

1. 質問

無視セルや不可セルの処理を定義するには、どのようにするのでしょうか？

2. 回答

無視セルや不可セルの処理を定義するには、[無視、不可セル関数コール] 機能を使用します。

[無視、不可セル関数コール] 機能の設定方法は、次章より説明します。

3. 無視不可関数

3.1 プロジェクト作成

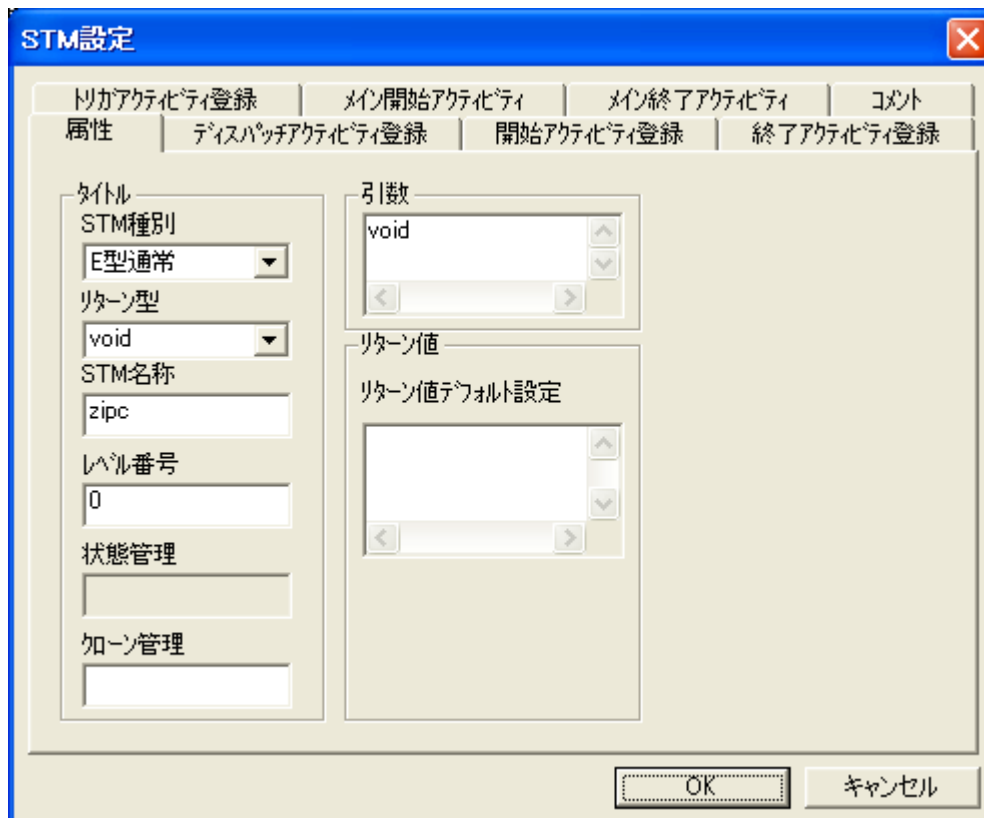
本章では、[無視、不可セル関数コール] 機能の設定手順を説明します。

■手順

1. ZIPC プロジェクトファイル（無視不可関数.zipf）を新規作成します。
2. " zipc" タスクを登録します。
3. " zipc" タスクに," zipc" STM 設計書(zipc.stm)を登録します。
4. " zipc" STM 設計書の属性を、下記のように設定します。

