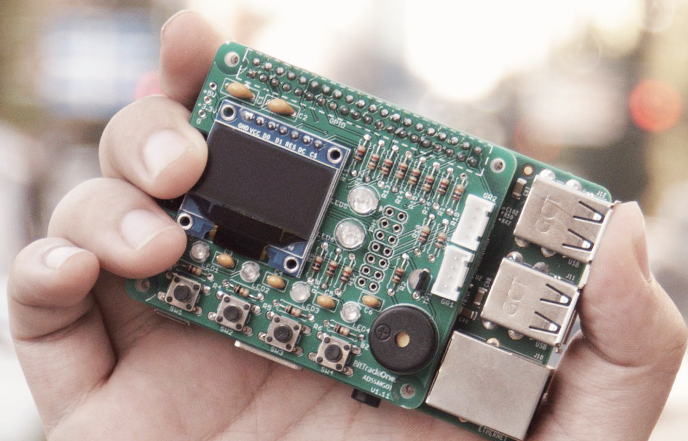


ラズパイではじめる 電子制御とIoT

TAKE
FREE



AssemblyDesk
Books

特集

ラズパイで
IoTをはじめよう!

Node-REDとMQTT

OLEDと光センサーを
使ってみよう

ラズパイ入門ボード応用編

ラズパイではじめる 電子制御とIoT

Assembly Desk Books volume02

■ 本誌について

Assembly Desk Books はビット・トレード・ワンが発行するエンジニアやエンジニアを目指す方々に向けたフリーペーパーです。シェルスクリプトマガジンとの連動企画による「ラズパイ入門ボード」の解説や昨今話題のIoTを手軽に試すことができる情報が満載です。

Assembly Desk Books Vol.2 では特集記事として、このラズパイ入門ボードに関して OLED や光センサーの使い方の一部を紹介するほか、Node-RED と MQTT を利用した IoT 入門をご紹介します。

ラズパイ入門ボードに関するソースコードについては以下ページにて公開されております。本書とあわせてご活用ください。



<https://shell-mag.com/raspi> original code/



INDEX

03 ~ 07page

特集1: ラズパイでIoTをはじめよう!

- ・ Node-RED と MQTT

09 ~ 13page

特集2: OLEDと光センサーを使ってみよう

・ラズパイ入門ボード応用編 1

ラズパイ入門ボードの入手方法

ラズパイ入門ボードは、ビット・トレード・ワン
(<http://bit-trade-one.co.jp/>)とシェルスクリプトマガジン編集部が共同
で制作したラズパイ用ハードウェアです。

完成品と組み立てキットの2種類を用意します。いずれもビット・ト
レード・ワンの公式オンラインショップ(<http://btoshop.jp/>)、シェルス
クリプトマガジン発行元であるユニバーサル・シェル・スクリプト研究所
のサイト <https://www.usp4ab.com/ORDER/CGI/PUB.ORDER.CGI>、
Amazon.co.jp、共立エレクトロニクス(<http://eleshop.jp/>)および、シリコン
ウズ、千石電商、デジット、マルツエレクトロニクスなどの各店舗から購入できます 価
格(税別)は、完成品が3980円、組み立てキットが2780円です。(送料別)



ラズパイではじめる 電子制御とIoT

■ はじめに

世間ではさかんにこれからはIoTだ！という言葉が飛び交っています
が、実際に自分でIoTを体験されている方はどれくらいいるのでしょうか？IoTを簡単に説明すると、「身の周りのあらゆるモノがインターネットにつながる」仕組みのことです。

あらゆるモノがつながると聞くと難しい感じもありますが、もっと簡単に我々がIoTを扱えるような方法はないもののでしょうか？

本特集は、これからラズパイを使ってIoTを初めてみたいというユーザーを対象に、ビット・トレード・ワンから発売されているpHATサイズの拡張基板群「ゼロワン」シリーズを利用して、モノをインターネットにつなげるIoTを簡単に実現してみます。「ゼロワン」シリーズはサーボを動かしたり、各センサをひとつだけ搭載するといったように単機能に特化したラズベリーパイ用の拡張基板です。

ゼロワンシリーズはラズベリーパイゼロにぴったりのサイズでIoTデバイスを作成するには最適な拡張基板です。

今後、“Node-RED”システムと“MQTT”プロトコルを利用し、Raspberry Pi 間の通信をインターネット経由で行ってゼロワンシリーズを連携させてみます。

今回は、Raspberry Pi Zero の準備からはじめ、いちから丁寧に解説していきます。その前にまずは Node-RED と MQTT のお話をさせてください。

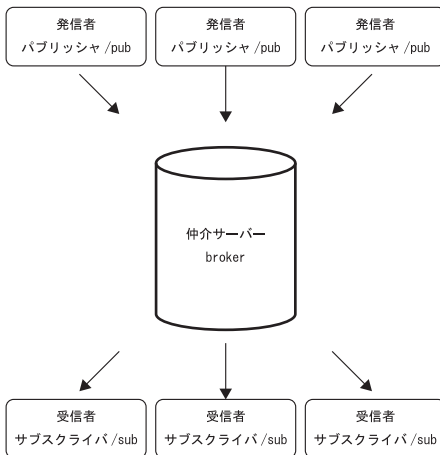
■ Node-RED/MQTTとは

Node-RED は、画面上に“ノード”と呼ばれる処理ブロックを配置し、ノードの間を結線しデータの流れ（フロー）を作成することで、簡単にデータ処理・マシン制御システムを構築できる開発環境です。

Raspberry Pi 用の OS として一般的な“Raspbian”にはプリインストールされており、ネットワーク経由で手元のパソコンからフローの作成、Raspberry Pi と周辺機器の制御が可能です。

■ MQTTとは

MQTT とは、発信機器（パブリッシャ/Pub）と受信機器（サブスクライバ/Sub）、その間でデータを仲介するサーバー（ブローカ/broker）から成り立つ、インターネット経由で遠隔地間の情報をやり取りする通信プロトコルです。



特性、成り立ちから、主にセンサの情報を収集し、シンプルな機械を制御するようなIoTシステムの通信プロトコルとして広く利用されています。

Raspberry Pi Zeroの準備

Raspberry Pi Zero シリーズ (v1.3,W,WH) は、現行の Raspberry Pi 3 と搭載されている SoC のコアのバージョンが異なります。

(Zero : ARMv6 / 3 : ARMv8)

