



# Embedded Code Generation *DEMO MODEL*

**RGB LED Peak Current Control**  
STM32 MCUでのRGB LEDピーク電流制御

Last updated in STM32 TSP 1.3.1

*Advancing Automation*  
**AUTO  
ADVANCEMENT**

翻訳:

アドバン オートメーション株式会社  
plecs\_adva@adv-auto.co.jp www.adv-auto.co.jp

# 1 はじめに

このデモモデルは、STMicroelectronics [1]のアプリケーションノートAN5345で定義されている高輝度RGB LED制御アプリケーションを、B-G474E-DPOW1ディスカバリキット[2]を使用して実現しています。

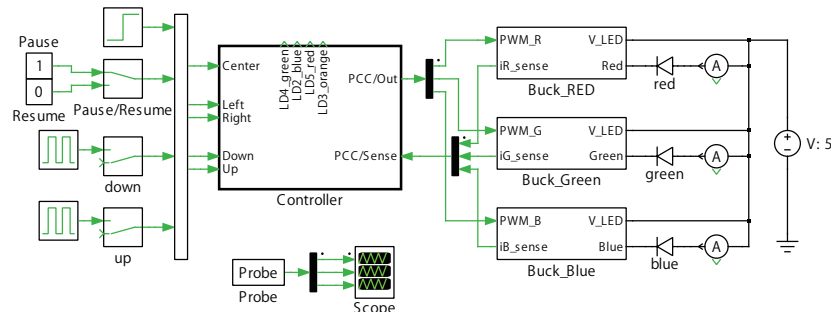
電源回路は、RGB LEDモジュールを駆動する3つの反転降圧型コンバータで構成されています。RGB LEDモジュールは赤色、緑色、青色の3つの独立したLEDで構成されています。これらのLEDは、バーストモード動作によるピーク電流制御方法を使用して個別に制御され、輝度と色を個別に制御します。STM32 Target Supportライブラリには、ピーク電流制御(Peak Current Control: PCC)コンポーネントが含まれています。このコンポーネントは、高分解能タイマ、コンパレータ、デジタル - アナログコンバータなどの複数のMCU周辺機器を統合し、必要なPCC機能を実現します。B-G474E-DPOW1ディスカバリキットのジョイスティックボタンは、PCC出力を有効にし、制御ロジック操作を構成するために使用します。信号LEDはコントローラの動作状態を表示します。

次のセクションでは、モデルの簡単な説明と、モデルをシミュレートして制御コードをSTM32ターゲットにデプロイする方法について説明します。

## 2 モデル

最上位レベルの回路図には、[図1](#)に示すように、電源回路とコントローラが含まれています。"Controller"サブシステムは、右クリックし、**サブシステム -> 実行の設定...**メニューからコード生成に対して有効になります。この手順は、PLECS Coderを経由してサブシステムのモデルコードを生成するために必要です。

図1: モデルのトップレベルの回路図



### 2.1 電源回路

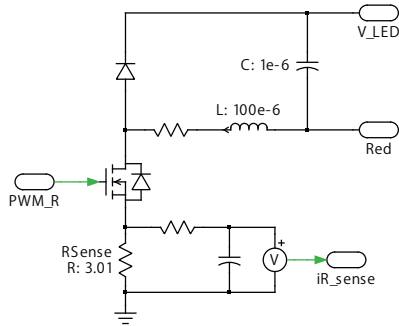
電源回路は、高輝度RGB LEDを駆動する3つの反転降圧コンバータで構成されています。RGB LEDは赤色、緑色、青色の3つの独立したLEDで構成されています。[図2](#)は赤色LED降圧コンバータのトポロジを示しています。緑色および青色LED降圧コンバータは、赤色LED降圧コンバータのトポロジと同一です。

LED負荷は、B-G474E-DPOW1ディスカバリキット[2]のRGB LEDの順方向電流と順方向電圧の値をオシロスコープで測定して得られる非線形V-I曲線で表されます。LEDを右クリックし、**サブシステム -> サブシステムのモデル表示**を選択して、非線形LEDの実装を確認します。

### 電流検出

各LEDの電流は、[図2](#)に示すように、MOSFETのソースピンに接続された検出抵抗*R*<sub>Sense</sub>を介して検出されます。この検出抵抗器の電圧はコントローラにフィードバックされます。

