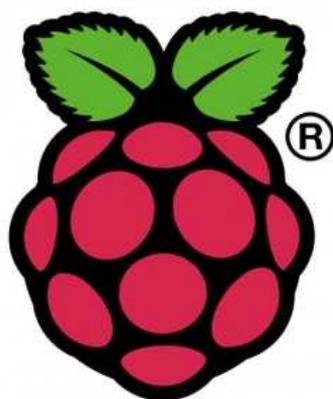


IoT 入門

Raspberry Pi で学ぶ「IoT」



目 次

1 章	Raspberry Pi の導入と設定	
1	本書の IoT 学習環境	1
2	Raspberry Pi とは	2
3	Raspberry Pi の構成	3
4	Raspberry Pi で使用できる OS	7
5	Raspberry Pi OS のダウンロード	8
6	Raspberry Pi OS のインストールと初期設定	15
7	Raspberry Pi のシャットダウン	18
8	日本語変換ツールのインストールと設定	19
2 章	Raspberry Pi OS の使い方	
1	アプリケーション概要	22
2	プログラミング	23
3	教育・教養	24
4	オフィス	24
5	インターネット	27
6	設定	28
7	GUI と CLI	29
8	Raspberry Pi の操作	30
9	アプリケーションの追加と削除 (CLI)	33
10	アプリケーションの追加と削除 (GUI)	35

1 章 Raspberry Pi の導入と設定

本書のIoT 学習環境

製造現場の情報の見える化を行うセンサ系プロトタイプ（実証模型や試験）を低コストで作成する環境は2020年現在、大きく2つ Arduino（アルデュイーノ）と Raspberry Pi（ラズベリーパイ）があります。

アルデュイーノはマイコン（マイクロコンピュータ）、ラズベリーパイは教育用コンピュータが開発コンセプトです。どちらもセンサを接続しプログラムで制御できます。



アルデュイーノ・レオナルド



ラズベリーパイ 3 B+

以下に主な違いを挙げます。

	Arduino	Raspberry Pi 3 B + / 4 B
コンセプト	マイコン	教育用コンピュータ
OS	なし	Linux(Raspberry pi OS)など
使用言語	Processing (C++に似ている)	C/C++/Python/Node.js/mRuby など
複数処理	できない。1つのタスクのみ。	できる。マルチタスク可能。
CPU処理能力	小さい	大きい
メモリ	小さい (uno 0.002MB)	大きい (Pi 3B 512MB/ Pi 4B 1~4GB)
ネットワーク	オプションで可能	イーサネット/Wifi
特徴	一つのセンサ制御が簡単で速い。省電力。 電子回路・ハードウェアの学習向き	一つのセンサ制御が複雑で遅い。 プログラミング・ソフトウェアの学習向き

このように、電子回路やセンサ、ロボットなどの制御にはアルデュイーノが向いています。プログラムを組んでセンサからデータを得て、さまざまなソフトウェアを利用してデータを処理する学習はラズベリーパイが適しています。本書はIoTのプロトタイプを作成し現場で生かす知見を得る目的を達成するため、ラズベリーパイを用いて学習を進めます。

1 Raspberry Pi とは

Raspberry Pi は、プリント基板 1 枚のシングルボードコンピュータです。低価格・低消費電力で小型（カードサイズ）です。安価で丈夫で子どもがプログラミングで遊べるコンピュータを目指してケンブリッジ大学のエベン・アプトン氏がラズベリーパイ財団を創設し開発しました。2012 年 2 月発売開始から 2019 年 12 月に累計販売台数が 3000 万台を超えるほど全世界的に普及を続けているコンピュータです。

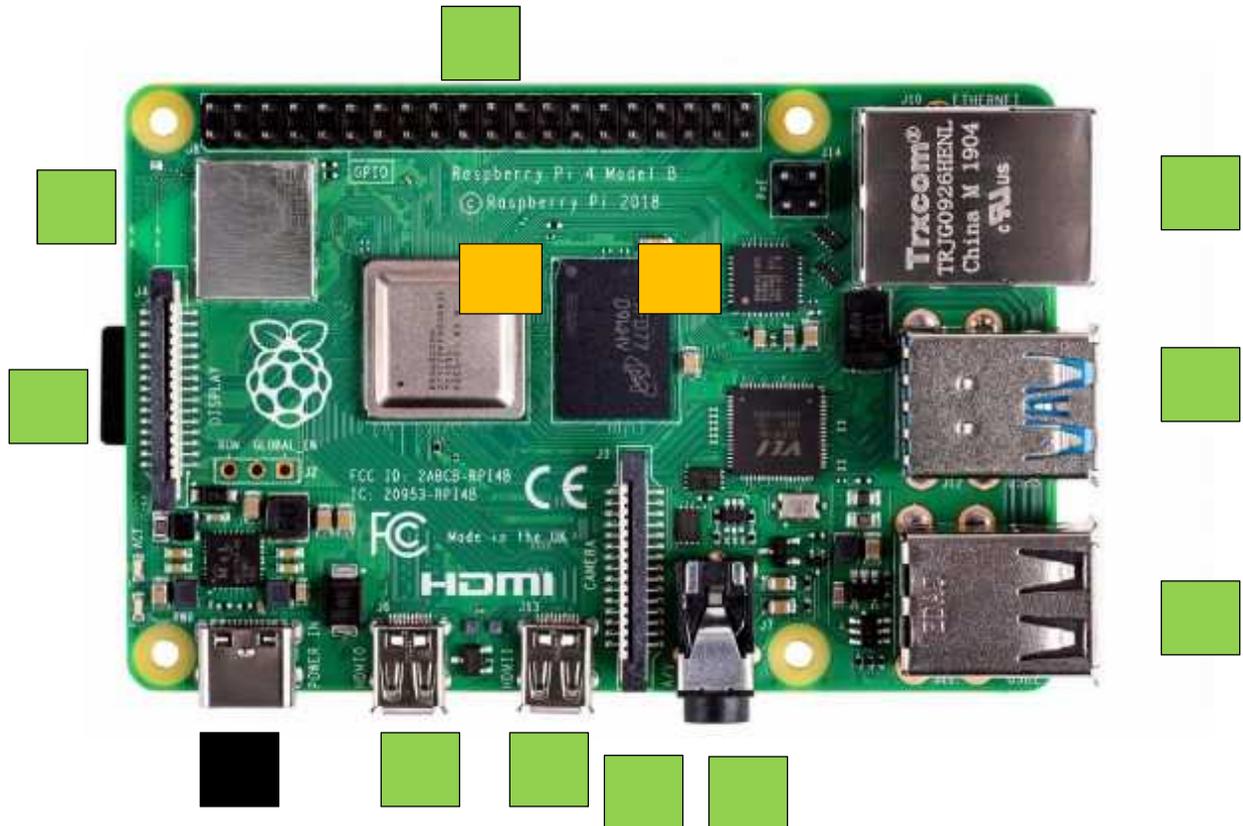
ラズベリーパイに OS をインストールすると、プログラミングツール、データベース・ドキュメント・表計算・図形描画などビジネス用のソフト、インターネット、動画音楽プレイヤー、ゲームなどが使えます。また、ディスプレイやキーボード・マウス、音声端子を接続するインターフェースだけでなく、GPIO（General Purpose Input/Output）を使うことで電球やモーターの制御やセンサーを接続して情報を得ることもできます。もちろんカメラを接続することもできます。

産業用としては 2013 年から使われ始め、2020 年現在では約 60% が産業用途向けとして出荷されています。通常のラズベリーパイは環境の劣悪な製造現場に耐える設計になっていないため、産業用途向けに防塵防水、耐衝撃耐振動性を備え熱暴走対策などが施されています。

3 Raspberry Pi の構成

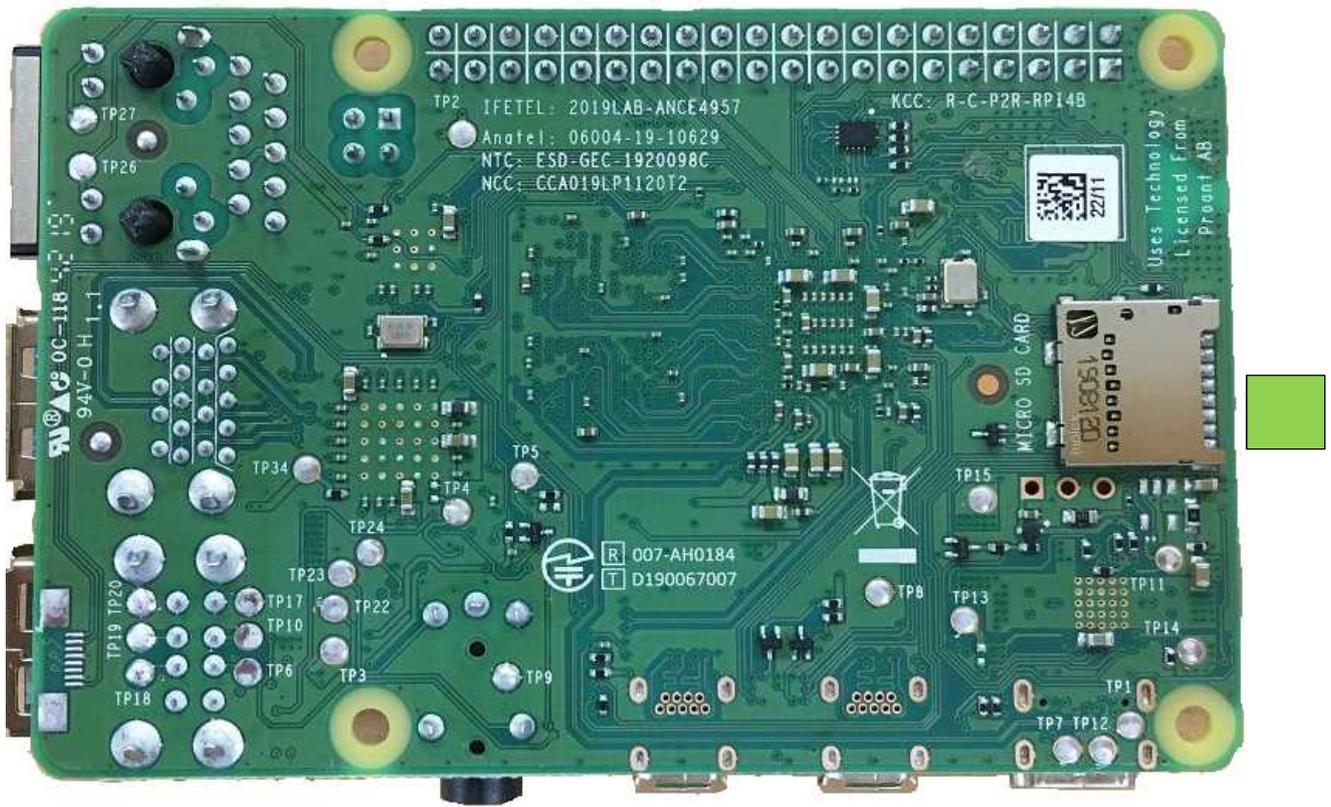
ラズベリーパイは、2020年現在第一世代から第四世代まで発売されています。本書ではRaspberry Pi 4 B と 3B+ について解説します。

