

第2章

PCからはHIDデバイス&マス・ストレージとして見える

USB キーボードを CircuitPythonでサッと作る

編集部

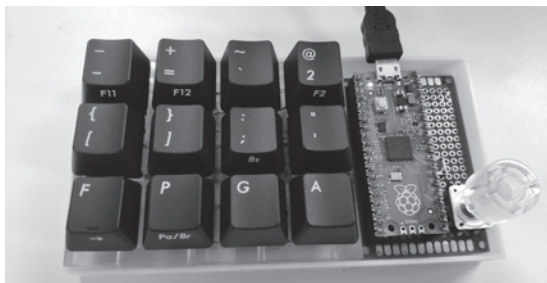


写真1 PicoならPythonで手軽にオリジナル・キーボードを作れる

● 自作USBキーボード

ラズベリー・パイPico(以降Pico)はUSBのホストにもデバイスにもなる機能を持っています。

開発に利用する言語にもよりますが、USB接続のキーボードやマウスについては、ライブラリがあるので手軽に自作できます(写真1)。

● デバイス・ドライバが豊富なCircuitPython

2022年8月現在、MicroPythonを使ったUSB HID(Human Interface Device)デバイスの製作事例はあまり見あたらないようです。

MicroPythonと似た処理系としてCircuitPythonがあります。これはマイコン・ボードやセンサ・ボードなどを販売するAdafruitが、自社のデバイス用にMicroPythonを拡張して作ったものです。

CircuitPythonには、既に多くのデバイス・ドライバが用意されており、USB HID用のドライバもある

ためキーボードやマウスを簡単に作れます。

USBキーボードの構成

図1にCircuitPythonを使う場合のシステム全体構成を示します。

● CircuitPythonファームウェアを書き込む

Picoに書き込むのは、CircuitPythonのファームウェアです。書き込み後、Picoの中にCircuitPythonのプログラムを実行できる環境ができます。

● プログラムは自動実行される

ルート・ディレクトリにcode.pyというファイル名でプログラムを保存しておくとし、Picoに電源が入ったときに自動で実行されます。

● PCからはマス・ストレージ・デバイスに見える

PCにUSBで接続するとストレージとして認識されるので、プログラムや画像などのリソース・ファイルをエクスプローラなどで読み書きできます。

ちなみに、上記のcode.pyをメモ帳などのエディタで編集後、上書き保存すると、自動でPicoが再起動しcode.pyを実行するので、トライ&エラーの開発が非常にやりやすいです。

USBキーボードとして動作させたまま、CircuitPythonのプログラムを編集できるので、キー・バインディング(キーボードの物理キーと発行される

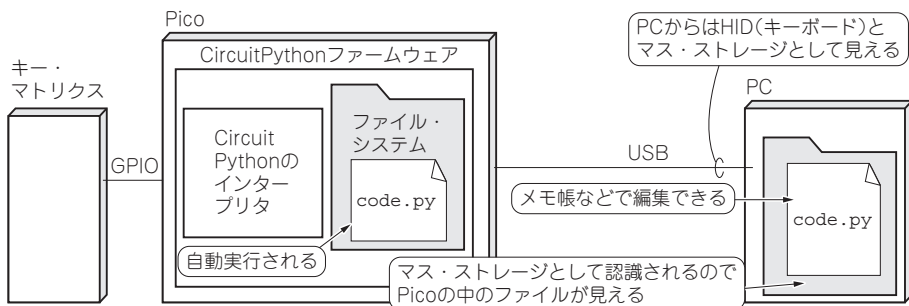


図1 Picoを使ったUSBキーボードの全体構成