



# PLECS *Tutorial*

## Using the PLECS DLL Block

PLECS DLLブロックの使用方法

Tutorial Version: 1.1

# 1 はじめに

PLECSのDLLブロックは、PLECSモデルを使用してカスタムのC制御コードをテストするのに役立ちます。C言語入力ブロックと比較すると、C言語入力ブロックの関数構造に準拠することなく、C制御コードをモデルに含めることができます。制御コードは、DSP制御コードに使用されるものと同じファイル構造を使用して記述できます。

このチュートリアルでは、デジタルPIコントローラを実装したDLLファイルを生成およびデバッグする方法を学習します。

**始める前に** DLLの作成に使用するMicrosoft Visual Studio Communityがインストールされていることを確認してください (インストールするには"C++によるデスクトップ開発"を選択し、次に"Visual C++"開発設定を選択します)。64ビットDLLを生成する場合は、まずMicrosoft Windows SDKと.NET Framework 4をインストールする必要があります。これは、64ビットプラットフォーム用のアプリケーションを作成できる無料パッケージです。インストール中に、.NET コンポーネントのインストールを省略できます。また、DllHeader.h、main.c、dll\_block\_1.plecs、dll\_block\_2.plecsファイルとVisual studio projectsフォルダがすべて作業ディレクトリに配置されていることを確認します。



**注意:** 64ビットDLLはPLECSの64ビットバージョン内でのみ動作します。



**注意:** このモデルには、以下からアクセスできるモデル初期化コマンドが含まれています:

PLECS Standalone: シミュレーションメニュー -> シミュレーションパラメータ... -> 初期化タブ

PLECS Blockset: Simulink model windowで右クリック -> Model Properties -> Callbacks -> InitFcn\*


## 2 DLLファイルの作成

### 2.1 プロジェクトの作成

DLL ファイルを作成するにはいくつかの方法があります。このチュートリアルでは、無料でデバッグ機能も提供している Microsoft のVisual Studio Community 2022を使用します。コードはCで記述しますが、C++も使用できることに注意してください。



**あなたのタスク:**

- 1 Microsoft Visual Studio Communityを実行し、**ファイル**メニューから**新規** -> **プロジェクト**を選択します。プロジェクトの種類として**Windows デスクトップウィザード**を選択します。プロジェクトの"名前"と"ソリューション名"としてpi\_controllerと入力します。準備したファイルをプロジェクトディレクトリ内に配置する必要があるため、プロジェクトディレクトリには任意の場所を選択できます。**作成**をクリックします。
- 2 "アプリケーションの種類"で"ダイナミックリンク ライブラリ(.dll)"を選択し、"追加オプション"で"空のプロジェクト"を選択します。**OK**をクリックします。  
 **注意:** またソリューションエクスプローラーウィンドウ内で"pi\_controller"を右クリックして **プロパティ**に移動し、"プラットフォーム"が"x64"に設定されていることを確認します。
- 3 PLECSインストールディレクトリの...\include\plecsにあるDllHeader.hファイルをプロジェクトコードのディレクトリ...\pi\_controller\pi\_controllerにコピーします。"ソリューションエクスプローラー"ウィンドウの"ヘッダーファイル"フォルダを右クリックして、**追加** -> **ファイル**を選択し、ファイルをプロジェクトのヘッダファイルに追加します。DllHeader.hを選択し、**追加**をクリックします。このヘッダファイルには、PLECSソルバが参照するマクロと関数の定義が含まれています。

- 4 ファイルmain.cをチュートリアル参照ファイルディレクトリからプロジェクトコード ディレクトリにコピーし、前の手順で説明したのと同じ方法でこれをプロジェクトの"ソースファイル"に追加します。このファイルには、PLECSソルバとのインタフェースとなる定型コードが含まれています。

## 2.2 制御コードの記述

制御コードはmain.cのplecsOutput関数に記述します。この関数は、各サンプル周期中に自動的に呼び出されます。PI制御則は次のように定義します:

$$i = i + t_s e \quad (1)$$

$$y = k_p e + k_i i \quad (2)$$

ここで、 $i$ は積分動作、 $e$ は誤差、 $y$ は制御出力です。比例ゲイン $k_p$ と積分ゲイン $k_i$ は外部パラメータとしてDLLブロックに渡されます。

内部状態と外部パラメータといったDLLブロックの入出力には、SimulationState構造体の配列フィールドを使用してアクセスします。aStateという名前のSimulationStateポインタで一般的に使用されるフィールドのリストを以下に示します:

- aState->inputs[n] - DLLブロックへのn番目の入力項目
- aState->outputs[n] - DLLブロックのn番目の出力項目
- aState->states[n] - DLLブロックのn番目の内部状態
- aState->parameters[n] - n番目の外部パラメータ

statesフィールドは、文字列のstatic double変数を格納するための配列です。静的変数はローカルで作成することもできますが、ソルバが定常状態解析またはモデルの初期化のために静的変数にアクセスする必要がある場合は、statesフィールドを使用します。