< Raspberry pi 4 B 正面 >



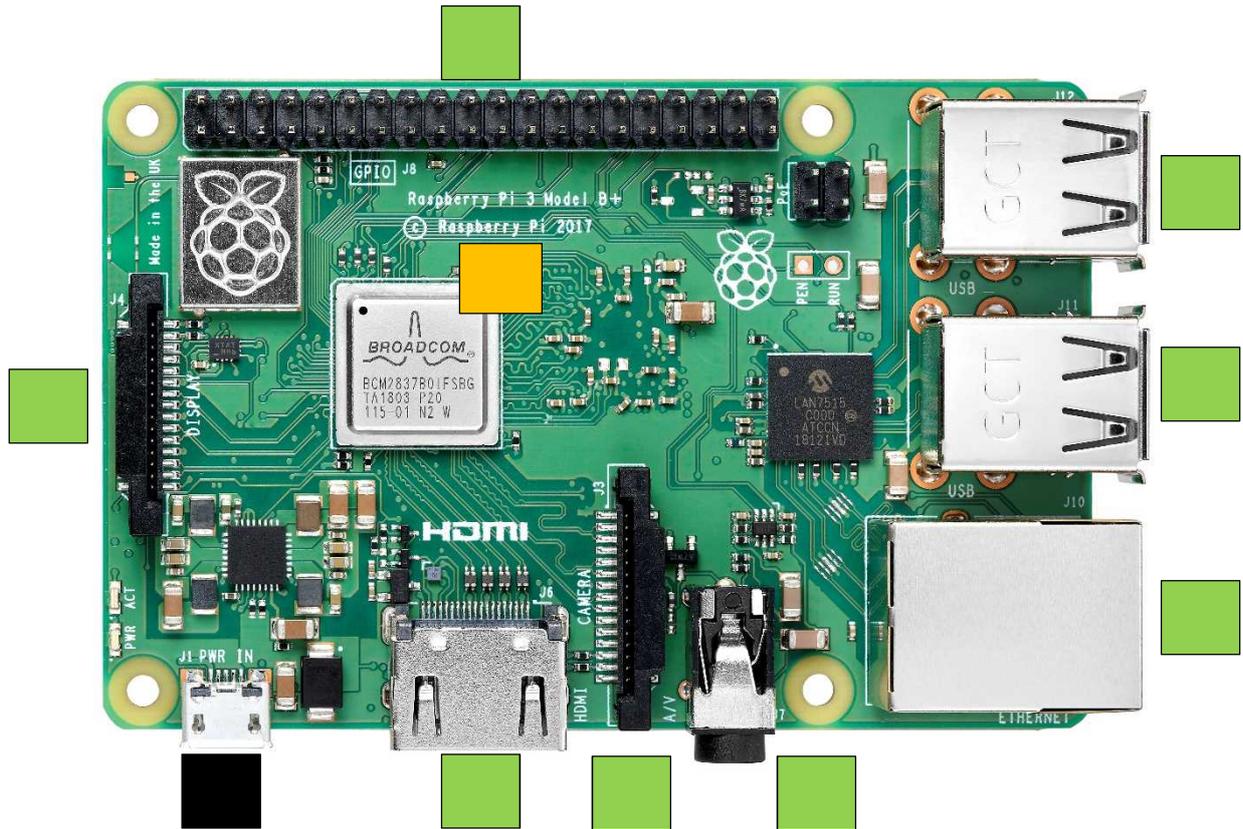
	C P U (SoC: System on a Chip)	主要な部品を1つにまとめた I C 1.5 GHz
	メモリ (R A M)	1 GB / 2 GB / 4 GB
	電源 (USB - C)	5 V 3 A
	USB2 ポート × 2	U S B 2.0 規格が 2 ポート。
	USB3 ポート × 2	U S B 3.0 規格が 2 ポート。
	ネットワークコネクタ (Gigabit Ethernet)	ネットワークケーブルの接続で通信ができる
	micro H D M I 端子 (映像出力端子) × 2	4 K デュアルディスプレイに対応。
	カメラシリアルインターフェース	カメラモジュールを接続
	オーディオ出力端子	3.5 ジャック
	ディスプレイシリアルインターフェース	タッチスクリーンディスプレイを接続
	G P I O (電子回路接続端子)	センサーなど電子回路を接続する
	Wifi/ ブルートゥース受信機	ブルートゥース 5.0

< Raspberry pi 4 B 背面 >



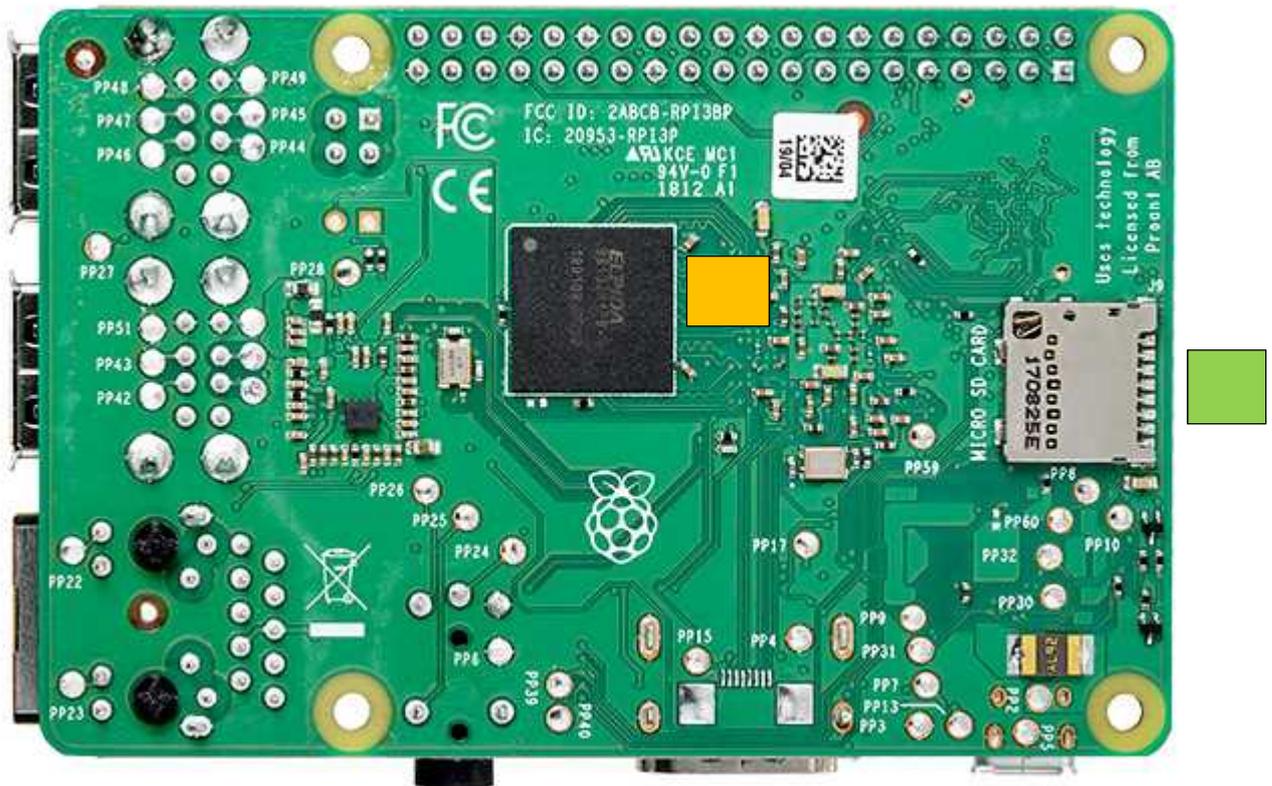
Micro S D カードコネクタ	OS を格納した S D カードを挿入する
-------------------	-----------------------