そのため、Pi 3 で作成した環境をそのまま Pi Zero に移し替えると、ソフトウェアが正常に動作しない可能性が高いです。Raspbian のインストールや、後述の Node-RED の準備などは必ず Raspberry Pi Zero 上で行ってください。

また、Raspberry Pi Zero は micro USB 端子と mini HDMI 端子が装備されています。通常規格の USB、HDMI は非対応となるので、あらかじめ miniHDMI-MDMMI 変換アダプタ、USB-OTG ケーブル、USB ハブを準備してください。

Raspbianのインストール

最初に、Raspberry Pi 公式ダウンロードページ

(<https://www.raspberrypi.org/downloads/noobs/>)より "NOOBS" をダウンロードします。NOOBS を用いることで Raspbian のインストールを簡単に行うことができます。

NOOBS のサイズは 1.3GB と大きく、ダウンロードに時間がかかりますので、その間に SD カードをフォーマットします。

この時、Windows 標準のフォーマットツールではなく SD アソシエーションの提供する SD カードフォーマッター (https://www.sdcard.org/jp/downloads/formatter_4/) を使用しましょう。ダウンロードし、PC にインストールしてください。

フォーマットオプションは、念のため上書きフォーマットを選択しましょう。

用意する SD カードの容量は 8GB をお勧めします。4GB ではシステムが入りきらないことがあり、16GB 以上はフォーマットに時間がかかったり、場合によっては相性により Raspberry Pi で使用できない可能性があります。

NOOBS のダウンロード、SD カードのフォーマットが終わったら、ダウンロードした NOOBS の Zip の中身を全て SD カードにコピーします。これもしばらく時間が掛かります。

コピーが終わったら、Raspberry Pi Zero に SD カード、miniHDMI ケーブル (もしくはアダプタ)、USB キーボード / マウスを接続し、電源を投入します。

しばらく待つと、インストールする OS を選択する画面が出てきます。Pi Zero W/WH で無線 LAN を利用する場合、ここで上の Wifi networks より SSID を選択し、パスワードを入力してインターネットに接続してください。有線接続の場合は、USB-Ethernet アダプタを接続し LAN ケーブルを接続するだけで完了です。

Raspbianの設定

Raspbian をインストールしたら、各種インターフェースの有効化を行います。ターミナルを起動し `sudo raspi-config` より、**5 Interfacing Options** を選択して、**SSH**と**I2C**をEnable(有効化)します。メニューを選択すると **"would you like the ARM I2C interface to be enabled?"** のダイアログと共に Yes/No の選択が出てくるので、Yes を選択してください。

これで、外部からのリモートログインと GPIO の I2C 通信バスが利用できるようになります。

また、Raspbian のデスクトップ環境は CPU への負荷が大きいため、無効化しコマンドラインのみの操作に切り替えることをお勧めします。切り替えるには `raspi-config` の **3 Boot Options** より **B1 Desktop/CLI** と進み、**B1 Console** を選択します。

以上の設定が終わったら、Esc キーで `Raspi-config` を終了します。

ここで `ifconfig` コマンドで IP アドレスを確認します。ifconfig で見た時の IP アドレスは、`"inet"` の後に続く 4 つの数字の組 (`xxx.xxx.xxx.xxx`) で、基本的には 192.168.x.y (x,y は 1桁から3桁の数字) となります。

無線 LAN を使用している場合は `"wlan0"` の欄を、有線 LAN の場合は `"eth0"` の欄の番号を確認してください

```
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>
mtu 1500
    inet 192.168.1.107 netmask 255.255.255.0
broadcast 192.168.1.255
    inet6 fe80::d671:4443:9927:f0d3 prefixlen
64 scopeid 0x20<link>
    inet6
xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx prefixlen
64 scopeid 0x0<global>
    ether xx:xx:xx:xx:xx:xx txqueuelen 1000 (
イーサネット)
RX packets 1263949 bytes 98661209 (94.0
MiB)
RX errors 0 dropped 247375 overruns 0
frame 0
TX packets 85882 bytes 15968851 (15.2 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0
collisions 0
```

このアドレスは Node-RED にアクセスするために必要になります。メモ帳などに保存しておきましょう。

パソコンより Tera Term

(<https://ja.osdn.net/projects/ttssh2/>) や MobaXterm (<https://mobaxterm.mobatek.net/>) といったリモート SSH クライアントを用い Pi Zero にログインすると、ディスプレイケーブルの繋ぎ換えやディスプレイの設定変更などを行わず Pi Zero を操作できます。

各ソフトの細かい操作法は割愛しますが、基本的に SSH の接続先設定にて

```
"host"/"host name": ifconfigで確認したIPアドレス
"User name": "pi" (デフォルト)
"Password": "raspberry" (デフォルト)
```

と入力すればリモートログインが行えます。このユーザー名とパスワードは変更できるので、デフォルトではセキュリティ的に不安があるという方は変更し、設定を行ってください。

最後に、`sudo apt-get update` と `sudo apt-get upgrade` を行い、アプリのアップデートを行います。これも時間が掛かるので気長に待ちましょう。

ここまでの操作が終わったら、システムを再起動しゼロワンシリーズ拡張基板を取り付けます。

`sudo halt` を入力し、基板上のLEDが点滅しなくなったら電源ケーブルを抜いてください。ここでゼロワンシリーズ拡張基板を取り付け、再度電源を接続し起動します。

■ Node-REDの準備 (node.js / Node-REDのアップデート)

Node-RED は Raspbian にプリインストールされていますが、バージョンが古いためアップデートが必要です。

以下のコマンドを実行してください。

```
"bash <(curl -sL
https://raw.githubusercontent.com/node-red/raspbian-
deb-package/master/resources/update-nodejs-and-nodered)
"
```

※通常よく使われる `update-nodejs-and-nodered` コマンドは Raspberry Pi Zero の場合、前述のコアバージョンの違いから失敗する可能性があります。

これで最新の安定版 Node.js と Node-RED がインストールされます。

では、Node-RED にアクセスしてみましょう。Web ブラウザを立ち上げ、アドレスバーに "[http://\(Pi Zero の IP アドレス\):1880/](http://(Pi Zero の IP アドレス):1880/)" を入力してください。

