーション環境にて、変数ウインドウで変数の値を設定す ることにより、デバッグを行うことができます。

しかし、メッセージイベントと同様に STM 上からイベントを発行することはできません。 STM 上からイベントを発行させるには、下記の手順にてイベント発行定義を行います。

手順

- 1. シミュレーション実行環境を生成します。
- [デバッグ] [イベント発行 / 設定]メニューを選択して、[イベント発行/定義]ダイア ログを起動します。



- 3. [イベント発行/定義]ダイアログにて、[STM]タブを選択します。
- 4. [イベント発行/定義]ダイアログ [STM]タブにて、[自動再生成]項目のチェックを外し ます。

イベント発行/定義		×
OBJ STM ಕ್ರಿಸೆಲಿಂತ	ע	
資源名称(<u>0</u>): ポーリング機構 ▼	イベント名称:	変更(<u>C</u>)
STM名称(<u>S</u>): ポーロング継棋 stm マ	データ定義: //flg == 0x01	削除(D)
イベンドスド(E):		追加(<u>A</u>)
[1] event0 [2] event1 [3] event2 [4] event3) システムコール:	再生成(M)
[5] event4		発行(<u>R</u>)
□ 自動再生成(0)	エクスポートイベント名:	
インポート	Jue 0x01	更新
エクスポート	開じる(<u>L</u>)	

- 5. [イベント発行/定義]ダイアログ [STM]タブにて、[資源名称]項目を "ポーリング機構 " を選択します。
- 6. [イベント発行/定義]ダイアログ [STM]タブにて、[STM 名称 項目を 'ポーリング機構.stm " に選択します。
- 7. [イベント発行/定義]ダイアログ [STM]タブにて、[イベントリスト]項目を "[1]event0 " に選択します。
- 8. [イベント発行/定義]ダイアログ [STM]タブにて、[変更]ボタンを選択して、[イベント定義]ダイアログを起動します。

イベント発行/定義	×
OBJ STM オプション	
	変更(<u>C</u>)
STM名称(S)・ ポーリング機構stm ▼ //flg == 0x01	削除(<u>D</u>)
イヘ ⁵ ンドリスト(F): [1] event0	追加(<u>A</u>)
[3] event2 [4] event3 [5] event4	<u>再主成而</u> 発行(<u>B</u>)
自動再生成(U) エクスポートイベント名: インポート flg == 0x01	 エクスポート名 更新
エクスポート 閉じる(し)	



9. [イベント定義]ダイアログにて、[データ定義]項目を下記コードに設定します。 データ定義設定コード

f Ig = 0x01;

10. [イベント定義]ダイアログにて、[OK]ボタンを選択します。

イベント定義	
タイトル(①: event0	ОК
イベント名称(<u>E</u>):	キャンセル
データ定義(<u>A</u>): flg = 0x01;	
у́хテムコール(у):	
	位置 : 1

11. 下表に示すイベントに対して、データ定義を設定します。

イベントデータ定義

No	イベントリスト	データ定義
1	event0	flg = 0x01;
2	event1	flg = 0x02;
3	event2	flg = 0x03;
4	event3	flg = 0x04;
5	event4	$f \lg = 0x05$

12. [イベント発行/定義]ダイアログにて、[閉じる]ボタンを選択します。

イベント発行/定義		×
OBJ STM オプション	ע <u>ו</u>	1
資源名称(<u>0</u>): ポーリング機構 ▼	イベント名称:	変更(<u>C</u>)
STM名称(<u>S</u>): ポーリング機構stm ▼	データ定義: flg = 0x01;	<u> </u>
イヘントリスト(E): [1] event0		追加(<u>A</u>) 再生成(M)
[2] event1 [3] event2 [4] event3 [5] event4		<u>発行(B)</u>
□ 自動再生成(U)	エクスポートイベント名:	
インポート)	
エクスポート		

13. シミュレーション環境を構築します。

これで、メッセージイベントと同様に STM 上からイベントを発行することはできます。