< Raspberry pi 3 B + 正面 >



	C P U (SoC: System on a Chip)	主要な部品を1つにまとめたI C 1.4 GHz
	電源 (microUSB)	5 V 2.5 A
	USB ポート × 4	U S B 2.0 規格が 4 ポート。キーボードやマウスや外付けH D Dなどを接続する
	ネットワークコネクタ (300Mbps Ethernet)	ネットワークケーブルの接続で通信ができる
	H D M I 端子 (映像出力端子)	H D M I 対応のディスプレイに接続する
	オーディオ出力端子	3.5 ジャック
	カメラシリアルインターフェース	カメラモジュールを接続
	ディスプレイシリアルインターフェース	タッチスクリーンディスプレイを接続
	G P I O (電子回路接続端子)	センサーなど電子回路を接続する
	Wifi/ ブルートゥース受信機	ブルートゥース 4.2

< Raspberry pi 3 B + 背面 >



メモリ (1 G B)	データのやり取りをするメモリ
Micro S Dカードコネクタ	O S を格納した S Dカードを挿入する

4 Raspberry Pi で使用できる OS

パソコンの OS (Operating System) を大きく 2 つに分けると、Windows 系と Mac 系に分かれます。Windows を OS とする PC はソフトウェアの種類が多く、PC も安いのが特徴です。しかし、Windows OS が有料で、脆弱性やソフトの相性などの問題が多く更新プログラムが頻繁に提供されています。しかもサポートが終了すると更新プログラムが配布されなくなることはご承知のとおりです。また、MacOS は、無料で使え Windows ほど脆弱性などの問題が生じにくいものの、MacOS 搭載 PC の価格が高いことがデメリットです。

Linux はソースが公開されていて誰もが無料で使えて改造ができるオープンソースの OS です。無料のソフトウェアも多く、Windows や MacOS との互換性もあるウイルスやマルウェアの影響を受けにくい特徴があります。ラズベリーパイは、Linux (リナックス) を簡単に使えるようにしたソフトウェアのパッケージ (Linux ディストリビューション) をインストールして使います。本テキストでは、代表的なディストリビューション

Raspberry Pi OS (旧 Raspbian (ラズビアン)) のインストールの解説を行います。

ラズベリーパイのスターターキットをお持ちの方は、キットに OS ダウンロード済の microSD が付属しているので、次の「4 Raspberry Pi OS のダウンロード」は省略して、「5 Raspberry Pi OS のインストールと初期設定」に進んでください。

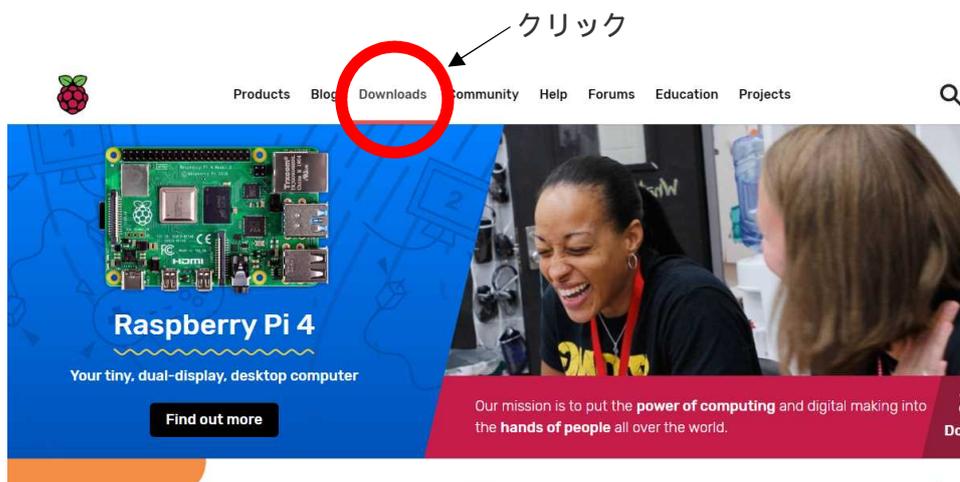
5 Raspberry Pi OS のダウンロード

2020年8月の公式サイトをもとに作成しています。

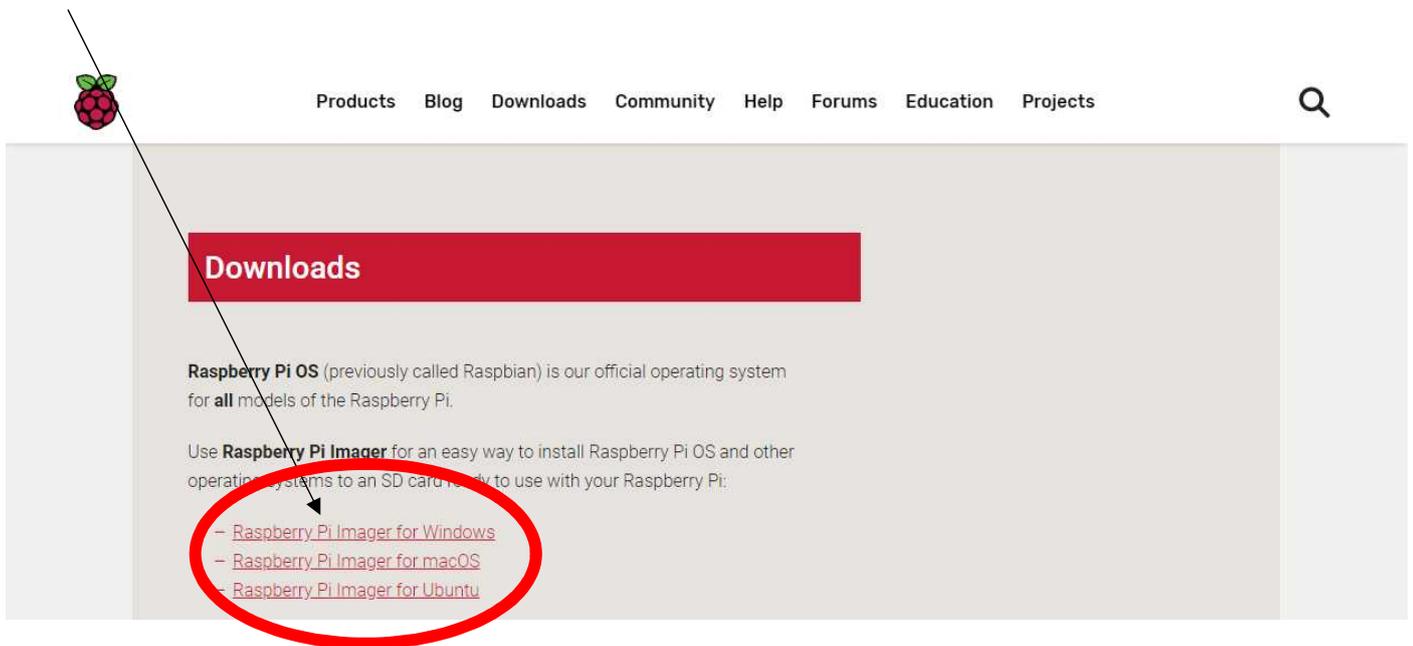
このテキストでは、2020年2月から導入された「Raspberry Pi Imager」を使ったダウンロード方法を解説します。なお、「Raspberry Pi Imager」を使わない方法として、～でNOOBSを使った方法を解説しています。

ラズベリーパイのスターターキットをお持ちの方は、キットにOSダウンロード済の microSD が付属しているので、この「4 Raspberry Pi OS のダウンロード」は省略して、次の「5 Raspberry Pi OS のインストールと初期設定」に進んでください。

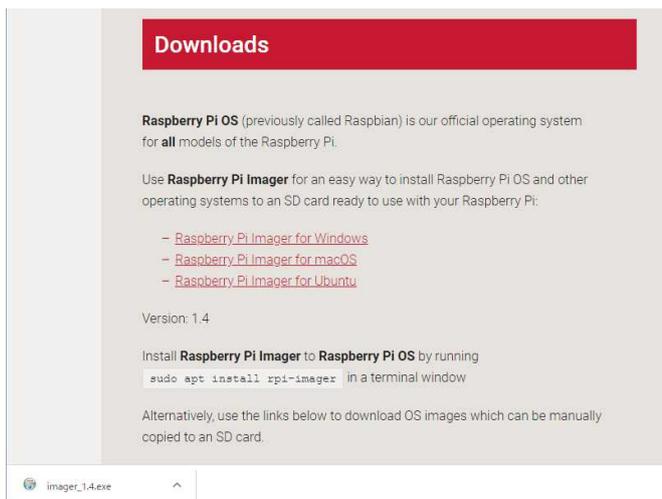
Raspberry Pi 公式サイト (<https://www.raspberrypi.org/>) を開き、Downloads をクリックします。



Raspberry Pi 公式サイト (<https://www.raspberrypi.org/>) を開き、Downloads をクリックします。Raspberry Pi Imager をクリックします。「Raspberry Pi Imager for Windows」を選んでください。(mac の方は macOS を選びます。)



