# 組み込みソフトウェアブロックのヘルプ

## Help Embedded Software Block

Mywayプラス株式会社



#### 1. はじめに

組み込みソフトウェアブロック(Embedded Software Block)は、ユーザーがマイクロコントローラ、 DSP、またはその他のデバイスなどの組み込みデバイスをモデル化できるようにするブロックです。 それは General DLL ブロックの変形です。 General DLL ブロックの使用方法の詳細については、「Help General DLL Block.pdf」のドキュメントを参照してください。

ただし、接続ノードが固定で、入力または出力として事前定義されている General DLL ブロックと は異なり、組み込みソフトウェアブロック(Embedded Software Block)のノードタイプは必要に応じて プログラムできます。

また、変数の状態が変化する正確な瞬間などの追加情報を計算して PSIM に渡すこともできます。 この情報を使用して、固定のタイムステップシミュレーションによって発生する誤差を最小限に抑え ることができます。

#### 2. 組み込みソフトウェアブロック(Embedded Software Block)の使い方

組み込みソフトウェアブロック(Embedded Software Block)の使用方法を説明するために、コンパレ ータ(比較器)の例を使用します。ソースコードは範例を参照します(回路図: "comparator\_ESB.sch"、 source code: "comparator.cpp")。

左側の回路は、コンパレータによって生成されたゲート信号を持つ降圧型コンバータを示していま す。 右側の回路は同じ回路を示していますが、コンパレータの機能は組み込みソフトウェアブロック を使用して実装されています。



この例では、組み込みソフトウェアブロック(Embedded Software Block)に 16 個のポートがありま す。 ポート P0~P7 は入力、ポート P8~P15 は出力です。 端子の数、端子の入出力タイプ、端子名、入力パラメータなどのブロックのプロパティは、カスタム C++コードで定義されています。 このコードには、次のデータ構造と関数が含まれています。

struct Internal\_DLL\_Block\_SimulationData: DLL内で使用される変数は、ここで定義されています。 void REQUESTUSERDATA:ポート番号、名前、およびパラメータを定義する関数。 void OPENSIMUSER:インタフェースからパラメータを読み込み、初期化する関数です。 void STARTSIMUSER:ポートタイプ(入力または出力)を定義する関数です。 void RUNSIMUSER2:PSIM から入力を読み取り、計算を実行し、結果を PSIM に返す関数です。

### 1.1 ポート番号とパラメータの定義

組み込みソフトウェアブロック(Embedded Software Block)を定義する最初の手順は、ポート数とパ ラメータを定義することです。 以下のコードに示すように、それらは関数 REQUESTUSERDATA で 定義され、赤で強調表示されています。

```
void REQUESTUSERDATA(int nRequestReason, int nRequestCode, int nRequestParam, void ** ptrUserData, int * pnParam1,
int * pnParam2, char * szParam1, char * szParam2)
switch( nRequestReason )
{
         case ACTION_DLL_SELECTED:
         {
                  switch(nRequestCode)
                  ł
                           case REQUEST IN OUT NODES:
                                     *pnParam1 = 16;
                                                                         //Define 16 ports
                                    *pnParam2 = 0;
                                    return;
                           case REQUEST_INPUT_NODE_INFO:
                                                                         //Define port names
                                    nNode = nRequestParam;
                                    switch(nNode)
                                     {
                                             case 0:
                                                      strcpy(szParam1, "VDD/P0");
                                                      break:
                                             case 1:
                                                      strcpy(szParam1, "P1");
                                                      break;
                                    }
                                    return;
                           case REQUEST PARAM COUNT:
                                     pnParam1 = 1;
                                                                         //Define 1 parameter
                                    *pnParam2 = 0;
                                    strcpy(szParam1, "All Files|*.*||"); //File Open Dialog Filter for InputFile.
                                    return;
                           case REQUEST PARAM INFO:
                                    switch(nRequestParam)
                                             case 0:
                                                                         //Define parameter name
                                             strcpy(szParam1, "Flag for Exact Switching");
                                             strcpy(szParam2, "1"); //Set default parameter value as 1
                                             *pnParam1 = 1;
                                                                         //Show Display check box
                                             break;
                                              ... ... ... ...
                                    -}
                           . . . . . . . . . . . .
```

