

第5章 ソフトウェアからのハードウェア制御の第一歩

Cプログラムから使える
LED制御回路の設計

来栖川 智久 Tomohisa Kurusugawa

ここでは、オリジナルのZynqボード向けに第4章で構築したプラットフォームを使って、ソフトウェアからハードウェアを制御してみます。例題は、ハードウェア制御の入門では定番のLED制御です。とはいえここで言うのは単なる点滅ではありません。プラットフォームに組み込まれているIPコアの機能を活用して明るさを変化させてみます。

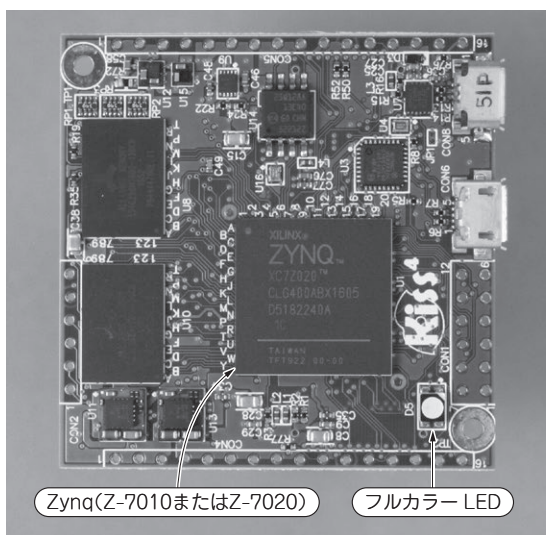


写真1 ターゲットのZynqボード

ハードウェア機能であるFPGAを搭載したボードの最初の動作確認では、1個のスイッチ入力の状態をLEDに出力する「LED点灯（通称Lチカ）」を行うことがよくあります。

SDSoC対応のプラットフォームでは、多くの場合、C言語によるソフトウェア機能を検証し、検証後にハードウェア機能によるアクセラレーションを目的としているはずですが、そのため、SDSoCを使った動作検証では、C言語によるデータ・ムーバやアルゴリズム実装が先に行われることになるかもしれません。

ですが、ZynqはFPGA機能も持ち、多くのボードにはLEDが搭載されています。そして、SDSoCに対応していれば、LEDをアクセスするための環境も用意されているはずですが。

本章では、第4章で作成したプラットフォームに組み込まれているLEDドライバの詳細と、C言語によ

るソフトウェアからのハードウェアの制御方法を説明します。

1. 調光機能を持つLED制御コア

写真1のボードには、1個のフルカラーLEDと4個の青色LED（裏面）が搭載されています。

フルカラーLEDは、R/G/Bの各色が点灯可能であり、各色の明るさ（輝度）を組み合わせることで、さまざまな色を表すことができます。そのため、フルカラーLEDは、LEDのON（点灯）とOFF（消灯）の2つの状態を制御するのではなく、輝度調整用の調光機能が必要になります。

● ON/OFF時間で明るさが決まる

調光機能は、ある一定の短い周期の中で、点灯と消灯の時間を変えることによって明るさを制御する方法を用いています。基本周期の中でのONとOFFの時間をパルス幅として生成する方法であるため、この方式をPWM（Pulse Width Modulation；パルス幅変調）といいます。

今回は基本周期を5ms、そしてLEDの点灯と消灯の分解能を256と設定しました。

$$5[\text{ms}] \div 256[\text{分解能}] = 19.5[\mu\text{s}]$$

になります。

そして、この分解能の時間を作るために、Zynqの内部バス・クロックである50MHzを977分周して作成します（図1）。

● レジスタに書き込むだけで明るさが変わる

LED制御コアのレジスタは、フルカラーLEDのR/G/Bと4個の青色LED用におのおの1つずつ用意されています。つまり、合計7個のPWM値設定用レジスタに対して、ソフトウェアがアクセスを行うことで調光制御付きLEDの駆動を行うことができます。

第4章で説明したプラットフォームでは、調光機能