ダウンロードが完了したら、下のタスクバーの「imager」の exe ファイルをクリックします。

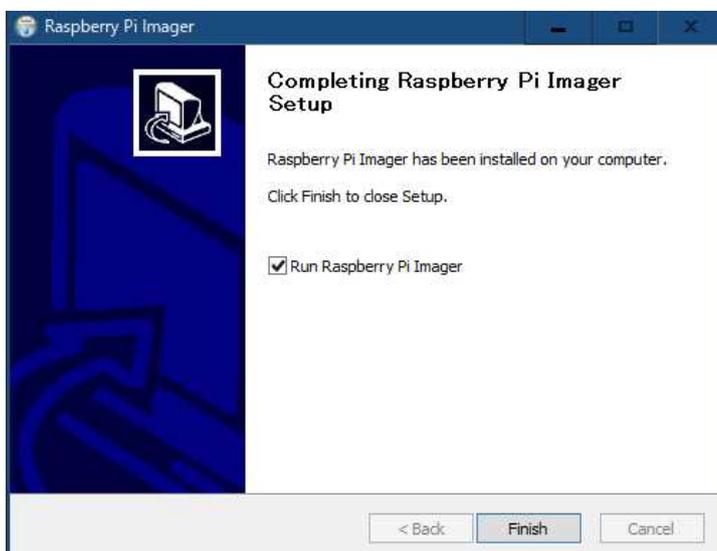


このような画面が出てきたら、「実行」をおします。

この画面が出たら、「Install」を押します。



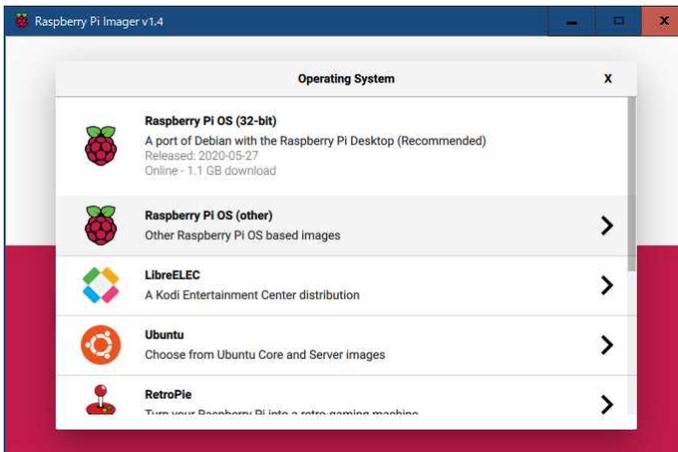
インストールが終わったら、「Finish」を押します。



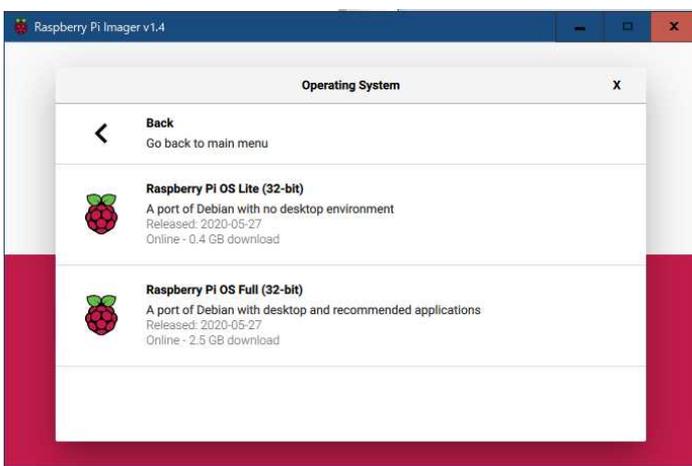
続いて、この画面が出来てきたら、「CHOOSE OS」をクリックして、



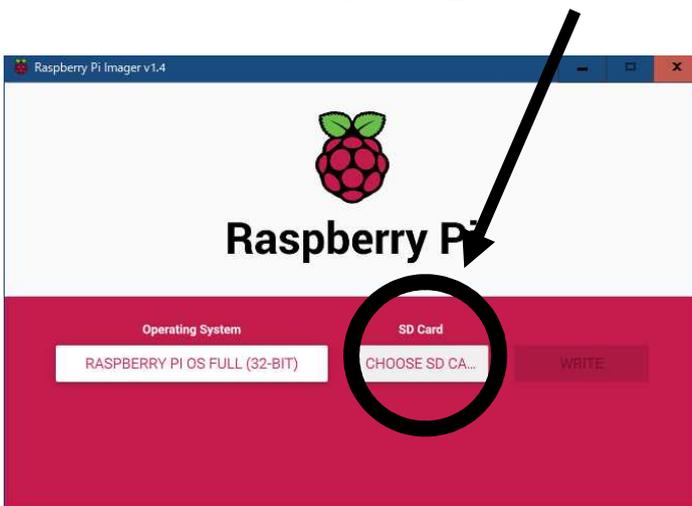
「Raspberry Pi OS(other)」を選びます。



インストールするOSのタイプを聞いてくるので、「Raspberry Pi OS Full(32-bit)」を選びます。



続いて、microSD カードを PC に差し、「SD Card」をクリックして該当するカードを選びます。

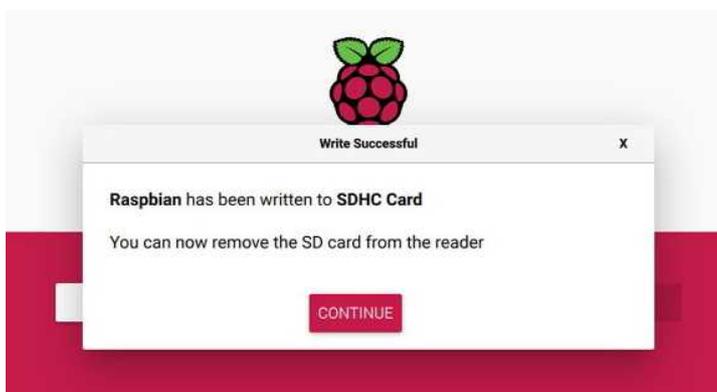


ここで、

書き込みが始まり、

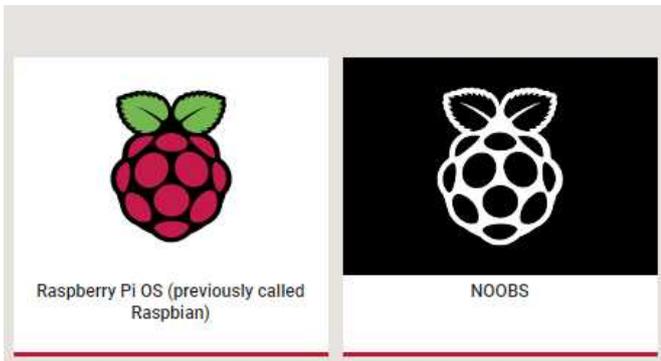


書き込みが終了します。

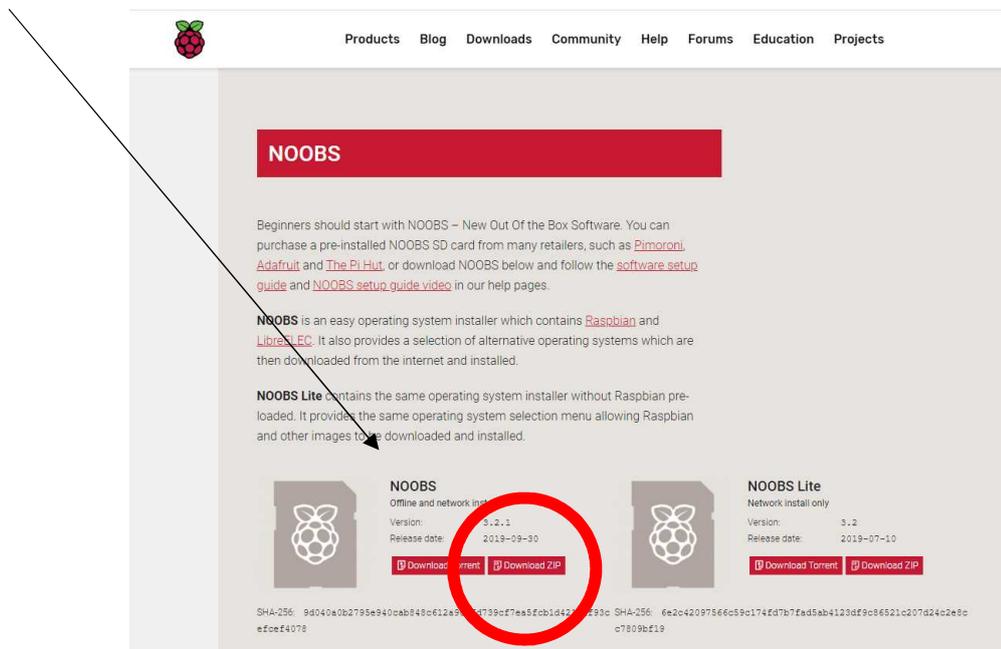


NOOBS を使ったダウンロードの方法

Raspberry Pi 公式サイト (<https://www.raspberrypi.org/>) を開き、Downloads をクリックします。
ページを下にスクロールすると、以下のアイコンがあるので、「NOOBS」を選んでください。



NOOBS の Download Zip をクリックします。



zip で圧縮されたファイルがダウンロードされます。

Downloads が終わったら解凍して展開します。

解凍ソフトは「7 - Zip (セブジップ)」を使います。7-Zip は Windows の解凍圧縮フリーソフトです。「7-Zip」で検索して、ダウンロードサイトからソフトをダウンロードし、実行してください。解凍をするとフォルダが作成されその中にファイルが保存されます。

解凍展開方法はパソコンの設定やブラウザによって異なります。通常はファイルを開けば自動的に展開されます。

OS を入れる microSD をパソコンに装着し展開したフォルダの中身をすべて microSD にコピーします。コピーするのはフォルダではなく、フォルダの中身です。