### Myway

 1.2 インタフェースからパラメータを読み込む: 次のステップは、インターフェースからパラメーターを読み取ることです。 この場合、正確な切り 替えのためのフラグ(Flag for exact switching)パラメータをひとつ定義します。 このパラメータは、 Embedded Software Block のダイアログウィンドウに表示されます。パラメータ値を DLL に読み込む には、まず DLL 構造体 Internal\_DLL\_Block\_SimulationData に「flag\_exact\_switching」という内部変 数を定義し、次に以下のコードに示すように、パラメータ値を DLL に読み込みます。

```
void OPENSIMUSER(const char *szId, const char * szNetlist, void ** ptrUserData, int *pnError, LPSTR szErrorMsg, void *
pPsimParams)
£
        EXT_FUNC_PSIM_INFO * pPsimInfo = (EXT_FUNC_PSIM_INFO *)pPsimParams;
         assert(*ptrUserData == NULL);
         *ptrUserData = new Internal_DLL_Block_SimulationData;
        Internal_DLL_Block_SimulationData * pData = (Internal_DLL_Block_SimulationData *)(*ptrUserData);
        memset(pData, 0, sizeof(Internal DLL Block SimulationData));
         ... ... ... ...
        pData->m nInputNodes = atoi(netlist[2]);
        pData->m nOutputNodes = atoi(netlist[3]);
        int nParamStartIndex = 5 + pData->m_nInputNodes + pData->m_nOutputNodes;
        // Read the value of Parameter 1 from the interface
        pData->flag exact switching = atoi( netlist[nParamStartIndex] );
        //Initialize internal DLL data
        pData->Vgat0 = 0.;
         *pnError = 0; //Success
```

コードでは、pData は内部 DLL データ構造体へのポインタで、「flag\_exact\_switching」と「Vgat0」 は両方とも構造体内で定義された変数です。 また、変数 Vgat0 の初期化など、内部変数の初期化もこ こで実行されます。

### <u>1.3 ポートタイプの定義</u>

組み込みソフトウェアブロック(Embedded Software Block)のポートタイプはコードで定義する必要 があります。 ポートは入力または出力として定義できます。 この例では、ポート P0 から P7 が入力 で、ポート P8 から P15 が出力です。 それらは、以下のコードに示すように定義されています。

void STARTSIMUSER(int *portTypes, void ** ptrUserData, int *pnError, LPSTR szErrorMsg)		
ì	Internal_DLL_Block_SimulationData * pData = (Internal_DLL_Block_SimulationData *)(*ptrUserData); if( pData == NULL) { return; }	
	for (int i=0; i<8; i++)	
	<pre>portTypes[i] = TYPE_PORT_INPUT; }</pre>	//Define P0 through P7 as inputs
	for (i=8; i<16; i++)	
	<pre>portTypes[i] = TYPE_PORT_OUTPUT; }</pre>	//Define P8 through P15 as outputs
3	*pnError = 0; //Success	

### Myway

<u>1.4 コンパレータ機能の実装:</u>

コンパレータの機能は、すべてのタイムステップで呼び出されるルーチン RUNSIMUSER2 に実装さ れます。 PSIM と DLL の間でデータを転送するために、*ports と ports2* の 2 つの配列が使用されてい ます。 配列 *ports* は各ポートの電圧を保存します。 たとえば、この例では、*ports [1]と ports [2]*にコ ンパレータの非反転入力 (Port P1) と反転入力 (Port P2) が格納され、*ports [12]*にコンパレータの出 力が格納されます (Port P12)。

コード実装は以下のとおりです。

```
void RUNSIMUSER2 (double t, double delt, double *ports, double *ports2, int *portTypes, void ** ptrUserData, int *pnError,
LPSTR szErrorMsg)
         Internal DLL Block SimulationData * pData = (Internal DLL Block SimulationData *)(*ptrUserData);
         if( pData == NULL) { return; }
         double Vm, Vcarr, Vm0, Vcarr0, Vdd;
         int iflag;
         Vdd = ports[0];
                                                                    //Inputs
         Vm = ports[1];
         Vcarr = ports[2];
         iflag = 0;
                                                                    //If iflag=1, output has changed the state.
         if (Vm \ge Vcarr)
                                                                    //Implement the function of a comparator
         {
                   ports[12] = 1.;
                   if (pData ->Vgat0 < 0.5) if lag = 1;
         else
                   ports[12] = 0.;
                   if (pData->Vgat0 > 0.5) if lag = 1;
         if (iflag && pData->flag exact switching)
                                                                   //Calculate the exact switching instant
                   Vm0 = pData -> Vm0;
                   Vcarr0 = pData->Vcarr0;
                   ports2[12] = delt*(Vm-Vcarr)/(Vm-Vm0-Vcarr+Vcarr0);
         pData > Vm0 = ports[1];
                                                                   // Store the value for use in the next step
         pData->Vcarr0 = ports[2];
         pData->Vgat0 = ports[12];
          *pnError = 0; //Success
```

