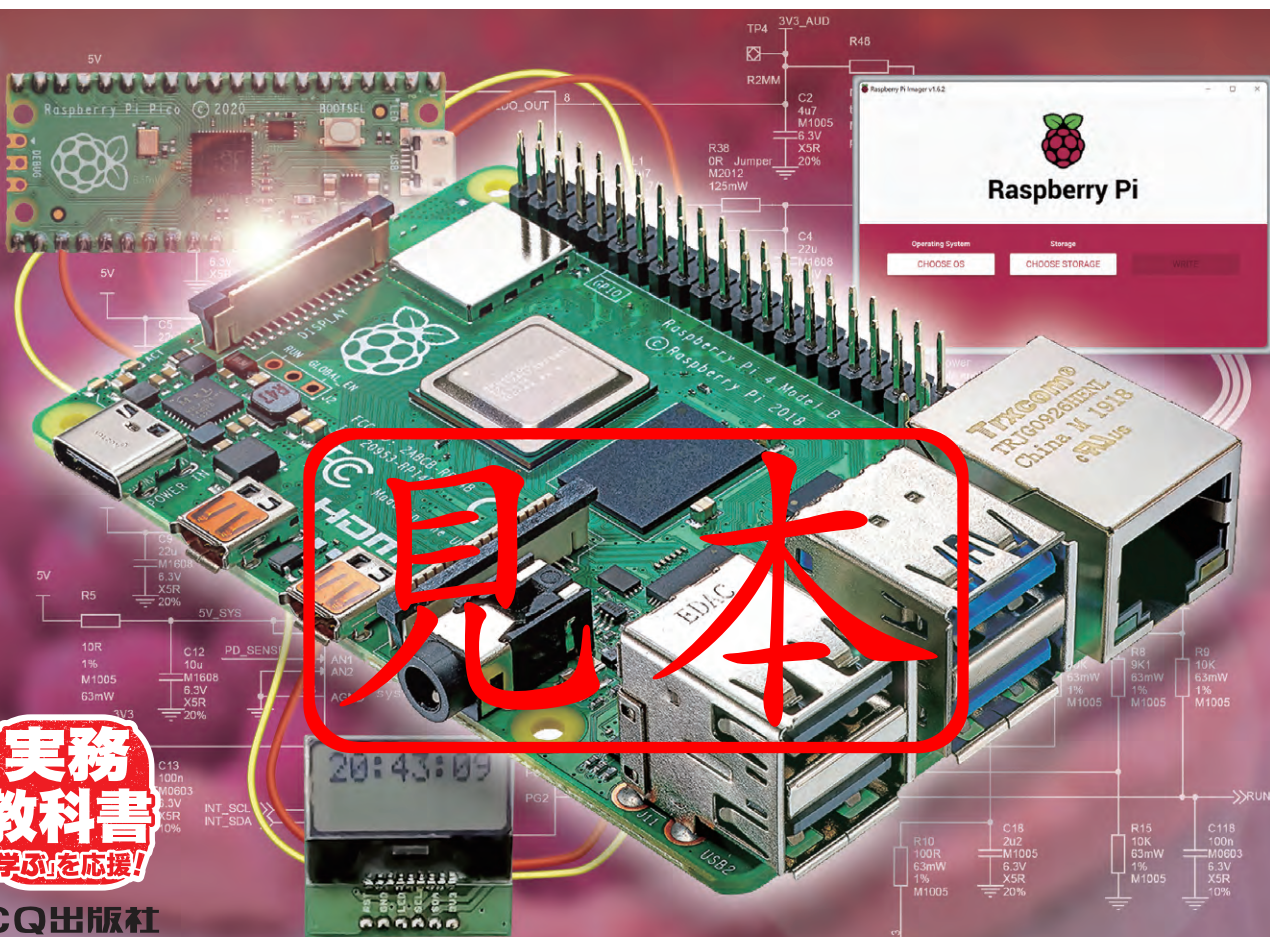


トランジスタ技術 SPECIAL

ハードを動かすメカニズム理解! ネットワーク&カメラまで

ラズパイI/O制御 図解 完全マスタ



**実務
教科書**
「学ぶ」を応援!

CQ出版社

Introduction 1 世界中で使われているラズベリー・パイ大図鑑

定番コンピュータ・ボード 「ラズパイ」の世界

永原 柊 Shu Nagahara

定番ラズベリー・パイ 4モデルB

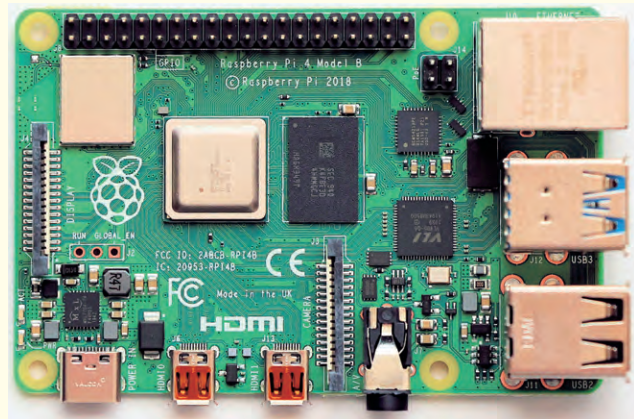


表1 ラズベリー・パイ 4モデルBの主な仕様

項目	値など	項目	値など
SoC	BCM2711	有線LAN	10/100/1000
CPU	Cortex - A72	Wi-Fi	IEEE 802.11
コア数	4		b/g/n/ac
クロック	1.5 GHz		2.4/5 GHz
RAM	1, 2, 4, 8 Gバイト	Bluetooth	○
ストレージ	microSD	カメラ	MIPI CSI-2
USB 2.0/3.0	2/2	ディスプレイ	MIPI DSI
GPIO	40ピン	HDMI	Micro×2

写真1 ラズベリー・パイ 4モデルB

ラズベリー・パイ (Raspberry Pi, ラズパイ) は教育目的で開発された、世界中で使われているコンピュータ・ボードです。その中でラズパイ 4B は、初代モデル B から B+, 2B, 3B, 3B+ と続いている、モデル B 系列の最新版です (2023 年執筆時点)。高性能化により、組み込み用途だけでなく、パソコンとしても十分使えるようになった、といえます。特に教育用途であれば、何の問題もないでしょう。本書でもこのボードを使用しています。外観を写真 1 に、主な仕様を表 1 に示します。

ラズパイ 4B は、従来のものに比べてさまざまな点で改良が加えられています。

▶ **高性能化**

SoC が変更され、CPU が Cortex-A72 ベースになり、クロックも高速化しています。

▶ **RAM 容量のバリエーション化**

RAM はラズパイ 2 以降 1G バイトしか選択肢がありませんでしたが、ラズパイ 4B では 1G バイト、2G バイト、4G バイト、8G バイトが用意されました。

▶ **USB 3.0 対応**

USB コネクタは 1B 以降変わらず 4 個ですが、4B ではそのうち 2 個が USB 3.0 対応になっています。

▶ **ディスプレイ 2 系統化**

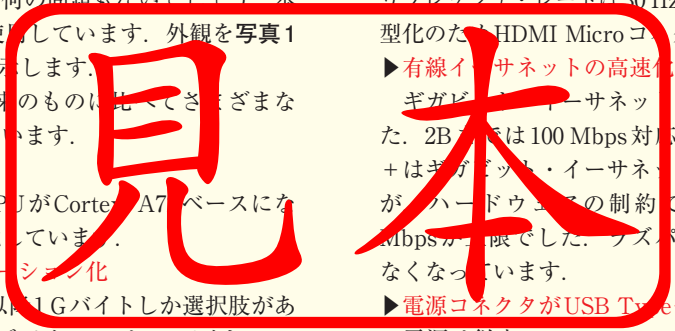
ラズパイ 4B で初めてディスプレイ・コネクタが 2 系統用意され、しかも解像度は 4K に対応しました。ただし 2 台のディスプレイで同時に 4K 表示すると、リフレッシュ・レートは 30 Hz になります。また、小型化のため HDMI Micro コネクタに変更されました。

▶ **有線イーサネットの高速化**

ギガビットイーサネットに本格的に対応しました。2B 以降は 100 Mbps 対応でした。ラズパイ 3B+ はギガビット・イーサネットに対応していましたが、ハードウェアの制約でスループットは 300 Mbps が上限でした。ラズパイ 4B ではその制約がなくなっています。

