このページはソーラボジャパン株式会社によるＰＲ記事です。

ソーラボ製品を Raspberry Pi で制御してみよう

須河 光弘（ソーラボジャパン株式会社　技術営業部）

顕微鏡をはじめとする光学装置を自作するメリットは、サイズ、コスト、機能などを用途に合わせてカスタマイズできる自由度の高さにあります。構築にあたっては、光学的な仕様設計に加え、制御系の選定も重要になります。ソーラボ製の多くのコントローラは外部トリガ入力端子を備えており、製品によっては専用ソフトウェアや SDK（ソフトウェア開発キット）も提供されています。これらを活用すれば、光源、ステージ、カメラなどをプログラムから制御し、ユーザー独自の光学装置を開発することが可能です。

本稿では、Linux ベースの OS を搭載した、安価でコンパクトなシングルボードコンピュータである Raspberry Pi を用いてソーラボ製品を制御する方法について概説します。Raspberry Pi はオンボードで GPIO 端子を備えており、デジタル信号や PWM 信号を利用して、外部トリガ入力を備えた機器を比較的シンプルな構成で制御できることも特長の 1 つです。

倒立型電動蛍光顕微鏡の自作構成例

光学装置の 1 つとして顕微鏡を考えてみましょう。例えば、ソーラボ製品を組み合わせることで、図 1 のようなコンパクトな倒立型電動蛍光顕微鏡を構築できます（自作顕微鏡については弊社の別稿をご参照ください[1, 2]）。この顕微鏡には以下のデバイスが使用されています。

- LED ドライバ LEDD1B
- モーターコントローラ K-Cube シリーズ KST201
- CMOS カメラ LP126MU/M

LED ドライバで励起用 LED を On/Off し、K-Cube コントローラでサンプルステージを移動させ、カメラで蛍光像を撮影します。コンピュータからこの 3 つのデバイスを制御できれば、タイムラプス撮影などの自動化へと発展させることもできます。



図 1. 倒立型蛍光顕微鏡の自作例

Raspberry Pi と Python

Raspberry Pi は、クレジットカードほどの大きさでありながら、Linux ベースの OS、HDMI 出力、Ethernet や Wi-Fi によるネットワーク接続機能、そして GPIO（General Purpose Input/Output）端子を備えたシングルボードコンピュータです。一般的な Windows PC には GPIO 端子が備わっていないため、外部機器制御用のデジタル信号を生成するには DAQ や Arduino などの I/O デバイスが必要になります[3]。一方、Raspberry Pi は GPIO 端子を備えており、コンピュータから直接デジタル信号を扱えるため、電子工作から研究機器の制御まで幅広く利用されています[4]。

また、Raspberry Pi には Python 開発環境の Thonny が標準でインストールされており、手軽に Python プログラミングを始めることができます[5]。

GitHub ソーラボ公式アカウント

GitHub のソーラボ公式アカウントでは、ソーラボ製品の代表的なコントローラのサンプルプログラムを C++、C#、LabVIEW、MATLAB、Python で公開しています[6]。サンプルを活用して各デバイスの動作確認を行い、それをもとにユーザー独自のプログラムを開発できます。

LED ドライバ LEDD1B の制御

LEDD1B は外部トリガ信号により LED の On/Off を制御できます。ここでは Raspberry Pi の GPIO18 から PWM 信号を出力し、LEDD1B のトリガ入力に使用します。GPIO18 と GND を LEDD1B の BNC 端子に配線し、本体のスイッチを TRIG に設定します（図 2）。PWM 制御用の Python ライブラリ（rpi-hardware-pwm [7]など）を利用して、LED の On/Off を制御します。



図 2. Raspberry Pi 4 Model B と LEDD1B の配線例

モーターコントローラ K-Cube シリーズの制御

K-Cube の制御に使用される Kinesis ソフトウェアは Windows 向けですが、モーションコントローラは低レベル通信プロトコル（Communication Protocol）による制御にも対応しています。これは OS に依存しないシリアルコマンド方式です。Communication Protocol のマニュアル(PDF)は弊社ウェブサイトよりダウンロードいただけます。モーションコントローラ製品には FTDI 製 USB-シリアル変換チップが使われていますが、Raspberry Pi 4 Model B などでは対応ドライバが Linux カーネルに標準搭載されています。

KST201 用のシリアルコマンドのサンプルプログラムはありませんが、同じ K-Cube シリーズの DC サーボモーターコントローラ KDC101 向けサンプル KDC101_serial.py が GitHub で公開されています[8]。両機種では多くのコマンドが共通するため、このサンプルを出発点として動作を確認できます。ただし、このプログラムで使われている移動コマンド MGMSG_MOT_MOVE_ABSOLUTE の入力値（カウント数）は、使用するアクチュエータに合わせて変更する必要があります。詳細はマニュアルをご確認ください[9]。移動量の入力値は、32 ビット符号付き整数の 2 の補数をリトルエンディアンで指定します。例えば、-123456789 は「EB 32 A4 F8」となります。

カメラの制御

カメラ制御用ソフトウェア ThorImageCAM 向けに Linux 用 SDK が用意されており、弊社ウェブサイトよりダウンロード可能です。Raspberry Pi では、SDK に含まれる AArch64 用の共有ライブラリを使用します。まず、SDK 付属のインストール手順書（THORLABS_SDK_INSTALLATION_INSTRUCTIONS.txt）に従って、共有ライブラリをインストールします。次に、Python_README.txt を参照して、Python SDK をインストールします。準備が完了したら、付属サンプルの polling_example.py などを実行し、カメラが認識され、画像を取得できることを確認します。

ここまでの手順で、Raspberry Pi を用いて、LEDD1B による LED 光源の On/Off 制御、KST201 による電動ステージの移動、カメラによる撮影をそれぞれ動作確認できます。次のステップとして、各デバイスを連動させるプログラムを開発すれば、タイムラプス撮影や多点撮影へと発展させることができます。Raspberry Pi を入口に、研究目的に合わせた装置制御を試してみてはいかがでしょうか。

*Raspberry Pi は Raspberry Pi Ltd.の商標または登録商標です。Python は Python Software Foundation の商標または登録商標です。その他記載の会社名および製品名は各社の商標または登録商標です。

[1] 『ケージシステムや光学レールを使ってみよう』
[2] 『CAD を使って光学装置を設計してみよう』
[3] 『サイエンティフィックカメラと周辺機器の同期』
[4] https://www.raspberrypi.com/
[5] https://thonny.org/
[6] https://github.com/Thorlabs
[7] https://pypi.org/project/rpi-hardware-pwm/
[8] https://github.com/Thorlabs/Motion_Control_Examples/tree/main/Python/Serial%20Command
[9] Communication Protocol, Issue 48, Introduction, 8

THORLABS
ソーラボジャパン株式会社
〒179-0081 東京都練馬区北町3-6-3