また、microSD のフォーマットを求められたら、以下の手順でフォーマットをします

(microSD が 32GB 以下の場合)

microSD のフォーマットされていない場合は、「ドライブ を使うにはフォーマットする必要があります。フォーマットしますか」というマイクロソフトウィンドウズの画面がでます。これは「キャンセル」を押して消します。

ラズベリーパイ公式サイトが推奨する「SDメモリカードフォーマッター」というソフトでフォーマットをします。「SDメモリカードフォーマッター」で検索してSD-Association (SDアソシエーション) のページから Windows 版をダウンロードしてください。ダウンロード後開くをクリックして実行します。途中でセキュリティー警告がでますが信頼できるソフトなので安心してインストールを続けてください。インストールが完了したという画面で「プログラムの起動」にチェックを入れておきます。

カードの選択の下にある更新を押して、表示されるカードを選びます。そして、「クイックフォーマット」にチェックをしてフォーマットをします。ボリュームラベルはどんな名前でもよいです。これでフォーマットができました。タスクバーからフォーマットしたデバイスの取り出しをクリックして「 はコンピュータから安全に取り外すことができます。」と表示されてからカードを抜いてください。

(microSD が 32GB より大きい場合)

microSD カードは容量によってデータの圧縮率が異なる形式でフォーマットされます。具体的には2 ~ 32GB までは FAT32 形式、32GB 以上は exFAT 形式でフォーマットされます。ところが、ラズベリーパイは exFAT フォーマットされたファイルが読めません。32GB より大きいカードの場合、「SDメモリカードフォーマッター」は exFAT でフォーマットしてしまいます。

そこで、64GB などのSDカードを FAT32 にフォーマットするために、「FAT32 フォーマットSD」で検索してください。「FAT32 Format」などフリーソフトがいくつかあります。ソフトをダウンロードして FAT32 でフォーマットしてください。

コピーが終わったら、アンマウント処理(「取り出し」「ハードウェアの安全な取り外し」)を行い、SDカードをパソコンから取り外します。

6 Raspberry Pi OS のインストールと初期設定

ラズベリーパイにキーボード・マウス・HDMI ケーブルなどを接続し、背面の microSD スロットに OS が入った SD カードを装着します。電源用の microUSB をつなぐと、赤い電源 LED が点灯しインストーラが起動します。

インストーラの言語を日本語にして、ラズビアンを選択してインストールアイコンをクリックしてインストールを開始します。インストールが完了して OK ボタンを押すと、再起動します。

NEXT を押します



NEXT を押します



国と言語とタイムゾーンを聞いてくるので、Japan・Japanese・Tokyo を選んで次に行きます。

パスワードを求められます。



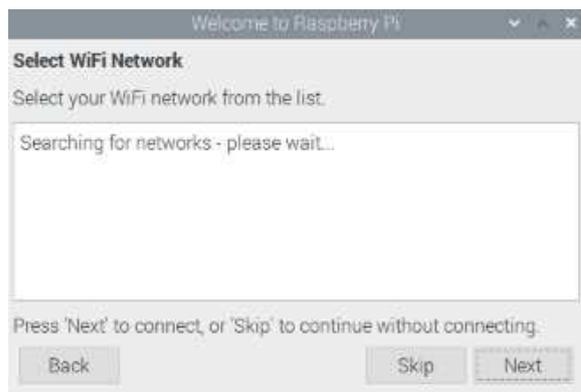
初期設定ではアカウント名が「pi」パスワードは「Raspberry」になっています。変更せずに次に行きます。もし、変更する場合は、他のPCとの通信などでよくアカウント名（ユーザー名）とパスワードを使うのでメモをするなどして忘れないようにしてください。

スクリーンの設定



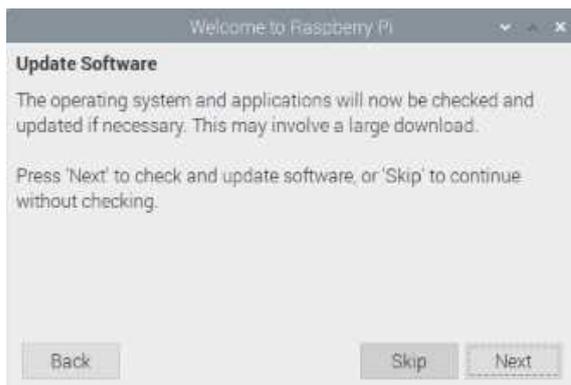
ディスプレイの端に黒い枠線が表示されている場合は、チェックをつけて次に行きます。

WiFi の設定

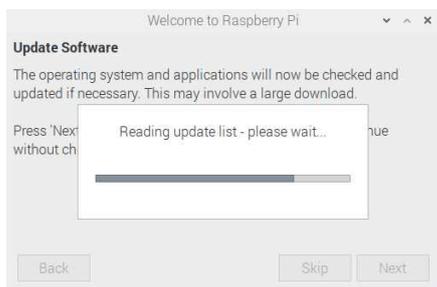


接続可能な無線 LAN の一覧が表示されるので、パスワードを入力して利用するネットワークの設定を行います。

ソフトウェアのアップデートを聞いてきます。アプリケーションの最新バージョンや日本語入力関係のパッケージなどがアップデートされるので、Next を押してアップデートをします。



アップデート中です

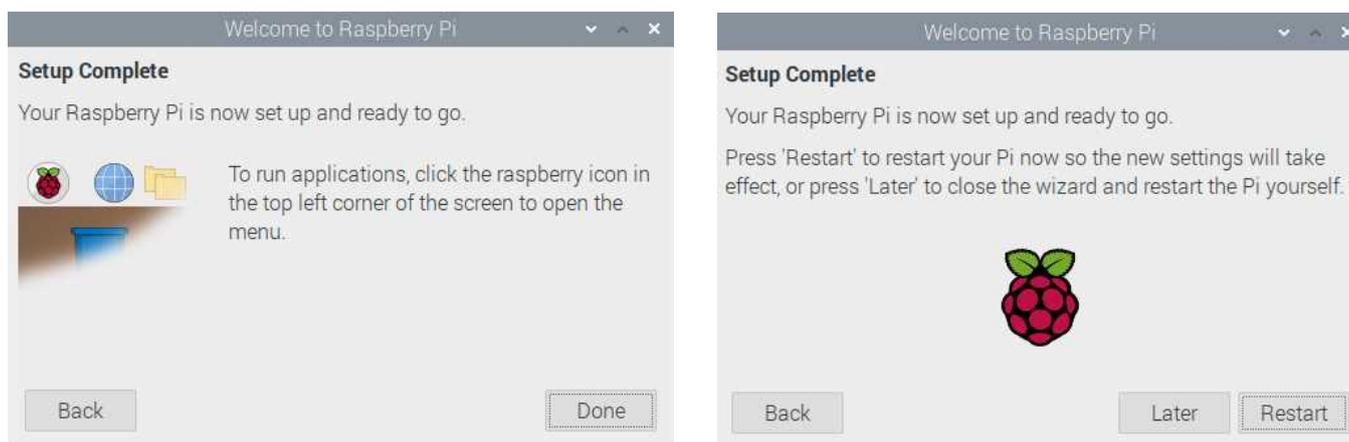


7 Raspberry Pi のシャットダウン

セットアップ完了。

1) 左の画面が出てきた場合は、「Done」を押してください。次に、左上のタスクバーから、一番左の「ラズパイ」のアイコンをクリックし、「メニュー」をだします。一番下の「shutdown (シャットダウン)」をクリックし、「reboot」を選んで再起動します。

2) 右の画面が出てきた場合は、「Restart」を押して再起動します。



再起動後、画面に文字化けがないか確認してください。文字化けがなければ、OSのインストールは無事終わりました。これでラズパイが使えるようになりました。

シャットダウンをする場合は、「shutdown」を押して終了させた後に電源を切ります。(電源プラグを抜きます。)

もし、デスクトップのゴミ箱アイコンの名前が「ゴミ箱」と日本語表示になっ
てなかつたり、「ラズパイ」のアイコンをクリックして出てくる「メニュー」に文字
化けが発生したりしている場合は、次の「8 日本語変換ツールのインストールと設
定」を行います。ほとんどの場合、次の作業は必要ありません。

8 日本語変換ツール（日本語入力メソッド）のインストールと設定

デスクトップのゴミ箱アイコンの名前が「ゴミ箱」と日本語表示になってなかったり、「ラズパイ」のアイコンをクリックして出てくる「メニュー」に文字化けが発生したりしている場合のみ次の作業を行います。ほとんどの場合、この作業は必要ありません。

上のタスクバーにアイコンから、一番右のL Xターミナルを起動します。



黒い画面が出てくるので、次のコマンドを入力しエンターキーを押します。

sudo apt update

<コマンドの説明>

sudo (superuser do) コマンド：コンピュータの管理者権限でプログラムを実行するときに使います。

apt (Advanced Package Tool) update：ラズパイのディストリビューションのパッケージ管理システムをAPTライブラリといいます。このAPTライブラリのリストを、最新バージョンにします。

```
pi@raspberrypi:~  
ファイル(F) 編集(E) タブ(T) ヘルプ(H)  
pi@raspberrypi:~ $ sudo apt update  
取得:1 http://archive.raspberrypi.org/debian buster InRelease [25.2 kB]  
取得:2 http://rasbian.raspberrypi.org/raspbian buster InRelease [15.0 kB]  
取得:3 http://archive.raspberrypi.org/debian buster/main armhf Packages [274 kB]  
取得:4 http://rasbian.raspberrypi.org/raspbian buster/main armhf Packages [13.0  
MB]  
13.3 MB を 22秒 で取得しました (609 kB/s)  
パッケージリストを読み込んでいます... 完了  
依存関係ツリーを作成しています  
状態情報を読み取っています... 完了  
アップグレードできるパッケージが 216 個あります。表示するには 'apt list --upgrad  
able' を実行してください。  
pi@raspberrypi:~ $
```