▶ **電源コネクタが USB Type-C に変更**

電源は従来の USB Micro-B コネクタに代わって USB Type-C コネクタになり、スマートフォンの充電器などを流用しやすくなっています。



細かいバージョンを入れるともっと種類があるのですが、ここでは主要なものに絞りました。

小型系列 ラズベリー・パイ Zero 2 W

2023年執筆時点で最後に発売されたラズパイで、小型のZero系列です。外観を写真2に、主な仕様を表2に示します。

Zero Wと同じ無線機能を搭載したまま、SoCをRP3A0というカスタム品に変更しました。このSoCのCPUは、ラズパイ3 A+と同じ4コアのCortex-A53に強化されています。CPUクロックは1 GHzです。RAM容量は512 Mバイトで、ラズパイZero W

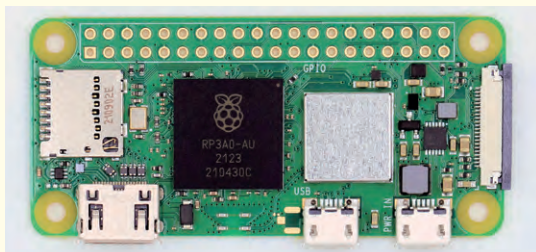


写真2 ラズベリー・パイ Zero 2 Wの外観

から変更ありません。このように書くと、ラズパイZero Wとの違いが小さいように思うかもしれませんが、しかしZero Wは2012年の初代ラズパイと同じSoCを使っているため、性能が低く抑えられています。

それに対してZero 2 WではSoCを更新することで、少々重い処理でも実用的な性能で実行できます。公式サイトには、若干宣伝文句のようですが、Zero Wより5倍高速化したという記述があります。

表2 ラズベリー・パイ Zero 2Wの主な仕様

項目	値など	項目	値など
SoC	BCM2710A1	有線LAN	-
CPU	Cortex-A53	Wi-Fi	IEEE 802.11
コア数	4		b/g/n
クロック	1 GHz		2.4 GHz
RAM	512 Mバイト	Bluetooth	○
ストレージ	microSD	カメラ	MIPI CSI-2
USB 2.0/3.0	1/-	ディスプレイ	-
GPIO	40ピン	HDMI	ミニ

ラズパイという名のマイコン・ボード Pico

ラズベリー・パイという名前は付いていますが、Linuxは動かず、純然たるマイコン・ボードです(写真3、表3)。マイコン・ボードとしてはシンプルな作りで、この半導体不足の状態でも入手は容易です。

ラズベリー・パイ財団が新たに開発した、RP2040



写真3 ラズベリー・パイ Picoの外観

というマイコンを搭載しています。CPUは2コアのArm Cortex-M0+です。RP2040単体でも販売されています。

表3 ラズベリー・パイ Picoの主な仕様

項目	値など	項目	値など
SoC	RP2040	GPIO	40ピン
CPU	Cortex-M0+	有線LAN	-
コア数	2	Wi-Fi	-
クロック	133 MHz	Bluetooth	-
RAM	264 Kバイト	カメラ	-
ストレージ	2 Mバイト	ディスプレイ	-
USB 2.0/3.0	1/-	HDMI	-

Wi-Fi付き Pico W

PicoにWi-Fi(2.4 GHz、802.11 n)を追加したボードです。外観を写真4に、主な仕様を表4に示します。

Wi-Fi搭載以外には、Picoと大きな違いはありません。ただし、2023年執筆時点では入手できるボードには技適表示がありません。このボードのWi-Fiを

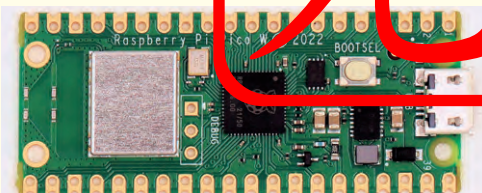


写真4 ラズベリー・パイ Pico Wの外観

を使う場合は電波暗室内で動かすか、特例制度を使うなどの対策が必要です。

表4 ラズベリー・パイ Pico Wの主な仕様

項目	値など	項目	値など
SoC	RP2040	GPIO	40ピン
CPU	Cortex-M0+	Wi-Fi	IEEE 802.11 n
コア数	2		2.4 GHz
クロック	133 MHz	Bluetooth	-
RAM	264 Kバイト	カメラ	-
ストレージ	2 Mバイト	ディスプレイ	-
USB 2.0/3.0	1/-	HDMI	-
有線LAN	-		

ラズパイを生かす… 本書のコンセプト

永原 柊 Shu Nagahara

2012年に初代Raspberry Piが衝撃的に登場してから10年以上が経過しました。登場した当初は入手まで半年待ちといった状況でしたが、今では容易に入手でき、幅広く利用されています。

いろいろなところで活用事例も多数紹介され、具体的な操作手順についても解説されています。また、Raspberry Pi OSの中核であるカーネルに関する説明も出版物やブログなどで見られます。

このように数多く用いられ、活用のための情報もそろっているように思えますが、実際に何か動くものを作ろうとすると、途端にわからないことが出てくるのではないのでしょうか。

そこで、もう少し楽に取り組めるようにならないかと考えて本書を執筆しました。

本書では、まずコマンド入力でのLEDなどの対象を操作します。次にそのコマンド操作を基に、単純な機能のプログラムとして実現します。そして、その単純な機能の組み合わせにより、より複雑なものを作っていきます。

言わば、まず単機能のソフトウェアの部品を作り、次にその部品を組み合わせて動くものを作る、という感じですが、本書がみなさまの困りごとに少しでも役立てば幸いです。



ラズベリー・パイのハードウェア構成

永原 柊 Shu Nagahara

Raspberry Pi(ラズベリー・パイ, 以下ラズパイと表記)は, もともとプログラミング学習用の安価なコンピュータとして開発されました. 安価で汎用性が高く, 幅広い用途に用いられています.

ラズパイは初代から2, 3, 4と強化されてきました.

またZeroという小型のタイプや, 組み込み用途の「Compute Module」も用意されています.

ラズベリー・パイ4ともなると, ちょっとしたパソコンとして使えます. 実際には, キーボード一体型のパソコン「ラズベリー・パイ400」も販売されています.

1 ラズベリー・パイのハードウェア構成

本書で使用したラズパイを図1, 写真1に示します. 主要な部品は表面 [写真1(a)] に実装され, 裏面 [写真1(b)] は microSD カード・ソケットが目立つ程度です.

外部とのインターフェースが充実しているのが特徴です. 特に40ピンの拡張端子は, 本書で扱うような

電子工作を行うには欠かせません.

SoC(System-on-a-Chip)にはBCM2711を搭載しており, 最高1.5 GHzで動作する Arm Cortex-A72×4をCPUのコアとして, GPIOやI²C, SPIなどのインターフェースを内蔵しています.