すると、画像のようなページに飛びます。これが Node-RED のコンソールとなります。以下のコマンドを実行してください。

フロー

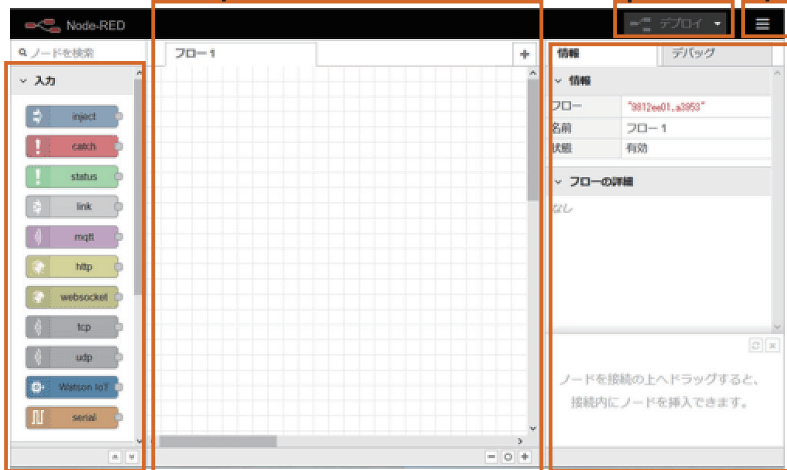
ノードを配置し、結線するエリア。
タブ形式で複数のフローを作成できる。

デPLOYボタン

フローに加えた変更を適用するボタン。
変更が入ると赤くなり、クリック可能になる。

メニュー

各種設定や、フローの
エクスポートを行う。



パレット

各種ノードを用途ごとにまとめた部品箱。
配置したいノードをここからドラッグ&ドロップで配置する。

サイドバー

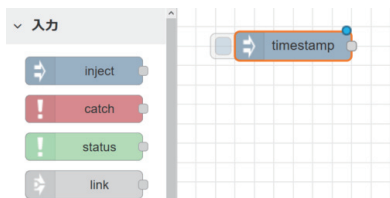
ノードの状態やデバッグノードの
メッセージを確認できる。

中央の“フロー”に左側から“ノード”を配置し、ノード同士を“ワイヤ”で接続することで処理の流れを作り、Raspberry Piやゼロワン拡張基板を制御します。

各ノードは全て左側が入力、右側が出力となっています。

まず最初に、ゼロワン制御基板に搭載されている動作確認用LEDを点滅させるフローを作ってみます。

左側から“inject”ノードを引っ張ってきます。

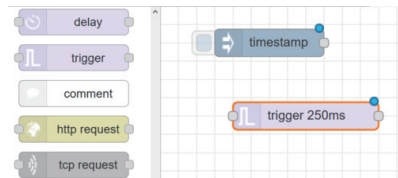


配置したら、ダブルクリックで設定画面を開きます。



“繰り返し”を“指定した時間間隔”に設定し、時間間隔を2秒に設定し、“完了”をクリックします。

次に左側のパレットの“機能”グループより“trigger”ノードを配置します。

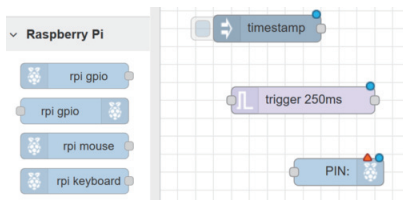


ダブルクリックで設定画面を開きます。



送信後の処理”の時間間隔を1秒に設定します。

最後に、左側のパレットの下の方に“Raspberry Pi”グループがあるので、その中で2つある“rpi gpio”ノードの内、右側にRaspberry Piのロゴマークがある出力ノードを配置し、ダブルクリックで設定を開きます。



”端子”のチェックを”GPIO26 - 37”に入れ、”端子の状態を初期化”にチェックを入れて初期レベルを”Low(0)”にします。

プロパティ

● 端子

3.3V Power - 1	2 - 5V Power
SDA1 - GPIO02 - 3	4 - 5V Power
SCL1 - GPIO03 - 5	6 - Ground
GPIO04 - 7	8 - GPIO14 - TxD
Ground - 9	10 - GPIO15 - RxD
GPIO17 - 11	12 - GPIO18
GPIO27 - 13	14 - Ground
GPIO22 - 15	16 - GPIO23
3.3V Power - 17	18 - GPIO24
MOSI - GPIO10 - 19	20 - Ground
MISO - GPIO09 - 21	22 - GPIO25
SCLK - GPIO11 - 23	24 - GPIO8 - CE0
Ground - 25	26 - GPIO7 - CE1
SD - 27	28 - SC
GPIO05 - 29	30 - Ground
GPIO06 - 31	32 - GPIO12
GPIO13 - 33	34 - Ground
GPIO19 - 35	36 - GPIO16
GPIO26 - 37	38 - GPIO20
Ground - 39	40 - GPIO21

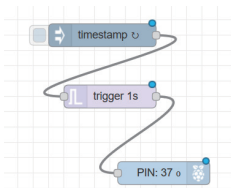
出力形式

デジタル出力

☒ 端子の状態を初期化

端子の初期レベル - Low (0)

ノードを配置したら結線をします。injectノード - triggerノード - rpi-gpioノードの順番に結線してください。

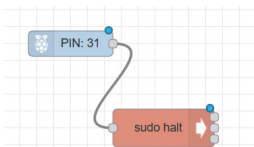


結線が終わったら、右上の”デプロイ”をクリックしてください。これでフローの変更がRaspberry Piに反映され、ノードが動き出します。

ゼロワン拡張基板の”STATUS”ランプを確認してください。1秒ごとに点滅を繰り返していたら成功です。

同様に、ゼロワン拡張基板上のシャットダウンスイッチを機能させるノードを作ってみます。

Raspberry Piグループより、左側にラズパイのロゴがある”rpi gpio”入力ノードを、”機能”グループより”exec”ノードを配置し、結線します。



gpio入力ノードの端子は”GPIO06 - 31”に設定します。

● 端子

3.3V Power - 1	2 - 5V Power
SDA1 - GPIO02 - 3	4 - 5V Power
SCL1 - GPIO03 - 5	6 - Ground
GPIO04 - 7	8 - GPIO14 - TxD
Ground - 9	10 - GPIO15 - RxD
GPIO17 - 11	12 - GPIO18
GPIO27 - 13	14 - Ground
GPIO22 - 15	16 - GPIO23
3.3V Power - 17	18 - GPIO24
MOSI - GPIO10 - 19	20 - Ground
MISO - GPIO09 - 21	22 - GPIO25
SCLK - GPIO11 - 23	24 - GPIO8 - CE0
Ground - 25	26 - GPIO7 - CE1
SD - 27	28 - SC
GPIO05 - 29	30 - Ground
GPIO06 - 31	32 - GPIO12
GPIO13 - 33	34 - Ground
GPIO19 - 35	36 - GPIO16
GPIO26 - 37	38 - GPIO20
Ground - 39	40 - GPIO21