■" main" STM 設計書属性

No	項目	値
1	STM 種別	E 型通常
2	リターン型	void
3	STM 名称	zipc
4	レベル番号	0
5	引数	void



5. "zipc" STM 設計書を、下図のように編集します。

□0 zipc	S	s0	s1
E		0	1
e0	0	s1 処理A	×
e1	1	/	s0 処理B

6. "zipc" タスクに、"zipc" FNC 設計書(zipc.fnc)を登録します。

7. "zipc" FNC 設計書に、下記コードを追記します。

■追記するコード

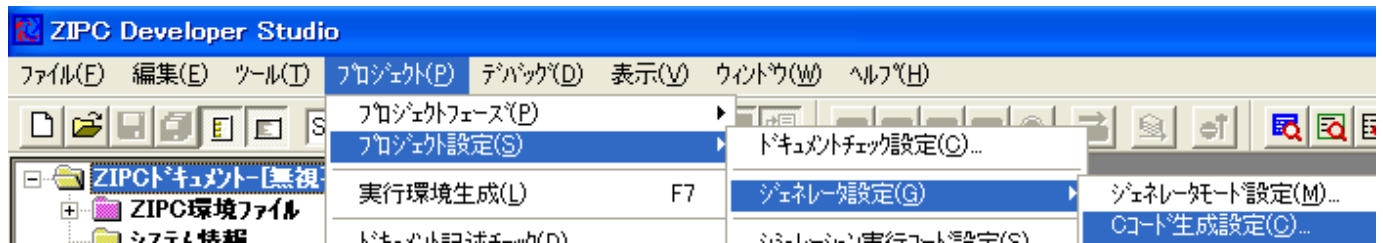
```

/*ZIPC FUNC HEADER*/
void ImpossibleFunc( void )
{
    不可処理
    printf( "不可処理\n" );
}

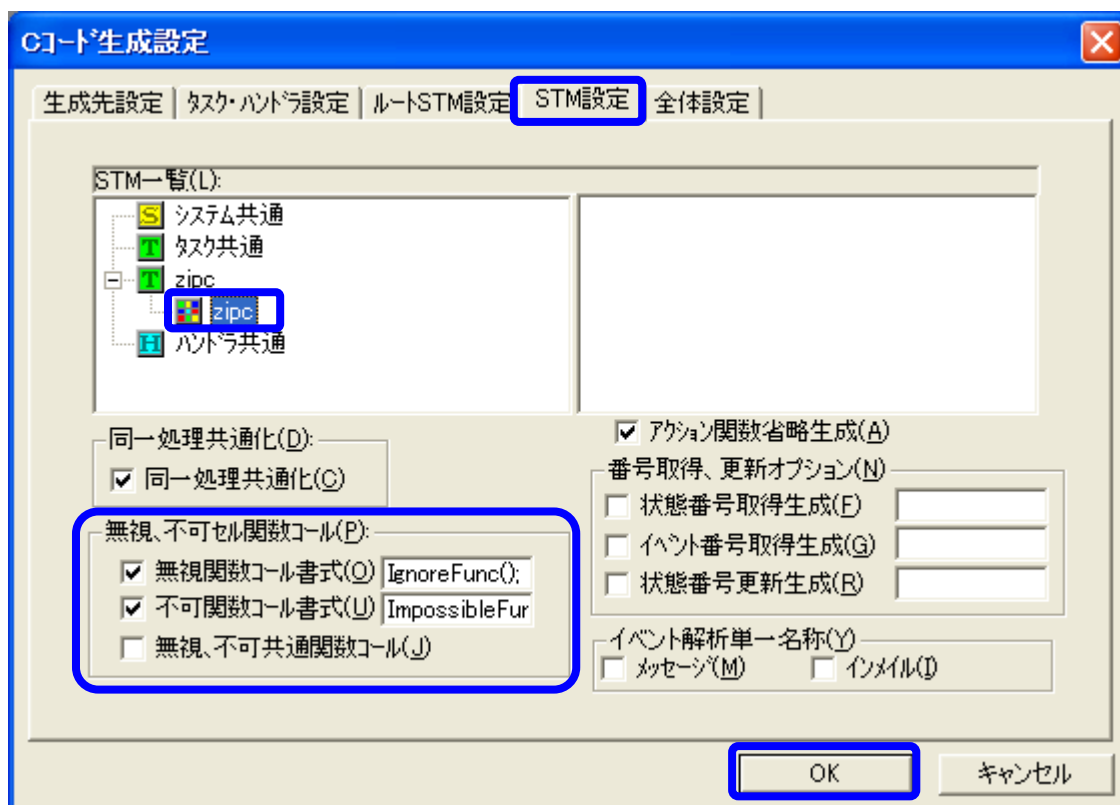
/*ZIPC FUNC HEADER*/
void IgnoreFunc( void )
{
    無視処理
    printf( "無視処理\n" );
}
    
```