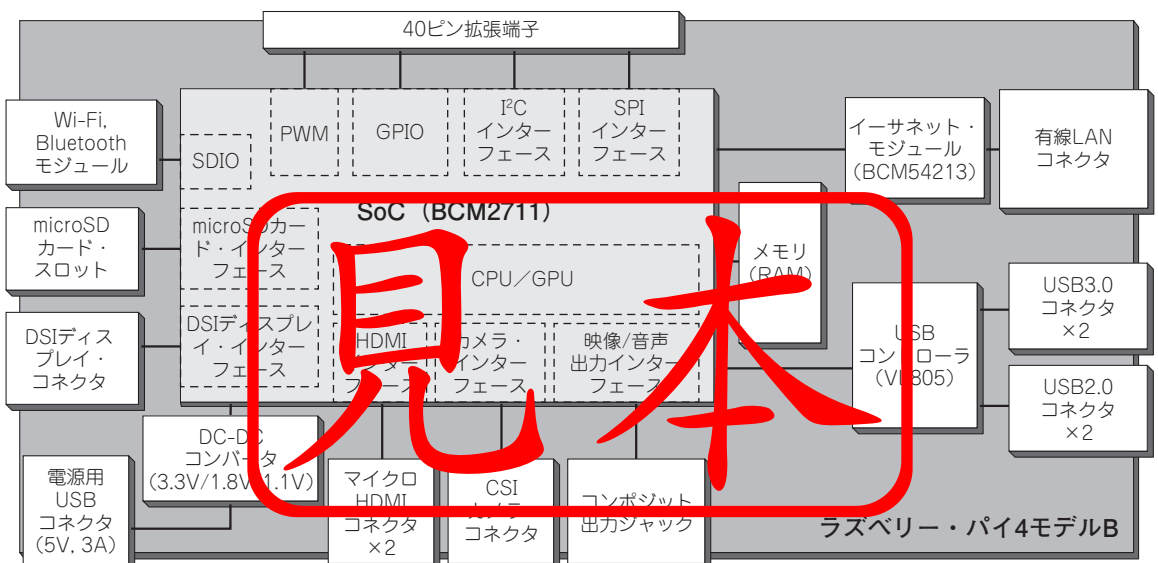
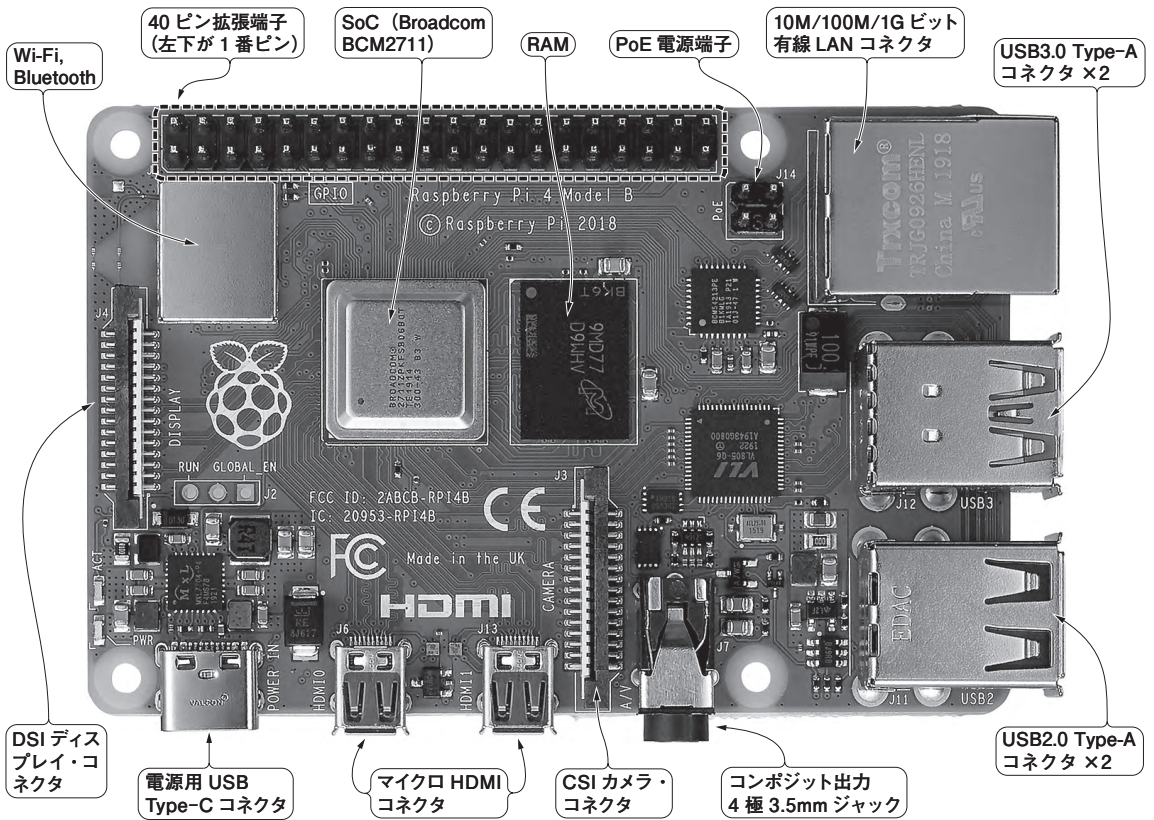
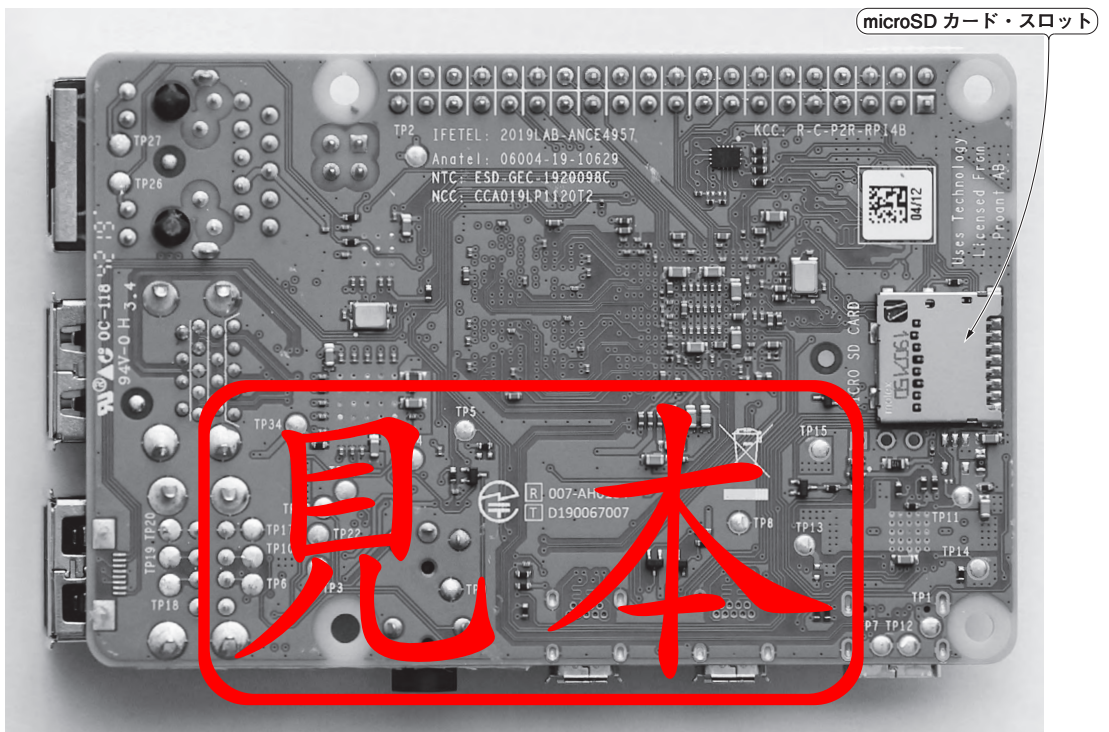


図1 ラズベリー・パイモデルBのハードウェア構成

第1章 ラズベリー・パイのハードウェア構成



(a) 表面



(b) 裏面

写真1 ラズベリー・パイ4モデルBの外観

ラズパイの世界

ハード&ソフト

制御の基本

よく使う/O

カメラ&ネット

実用的に動かす

ラズベリー・パイのソフトウェア環境

永原 柊 Shu Nagahara

さっそくラズパイを使える状態にしていきましょう。専用のツールが用意されているので、詳しい知識がなくても設定できるようになっています。

ラズパイを動かすには、まず、起動用 microSD カードを作成します。作成した microSD カードをラズパイのスロットに差し込んで電源を ON にするとラズパイが起動するので、各種設定を行います。

ラズパイが出始めたころは起動用の microSD カードを準備するだけで一苦労でしたが、今では専用ソフトウェアで簡単にできるようになりました。

起動用 microSD カードを作成する方法はいくつか用意されていますが、本章では最も基本的なやりかたを説明します。

1 起動用 microSD カードの作成

起動用 microSD カードの作成は、「Raspberry Pi Imager」というツールを用いて行います。このツールを使って microSD カードにラズパイ用の OS (Raspberry Pi OS) をインストールします。

私が使用した microSD カードを写真1に示します。容量 32 G バイト、Class 10 のカードがお勧めです。

起動用 microSD カード作成時に使用する機材を図1に示します。筆者はパソコンに Windows 機を使用しました。メモリ・カード・リーダ/ライタは、パソコン内蔵のものを使用し、microSD-SD カード変換アダプタを介して接続しました。

パソコンに Raspberry Pi Imager をインストールする

● ダウンロード

ツール (Raspberry Pi Imager) をダウンロードから始めます。ラズパイの公式 Web サイト (<https://www.raspberrypi.com/>) を開いて、メニューの中にある「Software」をクリックします(図2)注1。

注1：以前は、Raspberry Pi 財団の Web サイト <https://www.raspberrypi.org/> がラズパイの公式ページとなっており、[Computers]-[Software] とメニューをたどる形になっていた。今後も Web サイトの画面やメニューなどは変更になる可能性があるがあるので、適宜読み替えていただきたい。



写真1 使用した microSD カード

図3に示す、Raspberry Pi OS の Web ページが表示されます。このページを下にスクロールすると、Raspberry Pi Imager のインストーラをダウンロードするリンクがあります。

自分の環境に合わせて、ツールのインストーラをダウンロードします。Windows の場合は [Download for Windows]、macOS の場合は [Download for macOS] を選びます。

● インストール

ダウンロードしたツールをパソコンにインストールして、使えるようにします。

インストールの手順は、画面で指示されるとおりに行います。その後の操作手順や画面の見た目は Window でも macOS でも同じです。以降、Windows 版の例で説明します。