画面の内容はラズベリーパイによって異なります。

Update が終わったところです。

アップデートは、ライブラリのリストが最新になっただけで、ソフトウェアの更新 = アップグレードは行われていないことに注意してください。

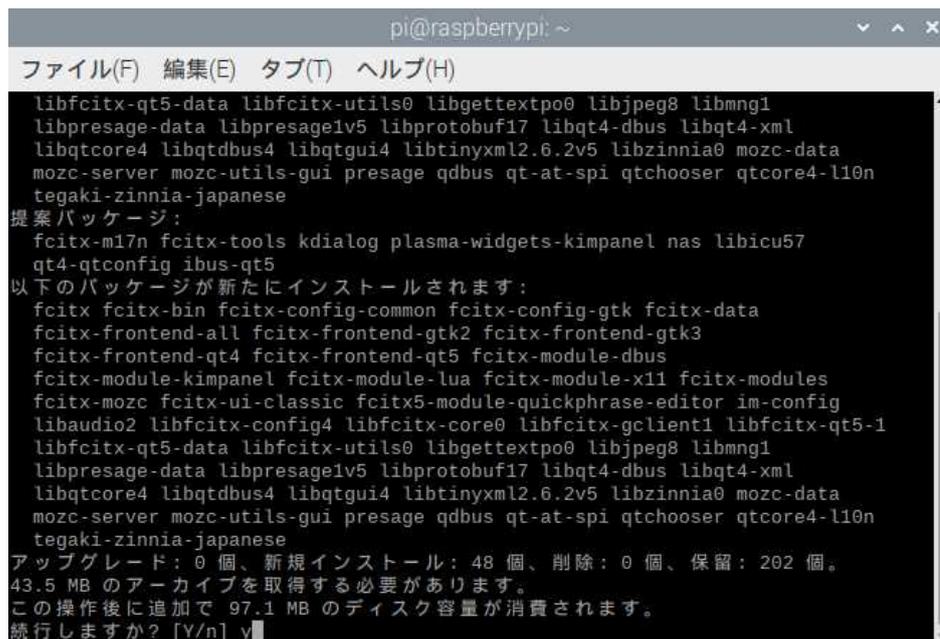
次に、以下のコマンドを入力しエンターキーを押します。

```
sudo apt install fcitx-mozc
```

< コマンドの説明 >

apt (Advanced Package Tool) install : APTライブラリリストの中から、fcitx-mozc をインストールします。

Fcitx-mozc (Flexible Input Method Framework-Mozc ファイティクスーモズク) : Fcitx は、リナックスの多言語サポートフレームワークです。Google 日本語入力のオープンソース版の Mozc (モズク) がベースの文字メソッドです。



```
pi@raspberrypi: ~
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
libfcitx-qt5-data libfcitx-utils0 libgettextpo0 libjpeg8 libmng1
libpresage-data libpresage1v5 libprotobuf17 libqt4-dbus libqt4-xml
libqtcore4 libqtdbus4 libqtgui4 libtinymce2.6.2v5 libzinnia0 mozc-data
morc-server mozc-utils-gui presage qdbus qt-at-spi qtchooser qtcore4-l10n
tegaki-zinnia-japanese
提案パッケージ:
fcitx-m17n fcitx-tools kdiallog plasma-widgets-kimpanel nas libicu57
qt4-qtconfig ibus-qt5
以下のパッケージが新たにインストールされます:
fcitx fcitx-bin fcitx-config-common fcitx-config-gtk fcitx-data
fcitx-frontend-all fcitx-frontend-gtk2 fcitx-frontend-gtk3
fcitx-frontend-qt4 fcitx-frontend-qt5 fcitx-module-dbus
fcitx-module-kimpanel fcitx-module-lua fcitx-module-x11 fcitx-modules
fcitx-mozc fcitx-ui-classic fcitx5-module-quickphrase-editor im-config
libaudio2 libfcitx-config4 libfcitx-core0 libfcitx-gclient1 libfcitx-qt5-1
libfcitx-qt5-data libfcitx-utils0 libgettextpo0 libjpeg8 libmng1
libpresage-data libpresage1v5 libprotobuf17 libqt4-dbus libqt4-xml
libqtcore4 libqtdbus4 libqtgui4 libtinymce2.6.2v5 libzinnia0 mozc-data
morc-server mozc-utils-gui presage qdbus qt-at-spi qtchooser qtcore4-l10n
tegaki-zinnia-japanese
アップグレード: 0 個、新規インストール: 48 個、削除: 0 個、保留: 202 個。
43.5 MB のアーカイブを取得する必要があります。
この操作後に追加で 97.1 MB のディスク容量が消費されます。
続行しますか? [Y/n] y
```

画面の内容はラズベリーパイによって異なります。

Yを入力してエンターキーを押します。

インストールが終わったら、再起動をします。



シャットダウンの中から reboot (再起動の意味) を選びます

次に、メニューから設定-Fcitx 設定を押します。



入力メソッドのオンオフの右側のボタンを押して、キーボードの半角/全角キーを押します。



これで日本語入力ができるようになります。

2 章 Raspberry pi OS の使い方

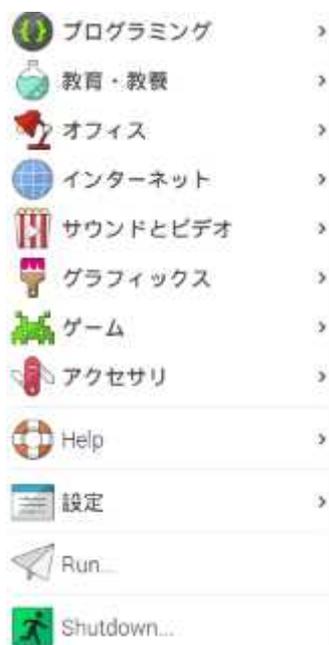
1 アプリケーション概要

< タスクバー >



	アプリケーションメニュー	アプリケーションのメニューが開きます
	ウェブブラウザ	ウェブサーバーに接続し、ウェブページなどを表示します
	ファイルマネージャー	ファイルの移動やコピー・削除・検索など、ファイルの操作や管理ができます
	LXターミナル	ラズパイのOSであるLinuxと対話をするソフトウェアです。コマンド(命令)を送ります。

< アプリケーションメニュー >



	プログラミング	プログラミング学習用や計算用のソフトがあります。
	教育・教養	スマートシムがあります。スマートシムでは電子回路の設計とシミュレーションができます。
	オフィス	Microsoft の Office と互換性のある、表計算やテキストエディタ、画像描画などが用意されています。
	インターネット	ウェブブラウザやメールソフトがあります。
	サウンドとビデオ	非常に多くのファイルの再生ができるフリーソフト、VLC メディアプレイヤーがあります。
	グラフィックス	画像表示用のソフトがあります。
	ゲーム	ゲームがあります。
	アクセサリ	計算機、ターミナル、PDFビューア、テキストエディタ、タスクマネージャなどがあります。

2 プログラミング



BlueJ Java IDE	Java の統合開発環境。オブジェクト構成が具体的に可視化できるため、オブジェクト指向プログラミングを学習するツールとして用いられています。
Geany	コンパイルが必要な C 言語や Java など、必要としないインタプリタ言語パイソンやパールなど、どちらの言語も書いて実行ができる、動作が軽いエディタです。
Mathematica (マスマティカ)	5000 以上の関数を利用できる世界最高峰の数式処理ソフト
Node-Red (ノードレッド)	IoT 機器の接続に適しています。Node と呼ぶ部品をつなげて様々な制御が直観的にできます。
Scratch 2, 3	初心者向けプログラミング学習ツール。GPIO の操作ができます。
Sense HAT Emulator	LED やセンサが組み込まれたボードを操作します。
Sonic Pi	音楽を作りながらプログラミング学習ができます。
Thonny Python IDE	Python の統合開発環境です
Wolfram	Mathematica を動かす言語です。
Mu	初心者用の Python エディタです

3 教育・教養



スマートシムは、電子回路の設計とシミュレーションができます。

4 オフィス



リブレオフィスの特徴

- 完全無料で全機能が使える。
- 何年たっても作成したデータを編集できる
- マイクロソフトのオフィス等と互換性がある。
- 非営利組織が開発・運営している。
- 誰でも利用できるし共有もできる。
- 100以上の言語に対応している。
- 動作が軽い

LibreOffice Base	データベースソフト
LibreOffice Calc	表計算ソフト。
LibreOffice Draw	ドローソフト
LibreOffice impress	プレゼンテーションソフト
LibreOffice Math	数式エディタ
LibreOffice Writer	ワープロソフト