もう1つの配列、ports2は、状態の変化の正確な瞬間の情報を格納するために使用されます。 PS IMは固定タイムステップシミュレーションを使用するので、変数は2つの離散点の間の状態(例え ば、比較器または論理ゲートの出力が0から1に変化する)を遷移します。 説明のために、正確な状 態変化の瞬間と現在の時間との間の時間間隔は*時間誤差(time error)*と呼びます。 したがって、配列 ports2 は各ポートの時間誤差を格納するために使用されます。 ポートが入力の場合、時間エラーは PSIM から DLL に渡され、ポートが出力の場合、時間エラーは DLL から PSIM に渡されます。

時間誤差が考慮されていない場合、特にタイムステップが大きい場合、小さな誤差がシミュレーションに導入されます。 ただし、PSIM は計算で時間誤差を考慮に入れることでシミュレーション誤差を排除できます。

この例では、コンパレータの出力が状態を変更すると、コードは非反転入力と反転入力の現在値と 履歴値を使用して時間誤差を計算します。この値は次に配列 ports2 に格納され、PSIM に渡されます。

誤り訂正の効果を見るためには、瞬時切換えフラグ-パラメータ(flag for switching instant)をOに設定 し、シミュレーションを実行します。 コンパレータを使用する回路の部分には時間誤差補正があり、 組み込みソフトウェアブロック(Embedded Software Block)を使用する回路には時間誤差補正がありま せん。 これら2つのインダクタ電流の間にはわずかな違いがあります。

### 1.5 組み込みソフトウェアブロックのイメージのカスタマイズ

コードがコンパイルされて組み込みソフトウェアブロック(Embedded Software Block)の DLL ファ イルが作成されると、そのブロックを回路図で使用したり、ブロックのイメージをカスタマイズした りできます。 次の手順でブロックをロードして画像をカスタマイズします。

- PSIM で、「素子⇒制御ライブラリ⇒その他の機能ブロック (Elements ⇒ Control ⇒ Other Function Blocks)」に移動します。組み込みソフトウェアブロック(Embedded Software Block)を 選択して回路図に配置します。
- ブロックをダブルクリックし、DLL File 入力欄の右にあるブラウザボタンをクリックして、DLL ファイルを選択します。 この例では、ファイル "comparator ESB.dll"が選択されます。
- DLL ファイルが選択されると、ブロックイメージは長方形に変更され、すべてのノードが左側 に配置され、ノードシーケンスが上から下に表示されます。
- 4) ブロックのサイズを変更するには、ブロックをダブルクリックし、[イメージ編集(Edit Image)] ボタンをクリックして画像エディタに入ります。 描画済みの四角形と左側の矢印線を削除しま す。 次に、「ファイル - >画像サイズの設定」に進み、ブロックの幅と高さを設定します。 こ の例では、幅と高さの両方が7に設定されています。
- 5) ポートの位置を変更するには、最初に「ファイル⇒出力ポートをリセット(File⇒Reset Outputs)」 を選択してすべてのポートをクリアします。次にダ菱形を左クリックしてポートの位置を定義 します。 この例では、たとえば、ポート P0 の位置について左上の2番目の菱形をクリックす ると、最初のポートを表す番号1が菱形の内側に表示されます。
- 6) 描画ユーティリティを使用してブロック画像を再描画します。

#### ご注意

- 1. 本資料に記載された製品の仕様は、予告なく変更することがあります。
- 本資料の内容については、万全を期しておりますが、万一ご不明な点などがあり ましたら、弊社までお申しつけください。
- 3. 本資料に記載された情報に起因する損害または特許権その他権利の侵害に関して は、弊社は一切の責任を負いません。
- 本資料によって第三者または弊社の特許権その他権利の実施権を許諾するもので はありません。
- 5. 弊社の書面許諾なく、本資料の一部または全部を無断で複製することを固くお断 りします。
- 6. (本資料に記載された製品をユーザ装置に組み込む際には、バックアップやフェイ ルセーフ機能をシステム的に設置してください。)
- 7. (弊社は、人命に関わる装置として特別に開発したものは用意しておりません。)
- 8. 本資料に記載されている会社名、商品名は、各社の商標または登録商標です。

Copyright 2019 by Myway Corporation All rights reserved. No part of this manual may be photocopied or reproduced in any form or by any means without the written permission of Myway Corporation.

> 発行 : Myway プラス株式会社 〒220-0022 横浜市西区新花咲町 6-145 横浜花咲ビル TEL : 045-548-8836 FAX : 045-548-8832

ホームページ :<u>https://www.myway.co.jp</u> Eメール :<u>sales@myway.co.jp</u>