↓ 抵抗

なし

デバウンス

25 ms

☐ デプロイや再起動時に端子の初期状態を読み込む

”exec”ノードは、任意のbashコマンドを実行できるノードです。

コマンド

sudo halt

+ 引数

☐ msg.payload

追加引数

⇒ 出力

コマンド終了時 - execモード

☐ 旧型式の出力を使用(互換モード)

⌚ タイムアウト

任意 秒

🏷 名前

名前

設定の”コマンド”欄に”sudo halt”と入力します。

”引数”のチェックボックスはOFFにしてください。

あとはデプロイをすれば、基板上のシャットダウンスイッチでラズパイZeroをシャットダウンできます。

今回はクラウドのMQTT brokerサービスの”beebotte”を利用しノードに必要な情報をセットし、ブローカと通信するまでをご紹介します。ご期待ください！



Office365・G Suiteとシームレスに連携
クラウド型ビジネスメールセキュリティサービス



TRIHIKO_AJ

形骸化したメールセキュリティを一新。
特定の機密情報を検知し、
メールの誤送信による情報漏洩を防ぐ。

64%

の人がメール誤送信の経験が
あると回答しています。

メール誤送信による
情報漏洩を
自動で防ぐ！

送信済みメール
取消機能

機密情報を
暗号化送信

宛先間違い・添付ファイル間違いなどの誤送信。

取り消せば・・・

と思った経験ありませんか？

TRIHIKO_AJの特徴

01



オペレーション変更不要

普段お使いのOffice365・G Suite
からいつも通りメールを送信するだけ。
インストールや初期設定等面倒な作業は必要
ありません

02



自動検知・自動暗号化

本文・件名・添付ファイル内に
機密情報などの重要なワードが含まれていな
かをチェックし検知されたメールは暗号化送
信。TRIHIKO_AJが自動でサポート。
個人のリテラシーに依存しません。

03



送信済みメール取消し

送信後のメールも自在にコントロール可能。
送信済みメール取消機能で万が一の誤送信の際
も安心。閲覧期限を設定することでセキュアな
情報を必要な時にだけ共有。



TRIHIKO

トリヒコ株式会社
東京都江東区青海2-7-4
<https://trihiko.com>



平日10:00から19:00

03-6428-0737

ラズパイ入門ボードで学ぶ 電子回路の制御[その1]

■ OLEDを使う

ガジェット 米田聡

本連載ではシェルスクリプトマガジンとの連動企画による、小型コンピュータボード「Raspberry Pi」(ラズパイ)のプログラミングが楽しめます。拡張ボード「ラズパイ入門ボード」の応用的な使い方を解説します。その1では OLED への日本語文字の表示、その2では Grove System の光センサーを扱います。

■ ライブラリのインストール

ラズパイ入門ボードには、128×64 ドットの単色有機 EL ディスプレイ(OLED)パネルが搭載されています(図1)。この自発光の OLED は視認性の高いくっきりとした表示が特徴で、さまざまな用途に活用できます。ラズパイ入門ボードの OLED パネルは、インタフェースを含めて米 Adafruit Industries 社(<https://www.adafruit.com/>)の「SSD1306 OLD モジュール」と互換性を持っています。よって、Adafruit Industries が提供しているライブラリ*1 がそのまま使えます。まずはそのライブラリをインストールしましょう。

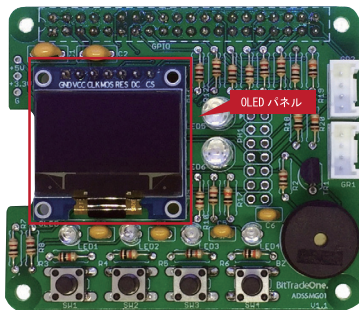


図1 128×64ドットの単色有機ELディスプレイ(OLED)パネルを搭載ラズパイ入門ボードを搭載した Raspberry Pi(ラズパイ)にディスプレイ、キーボード、マウス、(インターネットへ有線が接続するなら)ネットワークケーブル、USB 電源を接続し、標準 OS の「Raspbian」が使える状態にしておいてください。

ライブラリを利用するために、「pip*」をインストールします。Raspbian のデスクトップ画面で端末(LXTerminal)を開いて、次のコマンドを実行してください。

```
$ sudo apt-get update
$ sudo apt-get install python-pip python3-pip
```

続いて

```
$ git clone https://github.com/adafruit/Adafruit_Python_SSD1306.git
```

を実行してライブラリを入手します*2。「Adafruit_Python_SSD1306」ディレクトリ内に最新のソースコードがダウンロードされます。次のように、そのディレクトリに移動してライブラリをインストールします。

```
$ cd Adafruit_Python_SSD1306
$ sudo python setup.py install
$ sudo python3 setup.py install
```