リブレオフィスは、英語表記になっています。これを日本語化して使いやすくします。

アプリケーションメニューから、「設定」を選び、「Add/Remove Software」(ソフトウェアの追加と削除という意味)を選びます。

下の画面が出ます。



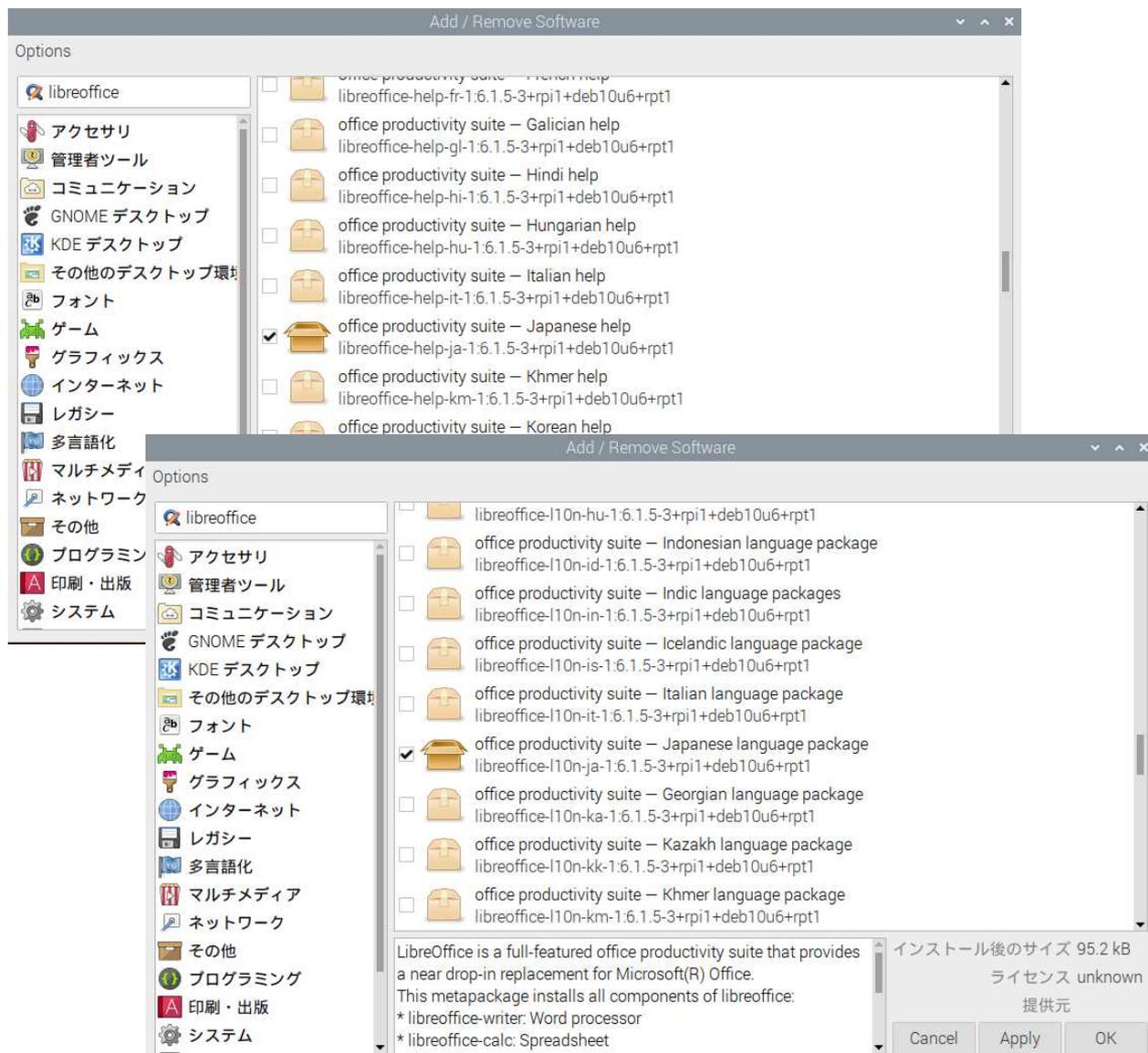
左上の検索画面に「libreoffice」と打って、エンターを押してください。



そうすると、「libreoffice」に関係するソフトを検索します。

検索が終わったら、リストから「office productivity suite - Japanese help」と「office productivity suite - Japanese language package」を探して、チェックをつけてください。

チェックをつけたら、右下の「OK」を押してください。インストールが始まります。



インストールが終わったら、「libreoffice」の「LibreOffice Calc」などを立ち上げてください。メニューが日本語で表記されています。

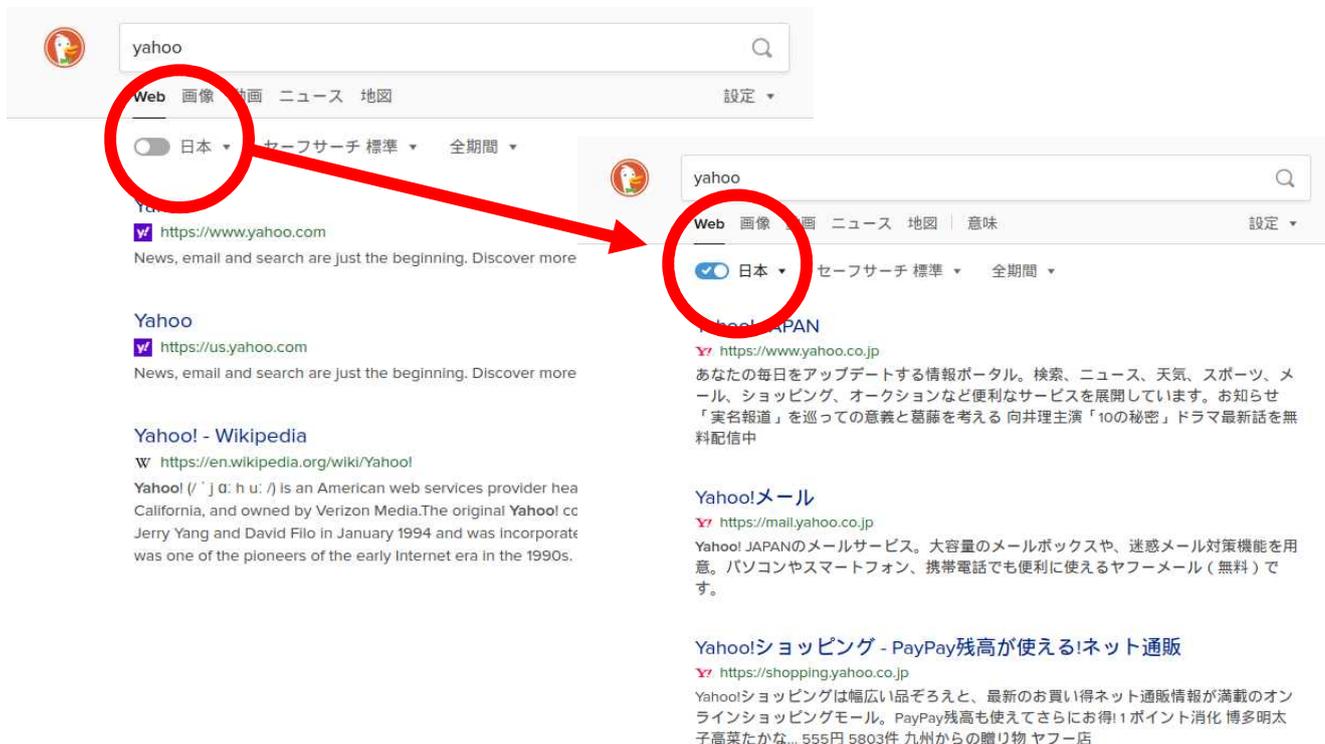
5 インターネット



Chromium (クロミウム) は、オープンソースのウェブブラウザです。Google Chrome (クローム) はクロミウムの一部を使って開発されています。

検索を「yahoo」で行ってください。検索地域の初期設定が全世界になっているのでアメリカの yahoo が検索されます。日本のサイトから検索をする設定にすると日本のサイトが検索できます。

下のように (フリップ) スイッチを ON にします。



6 設定



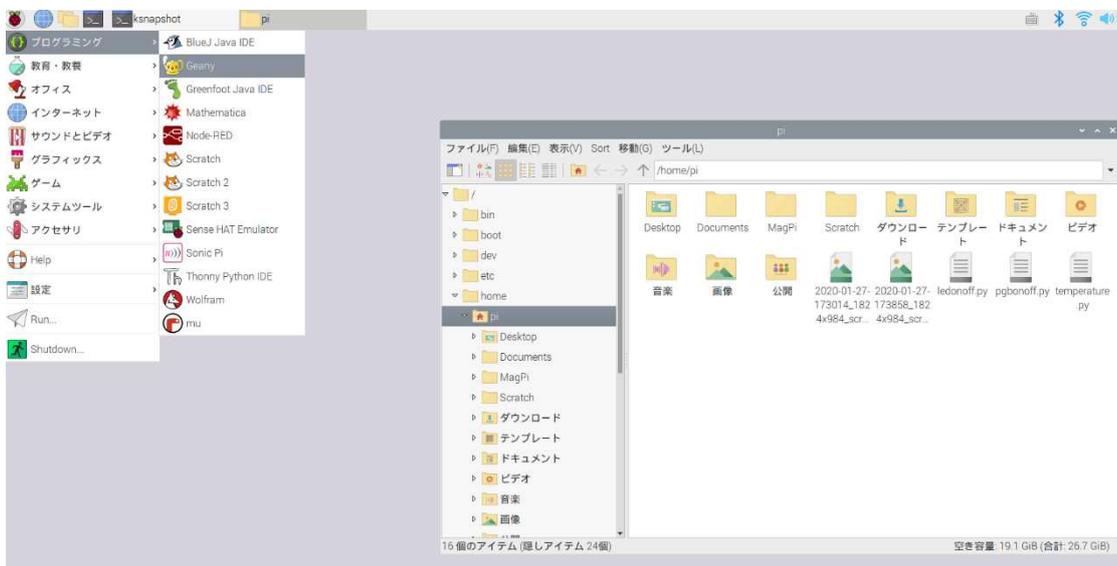
Add/Remove Software	アプリケーションのインストールを行います。
Appearance Settings	デスクトップ、メニューバー、システムの設定をします。
Audio Device Settings	オーディオデバイスの設定をします。
KDE Connect Settings	ラズパイと android スマートフォン・タブレットをペアリングします。
Main Munu Editor	メインメニューを編集します。
Raspberry Pi の設定	ラズパイのシステム、インターフェイス、パフォーマンスなどの設定をします。
Recommended Software	ラズパイのおすすめソフトが表示されます。インストール済のソフトはチェックが付きます。
キーボードとマウス	キーボードとマウスの設定をします。