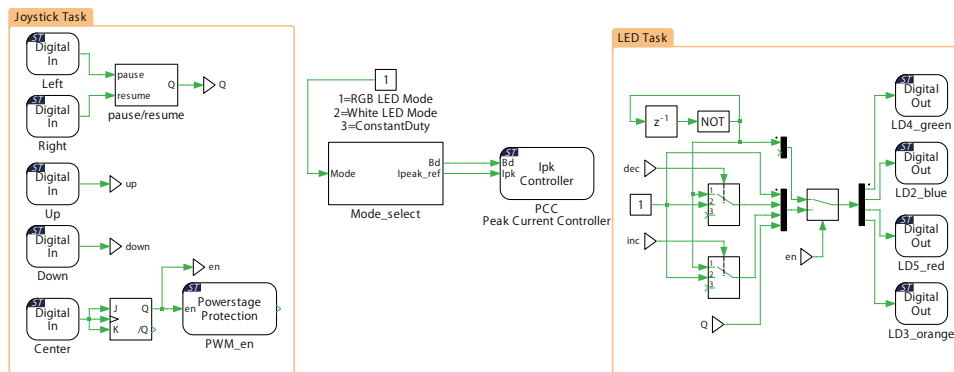
図2: 赤色LED降圧コンバータ



## 2.2 コントロール

"Controller"サブシステムの詳細図を図3に示します。"Peak Current Controller"コンポーネントは電流調整を担当します。"Mode\_select"サブシステムを使用すると、ユーザはRGB LEDモジュールのさまざまな制御モードを選択できます。B-G474E-DPOW1 ディスカバリキットのジョイスティック入力はデジタル入力信号として読み取られ、モードサイクルの開始と停止、グローバル輝度レベルの設定、PWM出力の有効化に使用します。動作モードを表示するインジケータとして4つの信号LEDが使用されます。

図3: Controllerサブシステム



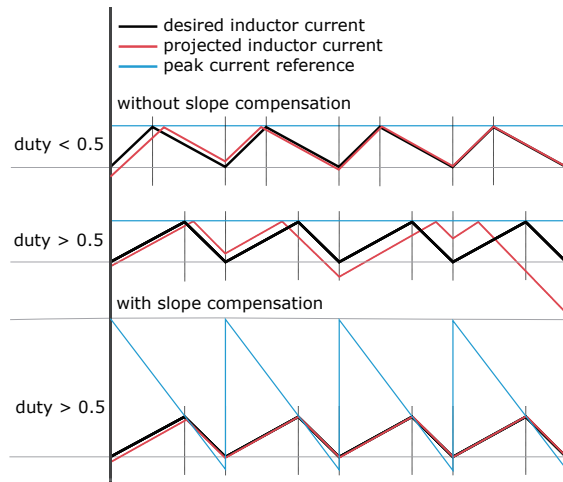
## ピーク電流制御

ピーク電流モードコントローラでは、各スイッチングサイクルの開始時に、事前に決定されたデューティサイクルなしで出力が設定されます(ゲート信号がオンになります)。そして、検知されたインダクタ電流がピーク電流基準値を超えると、出力がリセットされます(ゲート信号がオフになります)。したがって、デューティサイクルはオン時間中のインダクタ電流の上昇によって決まります。

ピーク電流モードコントローラの欠点の1つは、適用されるPWMデューティ サイクルが50%を超えると、不安定になる点です。これを図4で説明します。システムに小さな外乱が導入され、適用されるデューティサイクルが50%未満の場合、外乱は最終的にゼロに減少します。ただし、適用されるデューティサイクルが50%を超えると、インダクタ電流は発散し始め、安定しくありません。結果として生じるデューティサイクル値は、分数調波振動と呼ばれる交互サイクルベースで小さい値から大きい値まで

変化します。これらの低調波振動を制限するため、一定のピーク電流リファレンスを提供する代わりに、[図4](#)に示すように追加のスロープ補償が適用されます。これによりインダクタ電流の安定性が確保されます[\[3\]\[4\]](#)。