これで準備は完了です。

[pip] 「Pip Installs Python」(文に自身を含める再帰的頭字語)の略で、Python独自のパッケージ管理システムのこと。

*1 最新のソースコードは

「https://github.com/adafruit/Adafruit_Python_SSD1306」から入手できます。

*2 gitコマンドが使えない場合は「sudo apt-get install git」を実行してから再度実行してください。

図2 文字を表示するプログラム (hello.py)

```
1 #!/usr/bin/python3
2 # -*- coding: utf-8 -*-
3
4 import time
5 import Adafruit_GPIO, SPI as SPI
6 import Adafruit_SSD1306
7
8 from PIL import Image
9 from PIL import ImageDraw
10 from PIL import ImageFont
11
12 # OLED のサイズ設定
13 OLED_WIDTH = 128
14 OLED_HEIGHT = 64
15
16 # OLED と SPI バスの設定
17 RST = 24
18 DC = 23
19 SPI_PORT = 0
20 SPI_DEVICE = 0
21
22 # フォントの設定
23 DEFAULT_FONT = '/usr/share/fonts/truetype/
24 fonts-japanese-gothic.ttf'
25 FONT_SIZE = 14
26
27 # SPI オブジェクト
28 spi = SPI.SpiDev(SPI_PORT, SPI_DEVICE, max_speed_hz=8000000)
29
30 # 128x64 ドットの OLED オブジェクト
31 disp = Adafruit_SSD1306.SSD1306_128_64(rst=RST, dc=DC, spi=spi)
32
33 # OLED オブジェクトのスタート
34 disp.begin()
35
36 # OLED オブジェクトのバッファークリアー
37 disp.clear()
38 # バッファの表示
39 disp.display()
40
41 # Image オブジェクトの作成
42 image = Image.new('1', (OLED_WIDTH, OLED_HEIGHT), 0)
43
44 # draw オブジェクトの取得
45 draw = ImageDraw.Draw(image)
46
47 # TryeType フォントオブジェクト
48 jfont = ImageFont.truetype
49 (DEFAULT_FONT, FONT_SIZE, encoding='unic')
50
51 # 文字を image に描く
52 draw.text((0,0), 'こんにちは', font=jfont, fill=1)
53 draw.text((0,16), 'Hello, World', font=jfont, fill=1)
54 draw.text((0,32), 'シェルスクリプトマガジン', font=jfont, fill=1)
55
56 # 文字を image に描く
57 draw.text((0,48), 'ラズパイ入門ボード', font=jfont, fill=1)
58
59 # image を OLED バッファークリアーに書き込む
60 disp.image(image)
61 # バッファークリアーを表示
62 disp.display()
```


日本語を表示させてみよう

OLED パネルは、任意の図形が表示できるグラフィック型です。アルファベットや数字だけでなく、図形や日本語の表示も可能です。特に日本語の表示は需要があるはずなので、その方法を紹介します。

なお、OLED パネルは SPI バスに接続されています。ラズパイの設定で SPI バスを事前に有効化しておいてください*3。

インストールした Adafruit Industries 社製ライブラリは、「Python Imaging Library * 4」(PIL)のイメージオブジェクトを表示する設計になっています。このイメージオブジェクトは「draw」クラスを使って図形を描いたり、イメージファイルを読み込んだりといったことが簡単にできるので非常に便利です。TrueType フォントを読み込んでイメージオブジェクトに描くことも可能なため、日本語も表示できます。

ただし、日本語を表示するには、日本語 TrueType フォントが必要です。次のように「Takao フォント」をインストールしておきましょう。

```
$ sudo apt-get install fonts-takao
```

文字を表示する、Python のサンプルプログラムは「hello.py」です(図2)。エディタでこのプログラムを作成し、次のように実行権を与えてください。

```
$ chmod +x hello.py
```

そして

```
$ ./hello.py
```

のように実行すると、「こんにちは」「Hello, World」「シェルスクリプトマガジン」「ラズパイ入門ボード」という4行のメッセージが表示されます(p.52の図3)。シェルスクリプトマガジンは長いので、途中で表示が切れています。

GPIO や SPI バスの制御に Adafruit Industries 社製のライブラリを用いています。hello.py の 28 行目で、まず SPI バスを初期化しています。

```
spi = SPI.SpiDev(SPI_PORT, SPI_DEVICE, max_speed_hz=8000000)
```

図3 hello.pyの実行結果

「こんにちは」「Hello, World」「シェルスクリプトマガジン」「ラズパイ入門ボード」を表示。



第1引数はSPIポート番号、第2引数はデバイス番号(チップセレクトの番号)。「max_speed_hz」パラメーターはSPIバスのクロック周波数です。OLED パネルは 8MHz 程度まで対応できるので「8000000」(8M)を指定しています。

初期化によって生成した「SPI」オブジェクトを使って、さらに31行目で「Adafruit_SSD1306」オブジェクトを生成します。

```
disp = Adafruit_SSD1306.SSD1306_128_64(rst=RST, dc=DC, spi=spi)
# 表示スタート
disp.begin()
```

「rst」パラメーターはリセット端子が接続されているGPIO番号で、ラズパイ入門ボードでは「24」です。「dc」パラメーターは、OLED パネルに対するデータとコマンドを区別するD/C端子がつながっているGPIO番号で、ラズパイ入門ボードでは「23」です。最後の「spi」パラメーターがSPIオブジェクトです。

Adafruit_SSD1306 オブジェクトで注意が必要なのは、内部にバッファを持っており、そのバッファにPILイメージオブジェクトを書き込んだ上で、OLEDに表示させる手順を踏む点です。

ディスプレイを消去する場合も次のようにバッファをクリアして表示処理を実行します(37行目と39行目)。

```
disp.clear()
disp.display()
```

バッファに書き込むことができるPILイメージオブジェクトは、モノクロなので2値です(42行目)。

```
image = Image.new('1', (OLED_WIDTH, OLED_HEIGHT), 0)
```

最初の引数で作成するPILイメージオブジェクトのモードです。

「1」を指定すると2値モノクロとなります。2番目の引数はイメージのサイズです。最後の引数には黒に設定する値(0)を指定します。

あとは、PILの機能を使って文字や図形を「image」オブジェクトに描き、最後に、次のようにバッファに書き込んで表示します(57行目と59行目)。

```
disp.image(image)
disp.display()
```

PILに関しては、さまざまなドキュメントがインターネットに掲載されているので、それらを参考にすれば簡単に図形や文字を描くことができます。この連載でも今後、PILを活用してOLEDにさまざまな情報を表示していきます。

*3 SPIバスを有効にするには、まずデスクトップ画面左上メニュー(ラズベリーマーク)の「設定」>「Raspberry Piの設定」を選びます。開いた「Raspberry Piの設定」ダイアログの「インターフェイス」タブにある「SPI」で「有効」を選んで「OK」ボタンをクリックした後にラズパイを再起動します。

*4 PILの公式サイトは「<http://www.pythonware.com/products/pil/>」です。

ラズパイ入門ボードで学ぶ 電子回路の制御[その2]

■ 光センサーを使う

ガジェット 米田聡

その1ではOLEDへの日本語文字の表示を行いました。その2ではGrove Systemによる光センサーの使用方法を学びましょう。ラズパイ入門ボードには「Grove System」のコネクタが2個用意されています。今回は、Grove Systemの概要と使い方を紹介します。電子回路としては光センサーを扱います。