各配列のサイズnは、Simulation-Sizes構造体のフィールドを使用して指定する必要があることに注意してください。SimulationStateの配列フィールドのサイズを設定するこの構造体のフィールドは、numInputs、numOutputs、numStates、およびnumParametersです。



### あなたのタスク:

- 1 main.cの先頭に、 $k_p$ 、 $k_i$ 、 $t_s$  という名前のグローバルのdouble変数を作成します。関数plecsStart(struct SimulationState\* aState)で、外部パラメータを元の名前にマップし直します。例:  

```
kp = aState->parameters[0];
```
- 2 main.cの先頭にある#defineステートメントを使用して、DLLへの入力をeに、出力をyにマッピングします。defineステートメントはセミコロンで終わらないことに注意してください。例えば:  

```
#define e aState->inputs[0]
```
- 3 また、#defineステートメントを使用して、statesフィールドの最初の要素をiにマップします。これは積分動作を表すために使用します。
- 4 関数plecsSetSizes(struct SimulationSizes\* aSizes)で、配列フィールドSimulationStateのサイズを定義します。例えば:  

```
aSizes->numParameters = 3;
```
- 5 出力関数plecsOutputに式1と式2を実装します。

## 2.3 コンパイルおよび実行



### あなたのタスク:

- 1 **ビルド** -> **ソリューションのビルド**を選択してコードをDLLにコンパイルします。
- 2 このチュートリアルディレクトリからモデルdll\_block\_1.plecsをDLLと同じプロジェクトディレクトリ(...\x64\Debug)にコピーします。
- 3 PLECSでdll\_block\_1.plecsモデルを開き、PI voltage controllerのマスクの下(**Ctrl + U**)を確認し、コンポーネントライブラリ"制御器"のサブライブラリ"関数&テーブル"から"ダイナミックリンクライブラリ"ブロックを"入力信号ポート"ブロックと"出力信号ポート"ブロックの間に配置し、端子を接続します。パラメータ画面を表示させ、[図1](#)のように構成します。
- 4 シミュレーションを実行し、出力電圧と電流をPLECSのデモモデルライブラリのBuck Converter with Peak Current Controlと比較します。

図1: ダイナミックリンクライブラリブロックのパラメータ画面



この段階では、モデルはのチュートリアルディレクトリのdll\_block\_2.plecsと同じ設定で、制御コードは\Visual studio\pi\_controller\pi\_controller\main.cのmain.cファイルと同様のはずです。

## 3 DLLファイルのデバッグ

PLECSモデルで実行中の制御コードをデバッグすることもできます。これはVisual Studio上で実行します。



### あなたのタスク:

- 1 **デバッグ -> プロセスにアタッチ...**でタイトルにモデル名が付いたPLECS.exeを選択して、DLLをPLECSプロセスにアタッチします。コード型に**ネイティブコード**が選択されていることを確認します。そうでない場合は手動で選択します。
- 2 main.cで、 $y = k_p * e + k_i * i$ の行の左側にある垂直バーをクリックして、にブレークポイントを設定します。
- 3 PLECSでシミュレーションを実行します。
- 4 下部の"ウォッチ"ウィンドウで、"自動"タブを選択し、"aState"構造体の入出力フィールドに注目してください。これらのフィールドはeとyを表します。**F5**キーを使用してシミュレーションを実行し、これらの値が変化するのを観察します。

#### 改訂履歴:

1.0	初版
1.1	チュートリアルを64ビットに適応



☎ +41 44 533 51 00

+41 44 533 51 01

✉ Plexim GmbH  
Technoparkstrasse 1  
8005 Zurich  
Switzerland

@ info@plexim.com

<http://www.plexim.com>

#### Pleximへの連絡方法:

Phone

Fax

Mail

Email

Web



☎ +81 3 5282 7047

+81 3 5282 0808

✉ ADVAN AUTOMATION CO.,LTD  
1-9-5 Uchikanda, Chiyoda-ku  
Tokyo, 101-0047  
Japan

@ plecs\_adva@adv-auto.co.jp

<https://adv-auto.co.jp/>

#### アドバンオートメーションへの連絡方法:

Phone

Fax

Mail

Email

Web

#### Application Examples

© 2002–2024 by Plexim GmbH

このマニュアルで記載されているソフトウェアPLECSは、ライセンス契約に基づいて提供されています。ソフトウェアは、ライセンス契約の条件の下でのみ使用またはコピーできます。Plexim GmbHの事前の書面による同意なしに、このマニュアルのいかなる部分も、いかなる形式でもコピーまたは複製することはできません。

PLECSはPlexim GmbHの登録商標です。MATLAB、Simulink、およびSimulink Coderは、The MathWorks, Inc.の登録商標です。その他の製品名またはブランド名は、それぞれの所有者の商標または登録商標です。