ダウンロードしたインストーラを起動すると、ツ

コマンド操作による ピン出力の制御

永原 柊 Shu Nagahara

まずは理解を深めるために、外付けしたLEDの点滅から始めます。

ラズパイでハードウェアを操作する方法は、マイコンのプログラムによる操作とは全く異なっています。

ラズパイの拡張端子(GPIOピン)にLEDを接続して、コマンド・ラインからの操作によって対話的に点滅させてみます。

1 外付けLEDを制御するためのハードウェア構成

ラズパイに用意された40ピン拡張端子にある多数のGPIOのピンの中から、ここでは16番ピンにあるGPIO23を利用します。図1のようにGPIOにLEDと1kΩの電流制限用抵抗をつなぎます。GNDは14番ピンを使います。

ブレッドボードを使った接続のようすを写真1に示します。後章で部品を追加するので、大きいブレッドボードを使用しました。

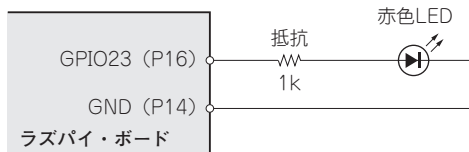


図1 外付けLEDを使った回路図

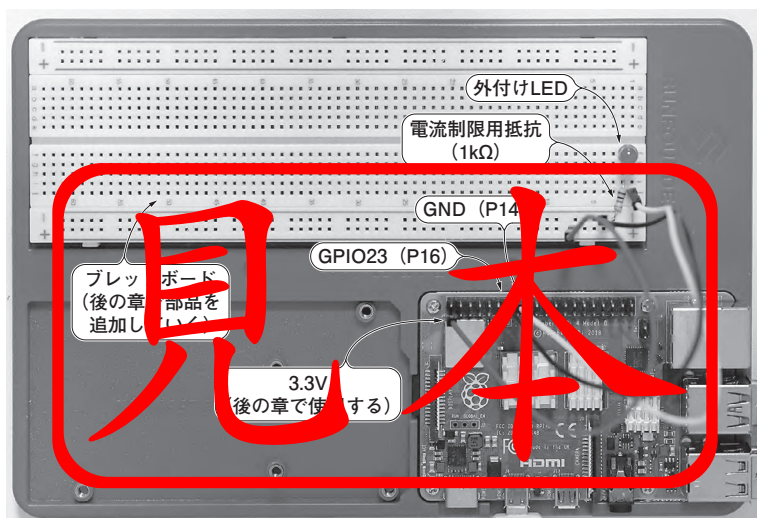


写真1 外付けLEDを使った実験のようす
後の章で部品を追加するので、大きいブレッドボードを使用した

シェル・スクリプトによる ピン出力の制御

永原 柁 Shu Nagahara

ラズパイでは、コマンド・ラインからの操作を、ほぼそのままプログラムにすることができます。ラズパ

イの標準OSであるRaspberry Pi OSには、そのためのしくみが用意されています。

1 シェル・スクリプトで作るプログラムのしくみ

ラズパイでは、キーボードからの入力はシェルというプログラムが受け取って処理しています。つまり、キーボードがシェルの入力とながっているイメージです。

もし図1のように、シェルの入力をファイルに置き替えることができれば、シェルはそのファイルの内容を読み取って、キーボードから入力されたかのように処理することができそうです。そのファイルに、キーボード入力するのと同じ内容を書いておけば、シェルにはキーボード入力したときと同じ入力があり、同じ動きができます。ということは、このファイルはあ

る種のプログラムと言えそうです。

ラズパイは、このようなしくみを備えています。シェルへの入力になるファイルは、シェル・スクリプトと呼ばれます。プログラミング言語を知らなくても、コマンド・ラインから操作する内容をシェル・スクリプトにすることで、そのままプログラムになります。

本章では「GPIOの事前準備をするシェル・スクリプト」、「後始末をするシェル・スクリプト」、「LEDを点灯するシェル・スクリプト」、「LEDを消灯するシェル・スクリプト」の4つを作ります。

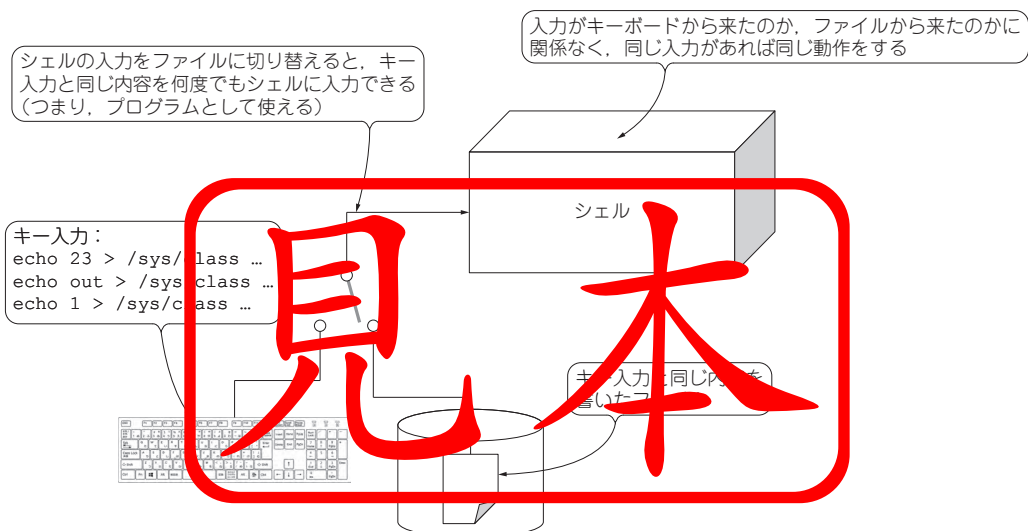


図1 シェルへの入力を切り替える
キー入力内容をファイルに記録しておけば、プログラムとして何度でも実行できる

第5章 外付けスイッチが押されているかをプログラムが知るには

ピン入力状態の読みとり

永原 柊 Shu Nagahara

GPIOからの出力ができるようになったので、次にGPIOの入力を試します。GPIOを介して外付けしたスイッチのON/OFFを検出します。

なお、念のために最初に書いておくと、ラズパイの

デジタル入出力ピンは3.3V入出力です。5Vトレラント(耐電圧)ではないので、5Vを加えると破壊する恐れがあります。

1 外付けスイッチで入力するためのハードウェア構成

ラズパイのGPIOは、チップ内部でプルアップ/プルダウンする機能が用意されています。

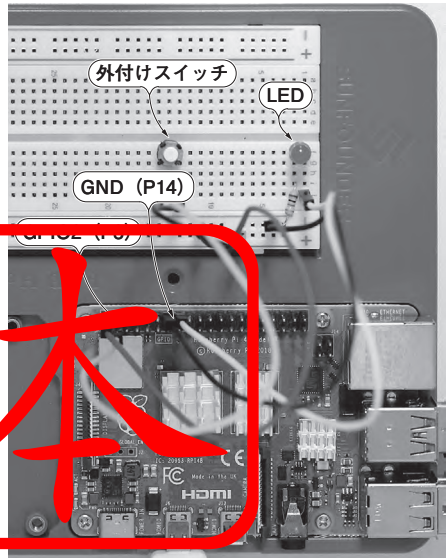
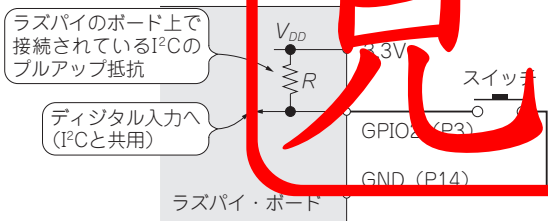
初期状態では内部的にプルダウンされているピンが多いようですが、それを明示したドキュメントを見つけれませんでした。また、シェル・スクリプトでプルアップ/プルダウン機能を操作する方法を見つけられませんでした。

そこで、ボード上でプルアップされているGPIO2を使うことにします。GPIO2とGPIO3はI²Cというインターフェース(第12章を参照)でも使用するピンです。おそらくI²C用のプルアップ抵抗なのだと思います。

回路を図1に、接続のようすを写真1に示します。GPIO2とGNDの間にスイッチを入れます。後で使うので、第3章と第4章で使用したLEDは残してあります。

初期状態ではI²Cは無効になっています。I²Cの有

効/無効は図2の設定画面で設定できます。画面左上のラズパイ・マークから、[設定]-[Raspberry Piの設定]-[設定]の順に選んで、設定画面を表示します。もしI²Cが有効になっていたら、このスイッチ入力の実験中は一時的に無効にしてください。後の章で、別のプログラミング言語を用いて内部プルアップを操作する例を示します。