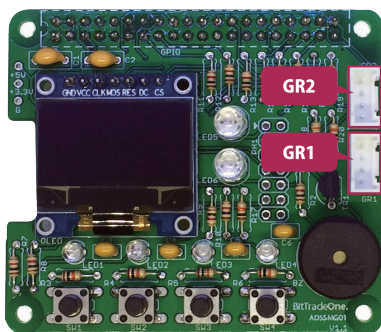
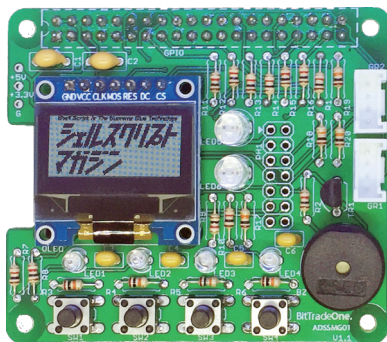


図1 Grove Systemのコネクタを2個実装

■ ラズパイ入門ボードのGrove Systemコネクタ

前述したように、ラズパイ入門ボードには「GR1」と「GR2」という二つの Grove System コネクタが実装されています(図1)。GR1には「GPIO21」ピンと「GPIO25」ピンのGPIO(汎用I/O)が、GR2には「PC」が接続されています(図2)。

ICは「Inter-Integrated Circuit」の略で、IC(半導体チップ)間を結び簡易なシリアルインタフェースです。Raspberry Piをはじめとするコンピュータボードで盛んに利用されているほか、パソコンでも「SMBus」という名称でマザーボード上のセンサーを接続するインタフェースとして使われています。

ICでは、データを送受信する「SDA」と、データクロックを送信する「SCL」という2本の線を使ってセンサーなどと通信します。電源を含めても、わずか4本の線でセンサーなどを接続できるのが特徴です。

また、ICをサポートするデバイスは、個々にアドレスを持っています。ホスト側からそのアドレスを指定して数珠つなぎにした複数のICデバイスと個別にデータをやり取りできるようになっています。ただ、Grove SystemではICを基本的に1対1接続のインタフェースとして利用しており、数珠つなぎには対応していません。その分だけICの扱いもシンプルになります。

■ Grove Systemとは

Grove Systemは、中国 Seeed Studio 社が展開しているセンサーやディスプレイ、そのほかの表示装置などを接続する規格の一種です。

Raspberry Piをはじめとするコンピュータボードにセンサーなどを接続する場合、実験的にはブレッドボードで対応できます。しかし、常用するには、はんだ付けの作業が必要になります。また、センサーとRaspberry Piをつなぐインタフェースの仕様に沿った配線も必要です。

こうした作業はさほど難しいものではありませんが、電子工作に慣れていない人にとってはハードルになるでしょう。Grove Systemでは、そんな電子工作の初心者でも簡単に扱えるようにコネクタの仕様を定めています。

4ピンのコネクタでコンピュータボードと、センサーや表示装置などを実装した基板を接続するだけで基本的には利用できます。採用されているコネクタは方向を間違え挿すことがない切り欠きの入ったものです。通常、接続するためのケーブルは、センサーや表示装置などを実装した基板に付属しています。

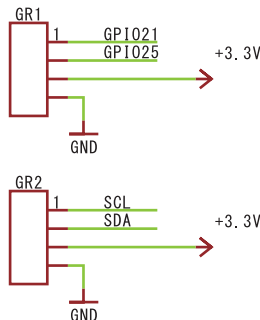


図2 GR1とGR2の回路図

■ 入門ボードでGrove Systemを使う注意点

ラズパイ入門ボードの Grove System コネクタを利用するとき、いくつかの点に留意します。

一つは動作電圧です。Grove System はさまざまなコンピュータボードをサポートしており、Raspberry Pi も Grove System の対応ボードとされています。インタフェースの動作電圧は、コンピュータボードによって異なります。ラズパイ入門ボードでは、Raspberry Pi の GPIO および I²C を直結しているため動作電圧は 3.3V です。

一方、Grove System がもっとも盛んに使われているのは「Arduino」というコンピュータボードで動作電圧は 5V です。Grove System 対応のセンサーなどの中には、5V のみに対応するものもあり、そうしたセンサーはラズパイ入門ボードでは使えません。

また、Grove System ではアナログセンサーを扱うことができますが、ラズパイ入門ボードはアナログ・デジタル変換回路(ADC)を搭載していません。Raspberry Pi もアナログの入力端子を持たないので、アナログセンサーを使うことはできません。

最後に、Raspberry Pi では「GrovePi+」という拡張ボードを介して対応センサーなどを接続するのが Grove System 公式の方法です。そのため、Raspberry Pi 向けに配布されている Grove System 用のサンプルコードやライブラリも GrovePi+ を前提にしています。

ラズパイ入門ボードの Grove System コネクタは前述のように Raspberry Pi の GPIO と I²C に直結されているので、Grove System 向けのサンプルコードやライブラリがそのままでは使えず、手直しが必要です。

■ Digital Light Sensorを使ってみよう

Grove System 対応センサーの利用例として今回は「Digital Light Sensor」(http://wiki.seeedstudio.com/Grove-Digital_Light_Sensor/)を使ってみます。

Digital Light Sensor は環境光などの強度を測定できるセンサーです。国内で Grove System を扱っているスイッチサイエンスや千石電商、秋月電子通商などで入手することができます。図 3 のようにセンサーが載った小さな基板と Grove System ケーブルのセットです。この基板上には米 TSL2561 社製の「TSL2561」という I²C 接続の光学センサーが搭載されています。

I²C なので、前述の通りラズパイ入門ボードの GR2 にセンサーに付属するケーブルを取り付け、センサー搭載基板と接続してください(図 4)。

Raspberry Pi の標準 OS「Raspbian」のデフォルトでは I²C インターフェースが有効化になっていません。連載の初回でも説明していますが、ラズパイ入門ボードを使うときには「SPI」と「I²C」を有効化しておく必要があります。まだしていない場合は、Raspberry Pi の設定ツール(raspi-config)を使って有効化しておいてください。

有効にしたら、Raspbian 上で次のページのコマンドを実行し、I²C 用のユーティリティをインストールします。

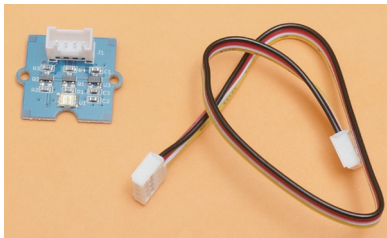


図 3 Digital Light Sensor

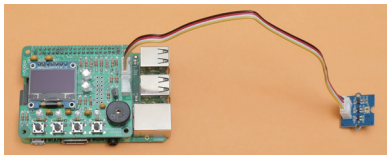


図 4 Digital Light Sensorとラズパイ入門ボードを接続する

```
$ sudo apt update
$ sudo apt install i2c-tools
```