※サンプルでは、不可セル実行時に"ImpossibleFunc" 関数をコールし、無視セル実行時に "IgnoreFunc" 関数をコールします。

8. シミュレーションフェーズに移行します。
9. [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] メニューを選択します。



10. [Cコード生成設定] ダイアログにて、[STM 設定] タブを選択します。
11. [Cコード生成設定] ダイアログにて [STM 設定] タブにて、[STM 一覧] 項目から” zipc” STM 設計書を選択します。
12. [Cコード生成設定] ダイアログにて [STM 設定] タブにて、[無視、不可セル関数コール] – [無視関数コール書式] 項目をチェックし、” IgnoreFunc();” と記述します。
13. [Cコード生成設定] ダイアログにて [STM 設定] タブにて、[無視、不可セル関数コール] – [不可関数コール書式] 項目をチェックし、” ImpossibleFunc();” と記述します。
14. [Cコード生成設定] ダイアログにて、[OK] ボタンを選択します。



※不可セル実行時と無視セル実行時に同じ関数を呼び出す場合は、[無視、不可セル関数コール] – [不可関数コール書式] 項目をチェックします。

これで、[無視、不可セル関数コール] 機能の設定は完了です。

3.2 シミュレーション

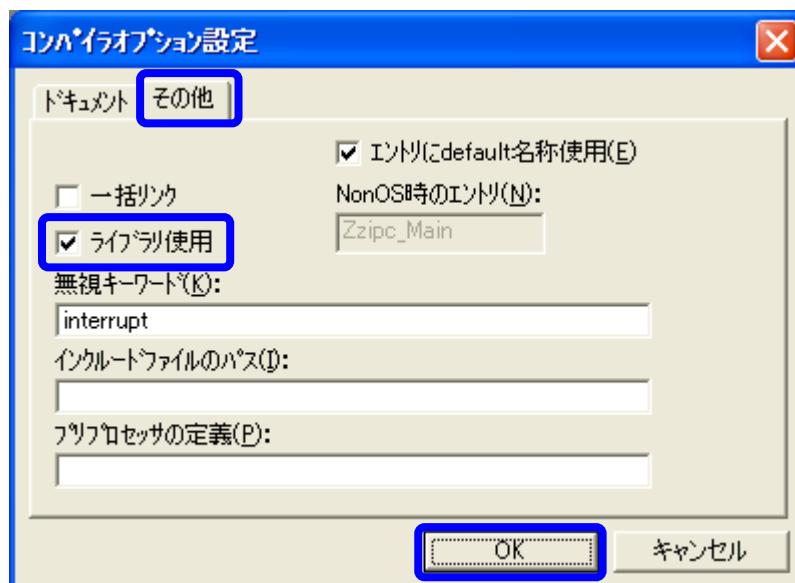
前章までで作成したプロジェクトでは、シミュレーションデバッグすることができません。
本章では、本プロジェクトにてシミュレーションデバッグを行う為に必要な手順を説明します。

■手順

1. [プロジェクト] - [プロジェクト設定] - [シミュレーション実行コード設定] メニューを選択します。



2. [コンパイラオプション設定] ダイアログにて、[その他] タブを選択します。
3. [コンパイラオプション設定] ダイアログにて、[その他] タブにて、[ライブラリ使用] 項目をチェックします。
4. [コンパイラオプション設定] ダイアログにて、[OK] ボタンを選択します。



5. シミュレーション環境を構築します。

これで、正常にシミュレーションすることが可能になります。

※サンプルでは、不可セル および 無視セルを実行すると、メッセージウインドウにメッセージを表示します。