見本

図1 外付けスイッチを使った回路図
P3はI²Cと共用しており、ここではI²Cのプルアップ抵抗を使う

写真1 外付けスイッチを使った実験のようす
拡張端子のP3にあるI²Cのプルアップ抵抗を使うため、スイッチをP3に接続している

第6章 なぜプログラムから電気信号を任意にI/Oできるのだろう

GPIO制御におけるラズパイ内部の動作

永原 柊 Shu Nagahara

ラズパイでは、ファイルを読み書きするだけでGPIOの入出力方向を変更できたり、GPIOを読み取れたりします。この方法は、人間から見てわかりやすいだけでなく、Raspberry Pi OSの基本的な考え方でもあります。この方法を使うことにより、さまざまな手段でGPIOを操作できます。

ここでは、GPIOを制御する際のラズパイ内部のしくみを説明します。まずは①～⑤で、GPIO出力でLEDの点滅を行ったときのしくみを説明します。続いて⑥～⑦で、外付けスイッチでGPIO入力を行ったときのしくみを説明します。

① ラズパイ内部にあるGPIOまわりの構成

GPIOまわりについて、ラズパイ内部のイメージとハードウェアを図1に示します。

関連するハードウェアとして、ラズパイ・ボード上のGPIOと、外付けLEDがあります。GPIOは、SoC内にあるデジタル信号の入出力口(入出力端子)です。それを制御するソフトウェアとして、GPIOドライバ

があります。

GPIOドライバに指示する方法として、`/sys/class/gpio`ディレクトリにある`export`ファイルなどのような実体のないファイルが用意されています。このようなファイルを読み書きすることで、GPIOを操作します。

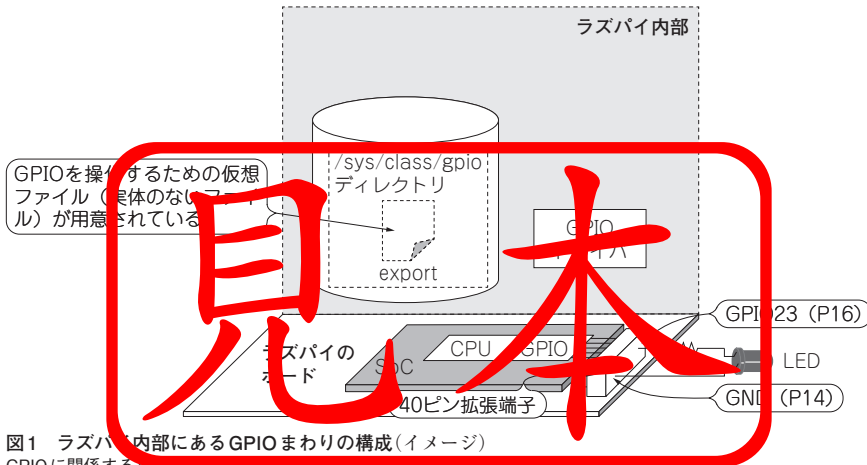
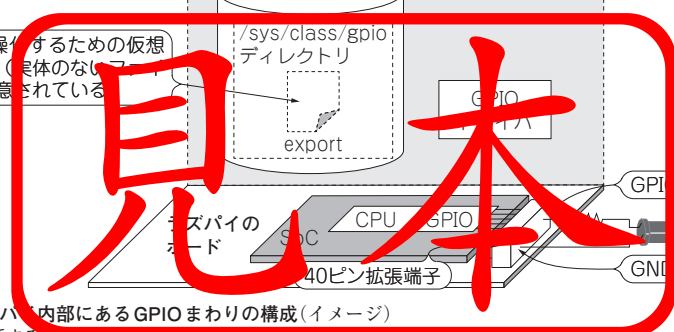


図1 ラズパイ内部にあるGPIOまわりの構成(イメージ)
GPIOに関係するハードウェアとソフトウェアを示す



Python プログラムによるI/O制御

永原 柊 Shu Nagahara

ここまで、ずっとシェル・スクリプトを使ってきました。しかしプログラミング言語でも、同様のプログラムを作成できます。また、異なるプログラミング言

語で書かれたプログラムを組み合わせることが可能です。ここでは第5章や第7章で作ったプログラムをPythonで書き直してみます。

1 ラズパイ内蔵のプルアップ抵抗を使ったスイッチ入力回路

前章までのスイッチ入力にはラズパイのGPIO2を使ってきましたが、このピンはI²Cというインターフェース(第12章参照)を使う際に用います。そこで他のGPIOピンを使うことにします。

ラズパイはGPIOにプルアップ抵抗もプルダウン抵抗も内蔵していますが、シェル・スクリプトではそれを有効にする方法がわかりませんでした。一方Pythonでは、プルアップ抵抗、プルダウン抵抗を有効にできます。そこでLEDに使っているGPIO23の隣にある、GPIO22のプルアップ抵抗を有効にしてスイッチをつなぎます。

作成する回路を図1に、実験のようすを写真1に示します。

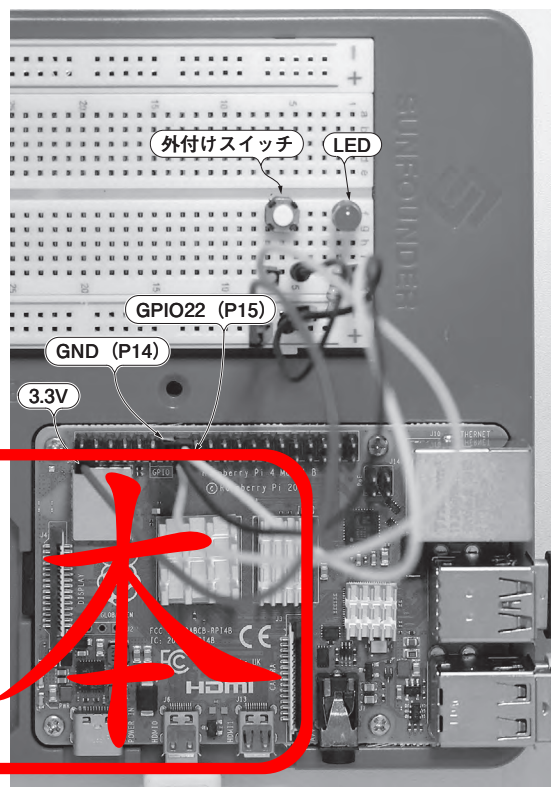
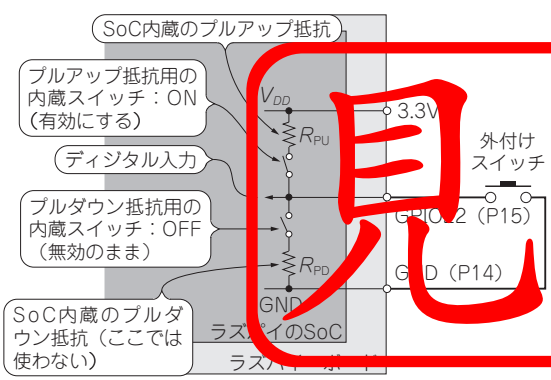


図1 外付けスイッチを使った回路図
GPIO22(P15)に外付けスイッチを接続し、ラズパイのSoCに内蔵されたプルアップ抵抗をPythonプログラムから有効にする

写真1 外付けスイッチを使った実験のようす
ラズパイのSoC内蔵プルアップ抵抗を使うので、使用できる拡張端子の幅が広い(ここではP22を使用)

異なる言語のプログラムも組み合わせる

永原 柊 Shu Nagahara

これまでの章では、シェル・スクリプトとPythonで書いたプログラムを作り、それぞれ個別に動作することを確認してきました。また、シェル・スクリプトどうしなら、スイッチ入力プログラムとLED出力プ