インストール後、次のコマンドを実行して I²C デバイスをリストアップします。

```
$ i2cdetect -y 1
```

```
$ i2cdetect -y 1
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
    (略)
```

図 5 I²C デバイスのリストアップ

図 5 のように、Digital Light Sensor のアドレス「0x29」が表示されます。これで Digital Light Sensor が正常に認識されていることが確認できました。

■ OLEDに現在の照度を表示させてみよう

では、ラズパイ入門ボードの OLED パネルに現在の照度を表示させてみましょう。

まず、TSL2561 を使うためのライブラリを入手します。前出の Grove Studio 社の公式 Web ページにも Raspberry Pi 用の Python プログラムサンプルが掲載されていますが、これは使えません。理由はいくつかあります。Python 2 専用であること、そして Adafruit Industries 社製の古い I²C ライブラリを基に作られているため、現在の同社製のライブラリでは動かないことが大きな点です。書き換えるのは簡単ですが、あえて書き換えて使うだけのメリットはないでしょう。

代わりに Python パッケージ管理コマンドの「pip」を使って「TSL2561」を入手します。ここでは、OLED を使うために前回解説した Adafruit Industries 社製のライブラリが既に Raspbian にインストールされている前提で話進めていきます。まだインストールしていない場合は、前回の連載を見ながらインストールしてください。

```
$ sudo apt install python3-pip
```

pip3 を使って tsl2561 ライブラリを導入します。

```
$ sudo pip3 install tsl2561
```

OLED に文字列を表示する簡易ライブラリ「OLED.py」を作成しました。図 6 の内容を OLED.py ファイルとしてカレントディレクトリーに保存してください。

```
1 import time
2 import Adafruit_GPIO.SPI as SPI
3 import Adafruit_SSD1306
4
5 from PIL import Image
6 from PIL import ImageDraw
7 from PIL import ImageFont
8
9 class OLED():
10     OLED_WIDTH = 128
11     OLED_HEIGHT = 64
12
13     RST = 24
14     DC = 23
15     SPI_PORT = 0
16     SPI_DEVICE = 0
17
18     DEFAULT_FONT = '/usr/share/fonts/truetype/fonts-japanese-gothic.ttf'
19     FONT_SIZE = 14
20
21     _LINE_HEIGHT = 16
22
23     def __init__(self, port=SPI_PORT, dev=SPI_DEVICE, rst=RST, dc=DC):
24         self._spi = SPI.SpiDev(port, dev, max_speed_hz=8000000)
25         self._disp = Adafruit_SSD1306.SSD1306_128_64(rst=rst, dc=dc, spi=self._spi)
26         self._disp.begin()
27         self._disp.clear()
28         self._disp.display()
29         self._image = Image.new('1', (self.OLED_WIDTH, self.OLED_HEIGHT), 0)
30         self._draw = ImageDraw.Draw(self._image)
31         self._font = ImageFont.truetype(self.DEFAULT_FONT, self.FONT_SIZE, encoding='unic')
32
33     def put_string(self, str, line=0):
34         self._draw.rectangle((0, line*self._LINE_HEIGHT, self.OLED_WIDTH, line*self._LINE_HEIGHT+self._LINE_HEIGHT), fill=(0))
35         self._draw.text((0, line*self._LINE_HEIGHT), str, font=self._font, fill=1)
36         self._disp.image(self._image)
37         self._disp.display()
```

図 6 OLED に文字列を表示する簡易ライブラリ (OLED.py)

```
1 #!/usr/bin/python3
2 # -*- coding: utf-8 -*-
3
4 from tsl2561 import TSL2561
5 from OLED import OLED
6 import time
7
8 oled = OLED()
9 # アドレスを指定する
10 tsl = TSL2561(address=0x29)
11
12 oled.put_string('Current LUX')
13
14 while True:
15     # lux() メソッドで現在の照度が得られる
16     led.put_string(str(tsl.lux()), 1)
17     time.sleep(1)
```

図 7 照度を表示するサンプルプログラム (lux.py)

照度を表示するサンプルプログラムは「lux.py」です(図 7)。「lux.py」を次のように実行しましょう。

```
$ python3 lux.py
```

図 8 のように OLED に現在の照度(ルクス)が表示されます。照度を得るのは簡単で、TSL2561 オブジェクトの「lux()」メソッドを呼び出すだけです。照度は整数値で得られます。「str()」関数で文字列に変換し、OLED に表示しています。

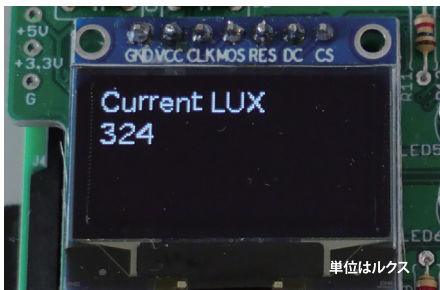


図 8 照度が表示された

このように、既存のライブラリと Grove System のセンサーを使うことで、Raspberry Pi に環境情報を簡単に取り込むことができるわけです。ぜひ試してみてください。

ラズパイではじめる 電子制御とIoT

Assembly Desk Books volume02

■ 奥付

発行

株式会社 ビット・トレード・ワン

企画構成

株式会社 ビット・トレード・ワン 東京デザイン室

協力

ユニバーサル・シェル・プログラミング研究所

シェルスクリプトマガジン編集部

連絡先

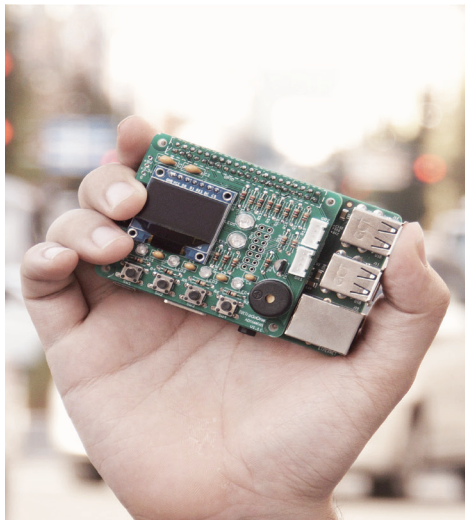
info@bit-trade-one.co.jp

印刷・製本

Gr@phic

発行日

初版 2018/12/18



奇跡のダーク 作: 桃町正暗

俺はプロの拉麺喰い。

日々、何もわかってないヤツらとの闘いに明け暮れている。だが、そんな俺にも休息はある。

東京の東、浅草まで東じゃないが、千住ほど北でもない。そんな駅の駅から行っても、それほど遠くはない、とある一軒の立ち飲み屋。

戦いに疲れた俺の心を癒し温めてくれる一杯の酒、赤星大瓶からそそく黄金色の美酒。何ものにも代え難い戦士の休息といったところが・・・

ここいらでは焼き鳥より焼きトンが主流を占めカシラ、テッポー、タンスジ、キンツル一見、素人筋では頼むこともできない呪文のような品名、頼もうものなら「タンバカシラシオオタレカイチー！」と威勢のいい店員が、オーダーをさらに蒸留した呪文にして焼き場に通す。