図4: スロープ補償



## 輝度と色の制御

このプロジェクトは、輝度と色の両方を個別に制御できるように構成されています。

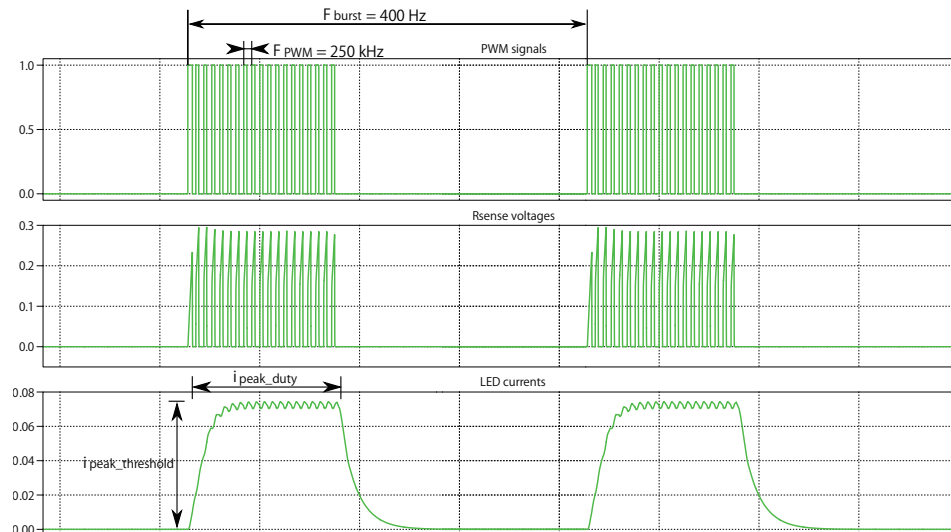
各LEDの全体的な輝度機能は、ピーク電流モード制御を使用して定電流で実行されます。特定の色混合で、調光はバーストモード動作によって実現されます。バーストモード動作では、[図5](#)に示すように、ピーク電流のパルスとゼロ電流の期間(MOSFETをオフにすることによって)が交互に発生します。人間の目が感知するちらつき効果を防ぐために、バーストモード周波数は400Hzに設定されています。LEDの輝度は、ゼロ電流期間に対するピーク電流期間に比例して調整されます。

色混合は、各LEDのピーク電流しきい値によって定義されます。RGB出力が赤色になるには、赤色LEDの電流が最大になり、他のすべてのLED電流はゼロになります。各LEDの電流しきい値が異なる値の場合、RGB出力は異なる色合いになります。たとえば、赤色と青色のLEDのピーク電流しきい値が等しく、緑色のLEDの電流がゼロの場合、紫色が見えるようになります。

PWM生成とピーク電流モード制御は、STM32 targetライブラリのピークPeak Current Controller(PCC)ブロックによって処理されます。ピーク電流コントローラの高分解能タイマ(HRTIM)は、250kHzのPWM周波数を生成するように設定されています。PCC操作の詳細な説明については、[ピーク電流制御](#)を参照してください。

ピーク電流リファレンス(*ipeak\_ref*)は、PCCブロックへの入力として提供されます。ピーク電流リファレンスは、バーストデューティ比(*Bd*)とピーク電流しきい値(*ipeak\_threshold*)の2つのパラメータの関数です。ピーク電流リファレンスのデューティ比(*Bd*)は、すべてのLEDの全体的な明るさを定義し、バーストモード周波数である400Hzで変調されます。デューティ比が高いほど、LEDは明るくなります。ピーク電流しきい値(*ipeak\_threshold*)は、各LEDのピーク電流と相対的な明るさを指定します。

図5: バースト動作によるLEDピーク電流モード制御



## 動作モード

このモデルでは、LEDは3つのモードで駆動できます。図6に示すように、ユーザはモードを切り替えることができます。

- RGB LED Mode: このモードでは、カラースペクトル全体をスキャンします。
- White LED Mode: このモードでは、白色の明るさの範囲全体をスキャンします。同じRGB LEDを使用しますが、赤色、緑色、青色の色が混合されて白色が生成されます。
- Constant Duty Mode: このモードでは、RGB LEDは一定の色と明るさを提供します。

## RGB LED Mode

RGB LEDモードの制御ロジックを図7に示します。このモードでは、図8に示すように、赤色、青色、緑色の各LEDのピーク電流リファレンスの電流しきい値( $i_{peak\_threshold}$ )を個別に調整することにより、RGBカラースペクトル全体をスキャンします。三角形のRGBパターンが現在のしきい値として設定され、一度に2つのLEDのみが有効になります。PLECSでこの三角形パターンを生成するには、ステートマシンブロックを使用します。RGB spectrum scanには、カラースペクトルスキャンを一時停止または再開するロジックも含まれています。