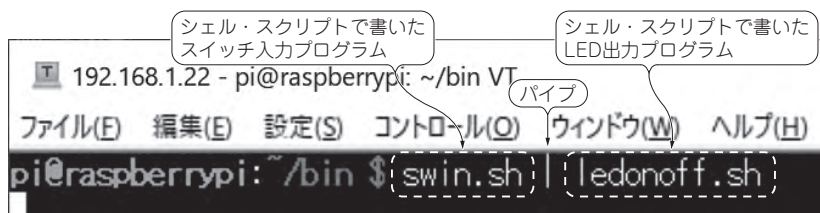
ログラムを、パイプでつなぐことで組み合わせる動作ができました [図1(a)]。

本章では、Pythonで書いたプログラムとシェル・スクリプトを組み合わせる使ってみます。

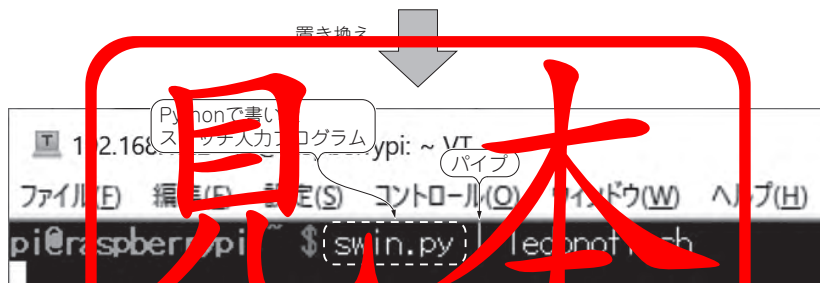
1 Pythonで書いたプログラムとシェル・スクリプトを組み合わせる

Pythonで書いたスイッチ入力プログラム(第8章のリスト1)と、シェル・スクリプトで書いたLED出力プログラム(第7章のリスト1)を、図1(b)のように組み合わせる使ってみます。

なお、第8章のリスト1は、もともとシェル・スクリプトで書いたプログラム(第5章のリスト1)をPythonで書いたプログラムに置き換えたものです。



(a) シェル・スクリプトで書いた2つのプログラムを組み合わせる



(b) Pythonで書いたプログラムとシェル・スクリプトで書いたプログラムを組み合わせる

図1 異なる言語で書いたプログラムを組み合わせる使おう
シェル・スクリプトで書いたスイッチ入力プログラムをPythonで書いたスイッチ入力プログラムに置き換えても、シェル・スクリプトで書いたLED出力プログラムと組み合わせると正しく動作する

GPIO 割り込み制御

永原 柊 Shu Nagahara

第8章で使用したラズパイ用のGPIOライブラリRpi.GPIOでは、GPIOの変化でソフトウェア的な割り込みを発生させることができます(本書では、これ

をGPIO割り込みと呼ぶ)。ここでは、スイッチを押すとソフトウェア的な割り込みハンドラが呼び出されるプログラムを作ってみます。

1 GPIOの入力が変化したときに割り込みを発生させる機能がある

ラズパイが時間がかかる処理を継続的に行っている途中で、ユーザがスイッチを押したら、実行中の処理に割り込んで何か特別な処理を行いたい、という場合を考えます。図1では、スイッチを押すと画面に1を表示する(標準出力に1を出力する)例です。

単純に実現しようとすると、ラズパイが全力で実行しているプログラムの途中で、スイッチが繋がっているGPIOを監視する処理を入れる必要があります。しかし、このやり方はプログラムのどこでスイッチを

監視すればよいかなど考えることが多く、汎用性はありません。

図2に示すように、スイッチが繋がっているGPIOへの入力に変化したときに、割り込みを発生できれば解決できそうです。

このように、GPIOの入力が変化したときに割り込みを発生させる機能を、ここではGPIO割り込みと呼びます。

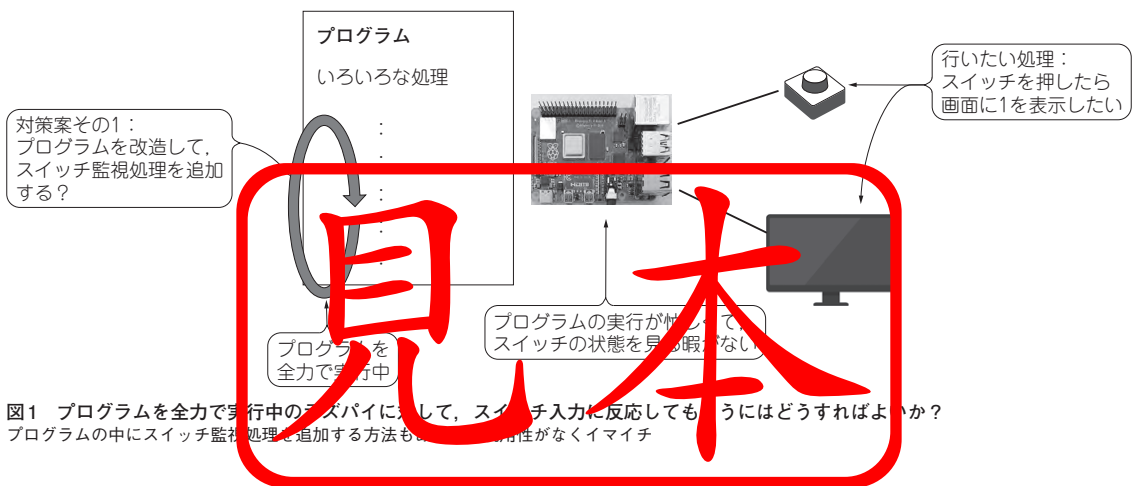


図1 プログラムを全力で実行中のラズパイに対して、スイッチ入力に反応してもプログラムの中にはどうすればよいか? プログラムの中にスイッチ監視処理を追加する方法も汎用性がなくイマイチ

第12章 液晶ディスプレイもセンサもサツと使えるバスのしくみ

定番 I²C 通信の制御

永原 柊 Shu Nagahara

ラズパイを使いこなすためにいろいろな機器を外付けしようとする時、どのように接続するかを考える必要があります。

ここまで使ってきたGPIOではON/OFF信号しか送れないので、多機能な機器をつなぐには多数のGPIO端子が必要になり、現実的ではありません。例えば3軸加速度センサを考えると、各軸の加速度を8ビットで表した場合、8ビット×3軸で24本のGPIO

が必要になります。ここでは、加速度センサを付けただけでラズパイのGPIO端子を使い切りそうです。

そこで、1本の電線で多数の信号をやりとりできる通信が必要になります。I²C通信はそのような通信規格の1つです。ほかにもSPI通信やシリアル通信などがあります。

本章では、I²C通信を使って液晶表示器に任意の文字列を表示します。

1 I²C通信のしくみ

● I²C接続のしかた

図1にI²C通信の接続例を示します。I²Cは2本の通信線(SCLとSDA)でバス型接続します。基本的に、SCLはクロックを、SDAはデータをやりとりする通信線です。

それぞれの通信線はワイヤードOR(複数の出力を並列に接続した論理和)でつながっており、プルアップ抵抗が必要になります。ラズパイの場合、ラズパイのボード上に実装されています。

このI²C通信バスに、ラズパイやセンサや液晶表示器など、さまざまな機器を接続します。

基本的に、1つがマスタで、残りの機器はスレーブと呼ばれます。図1ではラズパイがマスタで、残りのセンサや液晶表示器などがスレーブになります。仕様上はマスタが複数ある場合もあり得ます。

● I²C機器はアドレスで区別する

図1のように接続すると、マスタが送信した信号がすべての機器に届いてしまいます。マスタがどの機器と通信したいかを示すために、I²Cアドレスがあります。

I²Cアドレスは、個々の機器に割り当てられた値です。例えば、後で出てくる液晶表示器の場合、I²Cア

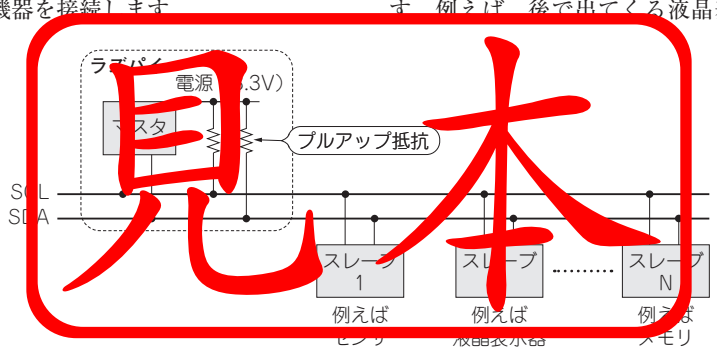


図1 I²Cの接続形態
複数のスレーブ(機器)であっても2本の通信線があればよい。マスタ/スレーブの呼び方は、コントローラ/ターゲットとも呼ばれる

複数プログラムを組み合わせるしくみ「FIFO」

永原 柊 Shu Nagahara

第12章では時刻を、第13章では温度を液晶表示器に表示しました。

使用している液晶表示器は2行表示できるので、1行目には時刻を、2行目には温度を表示したくなります。液晶表示プログラム(第12章のリスト1)は標準入

力から読み取りますが、標準入力1つしかないので、同時に読み取れるプログラムは1つだけです。

本章では、複数のプログラムの出力を1つにまとめるために、^{フアイフオ}FIFO(First In, First Out)と呼ばれるしくみを利用します。

1 パイプでは2つの出力を1つの入力につなげられない

ここで取り上げている問題を図1に示します。これまで使ってきたパイプでは1つの出力と1つの入力を接続できました。しかし、2つの出力を1つの

入力につなぐには、今まで使ってきたパイプでは難しくそうです。

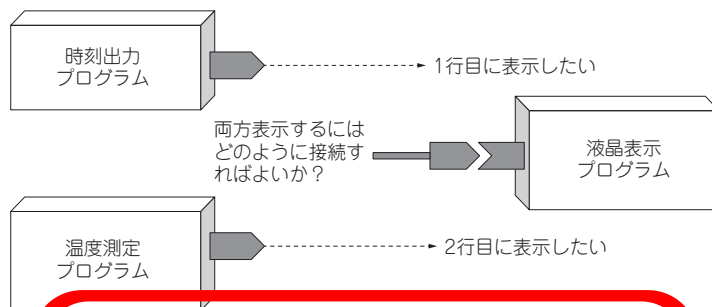


図 液晶表示器に2つのプログラムの出力を表示したい場合はどうすればよい?