そんなプロの癒しの酒場にヤツらは来た。戦士の感が働いたのか、カウンターで赤星を注ぐ俺はビールの入ったヒヤタンを本能的に強く握りしめる。カウンターだけの店はすでに肩が不触れ合うくらいに込みあっている。すでに酔客でいっぱいのカウンターに надаれ込んでくる男女。圧された俺は波打つ液面をすんでのところで抑え耐えなきを得た。「あーココ、ココ 安いよ!」「1本 100円だって、マジ安っ!」入るはしから「安い、安い」を連発するヤツら。

馬鹿目! 安いのは 20 年前からだ。その間、値上がりしたのは 2 回、消費税導入と増税のタイミングで、お前らがヨチヨチ歩きのことだ。

「カシラって何ー?」知性の感じられない声で女が聞く。「カシラっていうくらいだから頭のてっぺんの肉じゃね。」

馬鹿目! カシラは頭ではなく豚のこめかみから頬にかけての肉だ! さらに細かくわけると赤身系と脂身の多い白系があり、それはアブラとして部位分けされる場合もある。いきなり間違えるとは、実戦なら即死だったな。若い・・・俺はヤレヤレといった感じで焼き場の 2 代目大将に目線を送ったが、大将は焼きに夢中で、まるで気づいていないようだ。混みあいやカオス気味のカウンター、横の小うるさい男女を尻目に俺は夢想に入る。「おい! 若い! カウンターは腰で立つんだ!」このカウンターでとある爺さんに 10 数年前に言われた言葉だ。「最近の若いもんは、へっすり腰でいかん。カウンターに寄りかかってケツを突き出すすじゃねえ!」持ち込みのワンカップをグビリとやりながら赤ら顔の爺さんは誰に語るわけでもなく独り言のようにつぶやく。もちろんこの店では持ち込みはご法度だ。

「そんなんじや、ダークできねえだろ?」ダークとは、いわゆる昭和歌謡におけるダークタックス。カウンター激混み時に生み出されたポジションで彼らが体を斜めに重ね、並んでいる姿をカウンターに再現した飲み方だ。大体混んでくるとおかみさんの「みんな! ダークして!」の合図で始まったものだが、もう 10 数年前、のれんを蹴飛ばして冷酒を一気のみして帰っていくような威勢のいい爺さん。もうとっくに死にしまったに違いない。

「アタシ、カシスウーロンハイ!」唐突な女のオーダーにふと我に帰った俺、こんなメニューがこの店にあったのか・・・。

白木と呼ぶには年相の入らずなカウンターには不似合いなハイサワージョッキに入ったカシスウーロンハイが置かれ色のどぎついマニキュアの指がさって行く。そこにさらに白髪の中年がカウンター端に入ってくる。もうカウンターは一人も入れそうにないが、この店ならではのオレンジ色のおしぼりがさっとカウンターに置かれホッピー白セツが中年の前に置かれる。

その刹那、俺は見た! 白髪の子供からつらつ折りのように重なっていく親父たちのダークで!!

ゆるやかなアーチを描き新体操のように美しい親父たちのダーク、みな覚えていた。酔っ払いの約束のリング、俺もすかさず、この美しいアーチの一部になるが、しかし奇跡のダークは続かない。「混んできたね!」女は、そう囁くとき自分の方向に男を手繰りよせ、俺たちのダークとは逆方向に陣を張る。

「クツツィチャッタ!」女が言うやいなや男の手が腰に回る。眼前で展開されるヤツらのいちゃつきを眺めながら、別れた女に無性に電話を掛けたくなった俺は「まったく最近の若いモンは・・・」という言葉をやめなくなった赤星と一緒に飲み干すしかなかった。



Digi-Key社取扱い^{※1}全製品を
1個^{※2}からマルツで購入できます

3,000円^{※3}以上
お買い上げで

送料無料

通常送料全国一律

240円～450円

マルツは
Digi-Key社の

正規代理店です



www.marutsu.co.jp

マルツオンライン 



※1 マルツオンラインに一部未掲載製品がございますので、別途お見積りいたします。

※2 LOT 単位・複数個単位の製品もございます。

※3 マルツオンラインでの一度のご購入金額の合計。Digi-Key社製品に限らず、マルツオンライン掲載の全製品での合算になります。

● 価格はすべて税抜表記です。

Linux/Unixシステム、プログラミング総合誌

シェルスクリプト マガジン

Linux/Unix、シェルスクリプト/各種プログラミング言語、人工知能/深層学習、IoT/組み込み機器、IT開発/マーケティングの総合情報誌です。わずか500円(税別)で、さまざまなジャンルにおける注目の技術や手法を紹介しています。

- ★ LinuxやUnixの知識が身に付く
- ★ ラズパイや電子工作が始められる
- ★ プログラムが書ける
- ★ 人工知能と機械学習が理解できる
- ★ 開発やマーケティングの手法が分かる
- ★ レガシーシステムを移行できる

シェルスクリプトマガジン公式サイト
<https://shell-mag.com/>

定期購読

年間料金：3,000円(送料サービス)

申し込みURL

<https://www.usp-lab.com/pub.order.html>

気軽に
買える!

1コイン
ITマガジン

気軽に
読める!



サイズ：B5判
定価：本体500円+税
隔月(奇数月25日発売)

直販、Amazonおよび、20店舗以上の全国書店

●直販サイト

<https://www.usp-lab.com/pub.magazine.html>

●取り扱い書店一覧

<https://www.usp-lab.com/pub.store.html>

スタッフ募集!

システムエンジニア(SE)、プログラマー

URL <https://www.usp-lab.com/recruit.html> ユニバーサル・シェル・プログラミング研究所