ピーク電流基準のデューティ比( $Bd$ )によって設定されるLEDのGlobal brightness(全体的な明るさ)は、常に一定に保たれます。ただし、ジョイスティックの上下ボタンを使用して、Global brightnessを手動で増減するオプションがあります。これにより、Peak Current Controllerブロックに適用されたバーストモードのデューティイクルが変更されます。

## White LED Mode

White LEDモードの制御ロジックを図9に示します。このモードは、白色輝度範囲全体の自動調光に対応します。3つのRGB LEDがすべて同時に有効になり、白色が得られます。このモードでは、ピーク電流リファレンスのデューティ比( $Bd$ )を三角波を使用して調整し、明るさをパルス的に変化させます。図10に示すように、3つのLEDはすべて同じ $Bd$ を使用して有効になります。

図6: Mode\_select

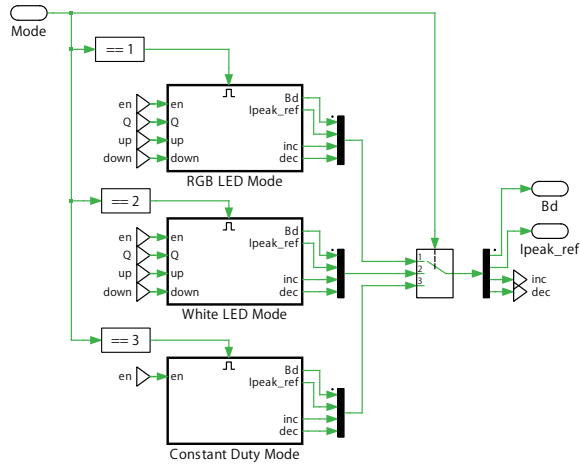


図7: RGB LEDモードの制御ロジック

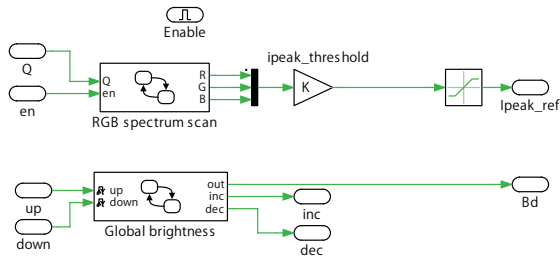
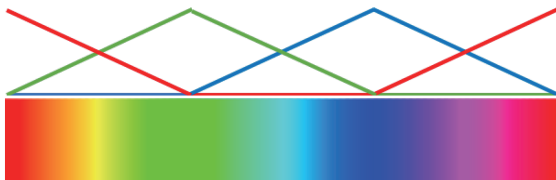


図8: ipeak\_thresholdカラースペクトルスキャン用のR, G, Bパターン [1]



低輝度レベルでは、 $Bd$ 値が0.25未満の場合、現在のしきい値( $ipeak\_threshold$ )も調整されます。これにより、LED ONシーケンスの開始時に全体的な光エネルギーの量が少なくなり、LEDをより効率的に調光できるようになります。明るさが高く、 $Bd$ 値が0.25を超える場合、 $ipeak\_threshold$ は最大値に設定されます。

RGB LEDモードと同様に、Whiteモードにも、ジョイスティックを使用してWhite-brightness range scanで自動的に一時停止または再開したり、LEDの全体的な明るさを増減したりするロジックが含まれています。

図9: White LED Modeの制御ロジック

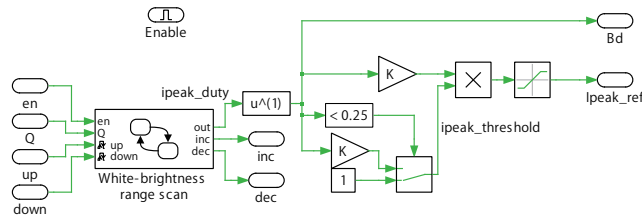
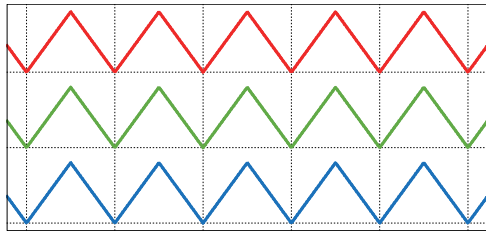