カメラ制御&画像処理入門

永原 柊 Shu Nagahara

ラズパイでは、カメラのような大容量データを処理する機器を取り扱えます。ここでは、ラズパイの公式カメラ・モジュールを使います。

まずスイッチを押したときに撮影するプログラムを作り、次にセンサ値によって撮影するプログラムに取り組みます。

1 カメラ撮影に使用した実験ボード

ここで使用するカメラは、ラズパイ公式から出ている(Raspberry Pi財団が認可している)Raspberry Pi Camera Module V2.1(以降、カメラ・モジュール)です。ラズパイ状のカメラ・コネクタに接続して使用し

ます。

実験のようすを写真1に示します。LEDと液晶表示器は使いません。また、カメラ・モジュールを写真2に示します。

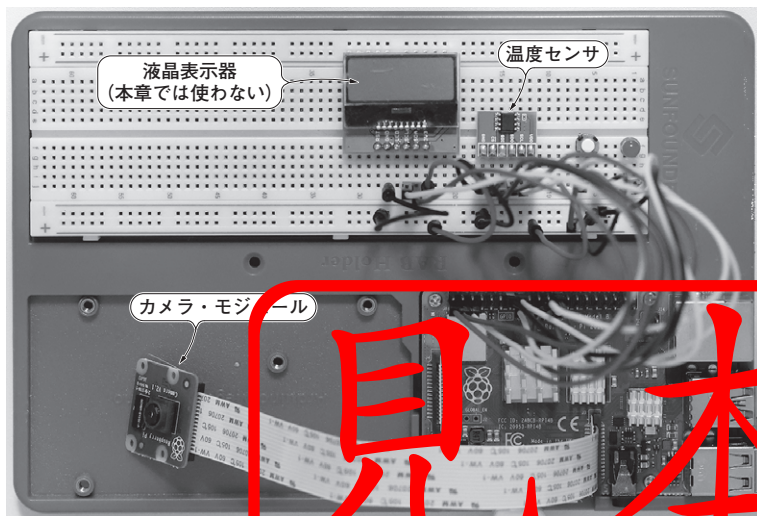
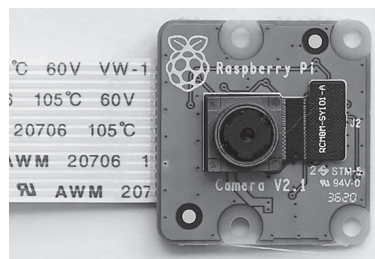
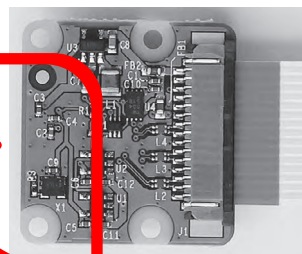


写真1 実験のようす
今回はカメラ・モジュールと温度センサを使う



(a) 表面



(b) 裏面

写真2 使用するカメラ・モジュール
ラズパイ専用のカメラ・モジュールを使用した

ラズパイ Web サーバ入門

永原 柊 Shu Nagahara

前章ではクラウドに対してラズパイがクライアントとして動いていました。本章では、ラズパイを簡易なWebサーバとして動かしてみます。

パソコンやスマートフォンなどのクライアントからアクセスして操作することにより、Webサーバであ

るラズパイにつながるLEDを点灯/消灯させてみます。

本章で行う実験は基本的なやり方を確認するものです。これができるば、ネットワーク上のクライアントからリクエストを受けて、さまざまな動作をさせることができるはずで

1 Webのしくみ

図1に、クライアントからWebサービスを利用する際の大まかな流れを示します。

- (1) クライアントのWebブラウザにアクセス先となるURLが入力されると、クライアントはWebサーバにアクセスしてリクエストを送信する
- (2) Webサーバは、あらかじめ用意してある静的なデータや、プログラムにより生成する動的なデータからレスポンスを返す
- (3) クライアントは、返ってきたレスポンスをもとに整形して表示する

今回の実験では、ラズパイ上でWebサーバを動かします(図2)。静的なデータはmicroSDカードに格納しておき、動的なデータはPythonプログラムで生成します。また、そのプログラムでLEDの点灯/消灯も同時に行います。

使用する通信プロトコルは図3のようになっています。必要なプロトコル・スタックはラズパイに標準で用意されています。

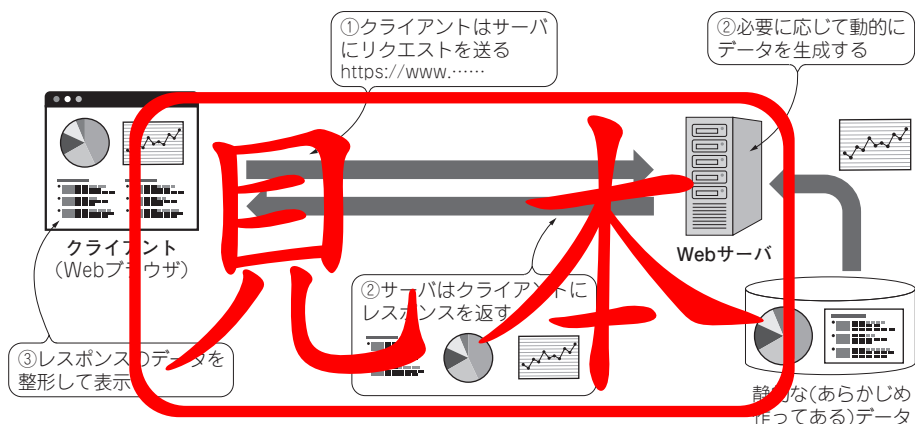


図1 Webサーバとクライアントのやりとりの流れ
クライアントであるWebブラウザからのリクエストに応じて、Webサーバは静的&動的なデータからなるレスポンスを返す。Webブラウザはそのレスポンスを整形して表示する