図10: White-brightness range scanのBd R, G, Bパターン



## STM32 Targetライブラリコンポーネントの構成

図3のコントローラには、STM32 Targetライブラリのコンポーネントが含まれています。

- **Peak Current Controller:** モデルで使用する主なSTM32 Targetコンポーネントは、ピーク電流コントローラ(PCC)ブロックです。このコンポーネントは、スロープ補償付きのピーク電流制御を実装します。

内部的には、PCCブロックは複数のMCU周辺機器を使用します(図11を参照)。最初のコンポーネントは、インダクタ電流を制御するためのランプ(*ipeak\_ramp*)を含むピーク電流設定ポイントを提供するDACです。2番目はコンパレータ(COMP)です。図2の*RSense*を介して検出された電流がコンパレータに供給され、DACによって提供される*ipeak\_ramp*と比較されます。COMPブロックの出力は、3番目のコンポーネントである高分解能タイマ(high-resolution timer: HRTIM)に供給されます。HRTIMは250kHzの周波数でPWM波形を生成し、指定された周波数でDACを同期します。バーストモード制御もHRTIMによって処理されます。図12は、これらのピーク電流モードの波形を示しています。

ターンオン時の過渡電流がピーク電流コントローラをトリガするのを防ぐために、リーディングエッジブランク時間が適用されます。図13は、*ipeak\_ref*を0.05に設定してオシロスコープで測定した*RSense\_red*電流を示しています。ターンオン過渡ピーク電流が最終的なインダクタピークよりも大きいことがわかります。リーディングエッジブランク時間を適用すると、この最初のターンオン過渡ピークは無視され、検出されたインダクタ電流が目的のピーク電流基準値を超えるまでデューティサイクルは増加し続けます。図13から、必要なリーディングエッジブランク時間は $2e-7$ 秒と測定されます。

- **Powerstage Protection:**

実行中にPWM信号を有効または無効にするには、ジョイスティックB2の中央の押しボタン(図3のラベルCenter)を使用します。Powerstage Protectionブロックは、ターゲットMCU上のすべてのPWM出力を有効または無効にする有限ステートマシンを実装します。ロジックがLowからHighに遷移するとPWM出力が有効になり、HighからLowに遷移すると無効になります。

詳細については、これらのブロックのヘルプを参照してください。

図11: ピーク電流コントローラの回路図

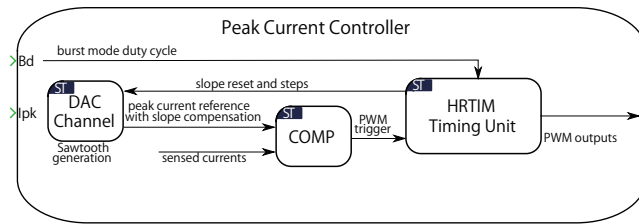


図12: ピーク電流モードの波形

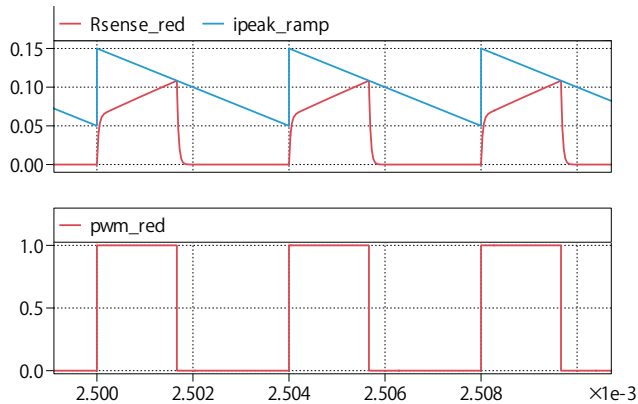
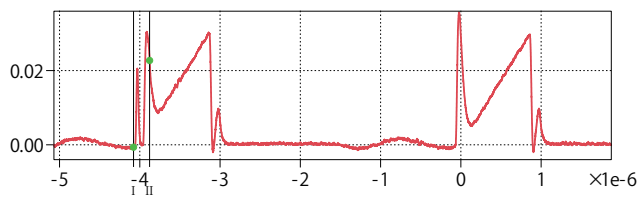


図13: RSense\_red電流の測定値



## マルチタスクコード

PLECS CoderおよびSTM32 Target Support Packageを使用すると、STM32ファミリのMCU用のマルチタスクコードを生成できます。マルチタスクコードにより、さまざまな時間スケールで動的に複数のシステム出力を調整する制御の処理能力を解放します。このモデルでは、ジョイスティックとLEDタスクはピーク電流コントローラに比べて低速で実行できるため、マルチタスクコードを使用しています。

マルチタスクコード生成は、**Coder** -> **Coderオプション...**ダイアログの**タスク**タブで構成されます。**タスクモード**をマルチタスクに変更し、**タスク設定**を指定することにより、各タスクのサンプル時間を設定できます。基本のサンプル時間は常に**離散ステップ**

サイズと同じになります。優先度の低いタスクのサンプリング時間設定は、基本のサンプル時間の整数倍である必要があります。異なる速度で実行される低速で優先度の低いタスクを最大15個指定できるため、アプリケーション内の最速で優先度が最も高いタスクのプロセッサ時間が節約されます。詳細についてはPLECSユーザマニュアル[5]の"コード生成"セクションを参照してください。

PLECS回路図内のブロックは、タスクコンポーネントを使用して、優先度の低いタスクに割り当てられます。このモデルでは、図3に示すように、Base Taskに加えて2つの低優先度タスクが定義されています。以下の表は、Base Taskに対する優先度の低いタスクのサンプル時間を示しています。この場合、Base Task(fc)の実行速度は 20 kHz です。

タスク名	サンプリング時間
Base Task	1/fc
Joystick Task	20/fc
LED Task	1000/fc

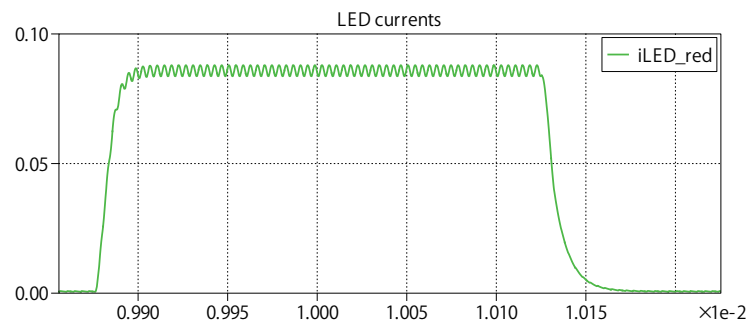
### 3 シミュレーション

#### オフラインシミュレーション

添付のモデルを実行して、LEDとインダクタ電流、およびPWM波形を観察します。必要に応じて、図3のMode\_selectサブシステムのModeというラベルの付いた入力からモードを変更できます。定数Bdが0.1の場合の赤色LEDのLED電流波形を図14に示します。オフラインモデルでは、ユーザはジョイスティックボタンの押下をエミュレートすることもできます。

RGBカラーキャンと白色LEDモードの輝度は数十秒という長い周期で動作するため、オフラインシミュレーションでは変化する波形が表示されない場合があることに注意してください。

図14: 定数Bdが0.1での赤色LEDを流れる電流



#### 組み込みターゲットの操作

このデモモデルのシミュレーションをコンピュータ上でオフライン モードで実行するだけでなく、"Controller"サブシステムをSTM32G474REボードのターゲット固有のコードに直接変換することもできます。

以下の手順に従って、"Controller"サブシステムを STM32 MCU にアップロードします。

- 1 USBケーブル経由でMCUをホストコンピュータに接続します。

- 2 **Coder** -> **Coderオプション...**ウィンドウの左にある**システム**から、"Controller"を選択します。
- 3 次に、**ターゲット**タブで、ドロップダウンメニューからSTM32G4xを選択します。次に、**General**サブタブの**Chip**で、G474REを選択します。
- 4 PLECSからMCUターゲットを直接デプロイするには、Generate code onlyパラメータのチェックを外し、ドロップダウンメニューから目的の**Programming interface**を選択します。デフォルトのプログラミングインタフェースはOpenOCDです。
- 5 次に、**ビルド**をクリックします。

---

**注意** 正しくプログラムされていれば、LD4\_greenというラベルの付いた緑色のLEDが点滅します。

---

STM32CubeIDEに精通している上級ユーザ向けに、Generate code onlyオプションがあります。テンプレートプロジェクトから適切なcgフォルダを見つけ(手順については[6]を参照)、そのパスを**STM32CubeIDE project directory**フィールドに入力して、**ビルド**をクリックします。"Controller"サブシステムのコードが自動的に生成されます。次に、通常のSTM32CubeIDEプロジェクトとしてプロジェクトのビルドとデバッグに進みます。