その①… バックグラウンド実行

永原 柊 Shu Nagahara

ここまで、いろいろなプログラムを作っては、コマンド・ラインから起動してきました。

特に、第14章の図7(図1として再掲)に示した例では、4つのターミナルを開いて8個のプログラムを実行していました。

この例では、スイッチ入力と温度センサの測定値をもとに、LEDへのPWM出力と液晶表示器への表示を

行っていて、キー入力が必要でした。

このようにキー入力が必要な場合は、プログラムをバックグラウンドで実行することもできます。図1の例をバックグラウンドで実行すると、1つのターミナルで実行できます。

本章では、プログラムをバックグラウンドで実行する方法について説明します。

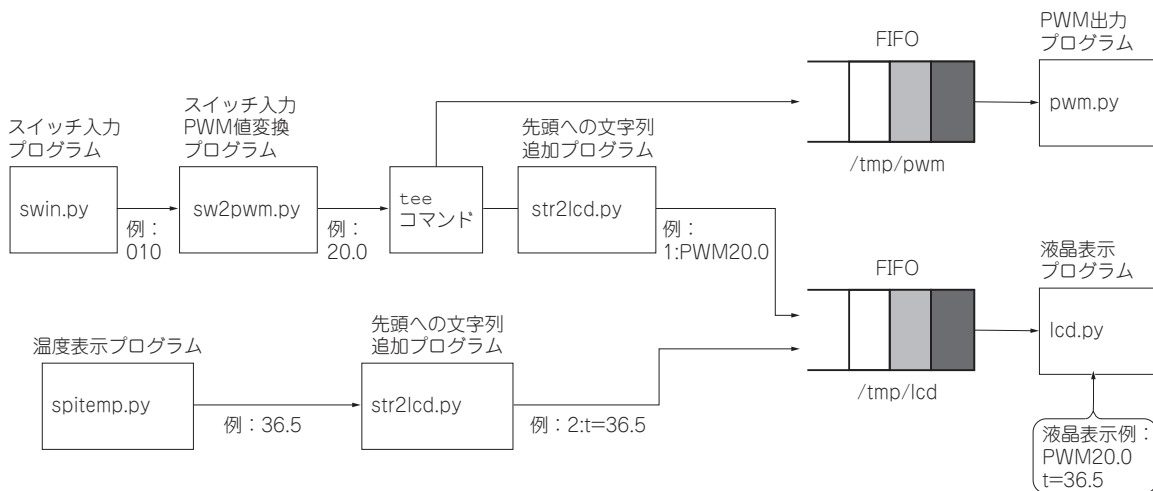
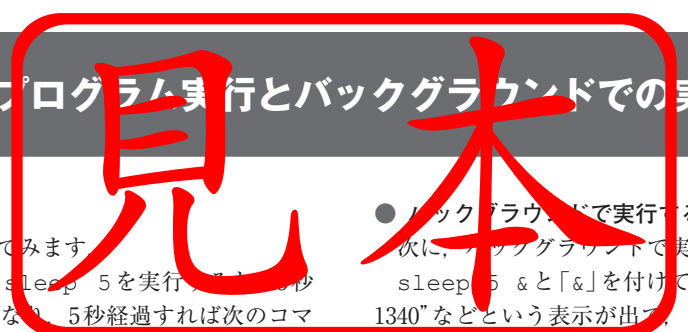


図1 8個のプログラムを同時に実行する例(第14章の図7を再掲)

Pythonプログラムとteeコマンド、合計8個のプログラムを実行する。第14章では、4つのターミナルを使って4つのコマンド・ラインを実行していた

① 通常のプログラム実行とバックグラウンドでの実行の違い



● 普通に実行すると

まず、普通に実行してみます。コマンド・ラインで `sleep 5` を実行すると、5秒間キー入力ができなくなり、5秒経過すれば次のコマンドを受け付けられる状態になります(図2)。

また、`sleep 5` を実行中の5秒間に `[Ctrl]+C` を入力すると、`sleep` コマンドの実行を中断できます。

● バックグラウンドで実行すると

次に、バックグラウンドで実行します。`sleep 5 & と「&」を付けて実行します。すると「[1] 1340” などという表示が出て、すぐに次のコマンドを入力できる状態になります [図3(a)]。`

表示されているものは、ジョブ番号とプロセス番号と呼ばれるものです。かっこ内の数字はジョブ番号(こ

第22章 役割分担すれば死角なし! UART通信から

ラズパイの良き相棒 Pico マイコン入門

永原 柊 Shu Nagahara

前章で説明したように、ラズパイはタイミング制御において精度の高い処理が得意ではありません。一方、シングルタスク動作のマイコンであれば、Arduino Unoに搭載されている8ビット・マイコンでさえ、ラズパイを上回る精度を出すことができます。

本章では、ラズパイにマイコンを外付けして連携動作させ、ラズパイが得意な処理はラズパイで、マイコンが得意な処理はマイコンで行わせてみます。

どのようなマイコンでも使えますが、ここでは

Raspberry Pi財団が開発したRP2040マイコンを使います。

ラズパイとマイコンの間は、これまで使ってこなかったシリアル通信で接続します。またマイコン側はフルスペックのPythonが使えないので、マイコン用に軽量化したMicroPythonを使ってプログラムを作成します。ラズパイ側はコマンド・ラインから操作します。

1 ラズパイとマイコンを組み合わせる

図1に示すように、ラズパイとマイコン・ボードを組み合わせ動かしてみます。ここでは、RP2040マイコンを搭載したRaspberry Pi Pico(以降、Pico)を組み合わせます。

実現するのは、前章で見た1msのパルスを出力する処理です。これを、ラズパイとマイコンの連携で実現します。

といっても、マイコンだけでパルス出力を実現できます。そこで、ここではラズパイからパルス幅を指定すると、マイコンが指定された幅のパルスを出力することに挑戦します。センサから読み取った値から計算したり、ネットワーク経由で指がされたり、といったことによりラズパイがパルス幅を決めると決めること

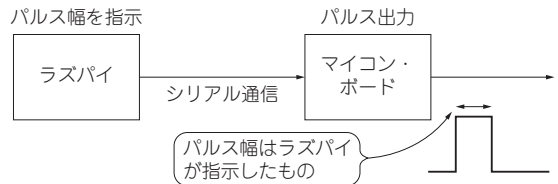
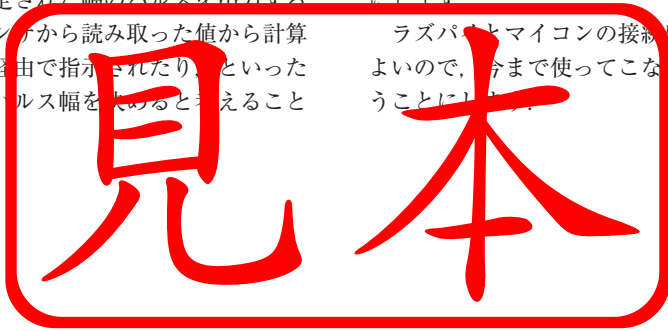


図1 ラズパイとマイコン・ボードの役割分担
ラズパイがシリアル通信経由でパルス幅をマイコン(マイコン・ボード)に伝え、マイコンは指定された幅のパルスを出力する

にします。ラズパイとマイコンの接続は、通信できれば何でもよいので、今まで使ってこなかったシリアル通信を使うことにします。



アナログ信号を ラズパイに取り込む

永原 柊 Shu Nagahara

外付けマイコンとの連携で、パルス出力以外の活用方法を考えてみます。

ラズパイにはユーザが利用できるA-D変換機能がありません。そこで、マイコン内蔵のA-D変換機能を利用して、その変換結果をラズパイで受け取ることを考えます。

ここではマイコンからラズパイにデータを渡すこと

になります。

マイコンとしてRaspberry Pi Pico(以降Pico)を使います。一般的な機能だけを使っているため、ほかのマイコンでも同様に実現できるはずですが。

ラズパイとPicoの接続は、前章と同様にシリアル通信を使います。また、A-D変換の簡単な回路を追加します。

1 ラズパイとマイコンをシリアル通信でつなぐ

● Raspberry Pi Pico側の回路

図1のように、シリアル通信(UART)の接続に加えて、31番ピンのA-D変換端子(ADC0)に半固定抵抗をつなぎました。Picoの36番ピンの3.3V出力と、33番ピンのAGNDを電源として使っています。実験のようすを写真1に示します。

● ラズパイからA-D変換開始の指示を出す

ここではラズパイからA-D変換開始の指示を出します。具体的には「adc」という文字列を送ります。

Picoが指示を受けると、A-D変換を実行して結果をラズパイに返します。ラズパイから1回指示を出すと、Picoが1回結果を返すやりとりになります。

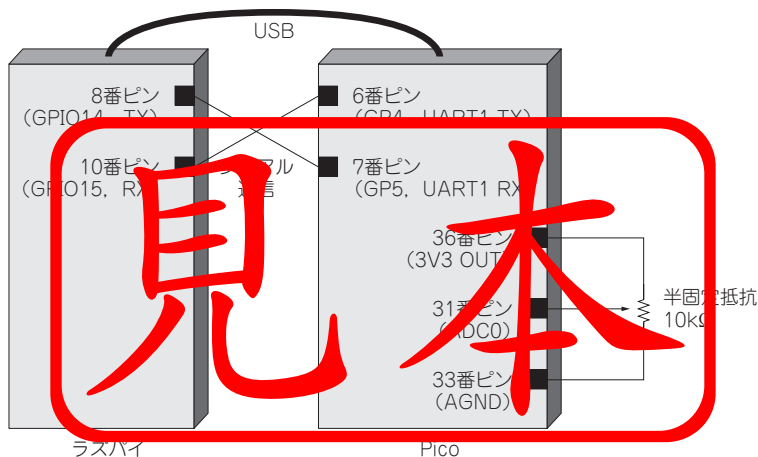


図1 ラズパイとPicoの接続
PicoのA-D変換端子に半固定抵抗をつなぐ

このPDFは、CQ出版社発売の「トランジスタ技術SPECIAL No.163」の一部見本です。
内容・購入方法などにつきましては以下のホームページをご覧ください。

内容 <http://shop.cqpub.co.jp/hanbai/books/MSP/MSP202307.html>
購入方法 <https://www.cqpub.co.jp/order.htm>