---

**注意** PWM信号を有効にするには、ジョイスティックの中央ボタンを押します。PWM信号が有効になると、LD4\_greenというラベルの付いた緑色のLEDが点灯します。

---

RGB LEDはRGBカラースペクトル全体をスキャンします。左/右のジョイスティックキーをそれぞれ押すことで、自動スキャンを一時停止/再開することができます。自動スキャンが一時停止すると、LD3\_orangeというラベルの付いた橙色のLEDがONになり、スキャンが再開されるとLEDはOFFになります。

RGB LEDのGlobal brightness(全体の輝度)を増減するには、それぞれジョイスティックの上/下のキーを押します。輝度を上げると、LD5\_redというラベルの付いた赤色LEDがONになります。そして明るさの上限に達するとすぐにLD5\_redが点滅し始めます。同様に、Global brightnessを下げると、LD2\_blueというラベルの付いた青色LEDがオンになり、Global brightnessの下限に達すると、LD2\_blueが点滅し始めます。

RGB LED ModeからWhite LED Modeに切り替えるには、以下の手順に従ってSTM32 MCUの外部モードに接続します。

- 1 **Coder** -> **Coderオプション...**ウィンドウの左にある**システム**から、"Controller"を選択します。
- 2 次に、**外部モード**タブから適切なターゲット デバイスを選択し、**接続**をクリックします。モデルのデフォルトの通信リンクはSerial over GDBに設定され、デバイス名は127.0.0.1に構成されています。
- 3 必要に応じて、**自動トリガを有効化**にして、"Controller"サブシステムのPLECSスコープでテスト結果を観察します。
- 4 次に、"Controller"回路図に戻り、[図3](#)のMode\_selectサブシステムへのModeに接続している定数を2に設定します。

RGB LED Modeと同様に、White LED Modeでも自動スキャンを一時停止/再開したり、LEDの全体の輝度を増減したりすることができます。

## 4 まとめ

このモデルは、STMicroelectronics社のアプリケーションノートAN5345で定義されている高輝度RGB LED制御アプリケーションを、B-G474E-DPOW1ディスカバリキットを使用して実証するものです。

## 5 参考文献

[1]AN5345 High-brightness RGB LED control using the B-G474E-DPOW1 Discovery kit.

[2]B-G474E-DPOW1 Discovery kit with STM32G474RE MCU,

URL: <https://www.st.com/en/evaluation-tools/b-g474e-dpow1.html>.

[3]AN5497 Buck current mode with the B-G474E-DPOW1 Discovery kit.

[4]NPTEL lectures from Indian Institute of Science, Bangalore. Click to access online:

[Slope compensation for current control](#)

[5]PLECS User Manual,

URL: <https://www.plexim.com/sites/default/files/plecsmanual.pdf>

日本語版は以下から:

<https://adv-auto.co.jp/products/plexim/manual.html>

[6]PLECS STM32 Target Support User Manual,

URL: <https://www.plexim.com/download/documentation>

日本語版は以下から:

<https://adv-auto.co.jp/products/plexim/manual.html>

改訂履歴:

STM32 TSP 1.0.1 初版  
STM32 TSP 1.3.1 ネイティブのバーストモードコントローラを使用

 Pleximへの連絡方法:

☎ +41 44 533 51 00 Phone  
✉ Plexim GmbH Mail  
Technoparkstrasse 1  
8005 Zurich  
Switzerland  
@ info@plexim.com Email  
<https://www.plexim.com> Web

Advancing Automation  
 アドバンオートメーションへの連絡方法:

☎ +81 3 5282 7047 Phone  
✉ ADVAN AUTOMATION CO.,LTD Mail  
1-9-5 Uchikanda, Chiyoda-ku  
Tokyo, 101-0047  
Japan  
@ info-advan@adv-auto.co.jp Email  
<https://adv-auto.co.jp/> Web

*Embedded Code Generation Demo Model*

© 2002–2026 by Plexim GmbH

このマニュアルで説明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の書面による事前の同意なしに、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。

本マニュアルは、Plexim社の英文マニュアルを日本語に翻訳したものです。本マニュアルと英文マニュアルとで差異がある場合、英文マニュアルを正とします。

本マニュアルの内容に基づいて発生した負傷や損害などに対して、Plexim GmbHおよびアドバン オートメーション株式会社は一切責任を負いません。製品とアプリケーションに関連したリスクを最小限に抑えるため、ユーザが適切な設計および保護対策を用意する必要があります。