7 GUIとCLI

ラズベリーパイは、インターネットの利用や他のPCと互換性のあるビジネスソフトも使え、センサなどの電子機器を制御することもできます。しかし、一般的なパソコンと比較すると処理能力と速度は劣ります。Windows や MacOS などと同じようにGUI (Graphical User Interface) を使うことはできませんが、どうしても処理に時間がかかってしまいます。そこで、動作の軽いCLI (Command Line Interface) も使えるようになっています。ラズビアンはLinuxなどをベースにしているので、使用するコマンドはLinuxのものになります。

< GUI >

ディスプレイ上に、ウインドウ、メニュー、ボタン、アイコンなどの画像(グラフィック)が表示され、ユーザーは必要な操作を行うために、画像(グラフィック)をマウスなどのポインティングデバイスやタッチパネルをクリックするインターフェース環境のことで、今のパソコンがこの形式です。直観的に操作ができ便利ですが、ラズベリーパイでは動作が遅くなる場合があります。



メニューをクリック、マウスを上を持っていったりすると、新しいウインドウが展開し、クリックして操作を選べるようにしています。また、ウインドウにフォルダや文書や画像を表すアイコンを表示させて、それらをクリックするとファイルが開いたりプログラムが起動したりします。

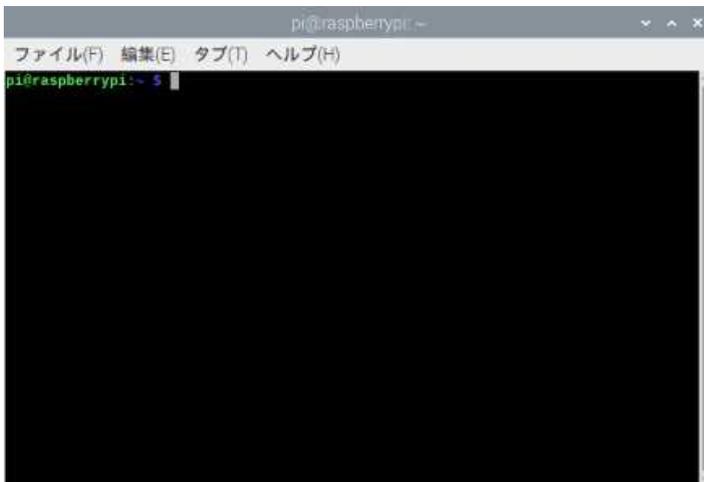
< CLI >

CLIを使うとディスプレイに表示させる処理が大幅に軽減されるので動作が軽くなります。特に新しいソフトウェアパッケージをインストールするときなどによく使います。Pythonのプログラミングでも使用したシェル(プロンプト)がCLIです。(初期設定では背景が黒ですが、背景色は設定で変えることもできます)

8 Raspberry pi の操作

< L Xターミナル >

左上のタスクバーから、L Xターミナルを起動してください。



pi@raspberrypi : ~ \$ █

と表示されています。右端は灰色の長方形があり、コマンド（命令）の入力を待っている状態です。

最初の pi はユーザー名です。

@の次は、今、動いているコンピュータ（ラズパイ）の名前です。これをホスト名と言います。

~は、作業をしているフォルダ（カレントフォルダ）が、ユーザーのホームフォルダであることを表します。

\$は、\$の次にコマンドを入力するという印です。プロンプト（促進・促す）と言います。

piから\$までの文字列をコマンドプロンプトと言います。コマンド（命令）を待っている状態です。

灰色の長方形の部分にコマンドを入力します。

なお、この説明ではイメージしやすくするために「フォルダ」と呼んでいますが、Linux では「ディレクトリ」と呼ぶことが多いことを覚えておいてください。

では、ls (エル・エス) とコマンドを入力してください。

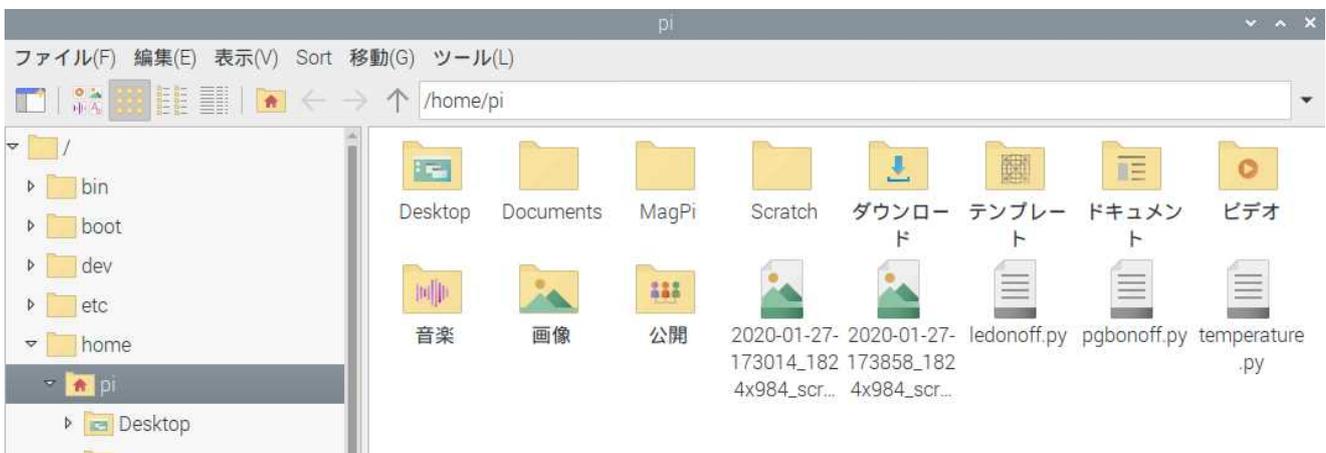
ls コマンドは、カレントフォルダにあるファイルのリストを表示させるコマンドです。



```
pi@raspberrypi: ~  
ファイル(F) 編集(E) タブ(T) ヘルプ(H)  
pi@raspberrypi:~ $ ls  
2020-01-27-173014_1824x984_scrout.png Desktop MagPi ledonoff.py temperature.py テンプレート ビデオ 画像  
2020-01-27-173858_1824x984_scrout.png Documents Scratch pgbonoff.py ダウンロード ドキュメント 音楽 公開  
pi@raspberrypi:~ $
```

画面の内容はラズベリーパイによって異なります。

ここで表示されたファイルは、左上のタスクバーのファイルマネージャで表示させたものと一致していることがわかります。



次に、pwd (ピー・ダブリュー・ディー) とコマンドを入力してください。

pwd (positioning working directory) コマンドは、カレントフォルダ (作業フォルダ) の場所 (パス) を表示させるコマンドです。



```
pi@raspberrypi:~ $ pwd  
/home/pi  
pi@raspberrypi:~ $
```

/home/pi と返してきました。

上のファイルマネージャに表示されているフォルダの位置と一致しています。

(Linux の主要コマンド例)

コマンド		用途
ls	List segments	フォルダ内の内容表示
mkdir	Make directory	フォルダの新規作成 [例 : mkdir newfile2020]
cp A B	Copy	A を B の名前を付けてコピー [例 : cp A.txt B.txt]
rm	Remove	ファイル削除
chmod	Change mode	ファイルの読み・書き・実行の制御をする
find	Find	ファイルを探す
grep	UNIX の g/re/p コマンドの略	文字列を探す
cat	Comcatenate (連結)	ファイルの内容をすべて表示
sudo	Superuser do	管理者権限でコマンドを実行する
pwd	positioning working directory	現在の作業フォルダの場所

9 アプリケーションの追加と削除 (CLI 環境)

ラズパイOSは、アプリケーションをパッケージで配布しています。多くのアプリケーションを無料で使えますが、ラズベリーパイの microSD メモリ容量が限られているためアプリケーションの管理が必要になってきます。アプリケーションパッケージを操作するコマンドは apt (Advanced Packaging Tool) コマンドです。この項では、よく使う apt コマンドについて説明をします。

apt update (パッケージ情報の更新)

sudo apt update

パッケージのインストールを行うときに、サーバー上にあるパッケージの倉庫 (リポジトリ) から、パッケージのバージョンとパッケージ間の依存関係を取得します。パッケージ間の依存関係というのは、例えば「A ソフトをインストールするためには B ソフトと C ソフトがインストールされてないといけない」などの情報です。このコマンドを実行すると最新のパッケージ情報が「/var/lib/apt」の中に更新保存されます。そして、インストールを行うとき参照にします。

このコマンドは管理者権限で行うので最初に sudo (superuser do) を付けます。

apt upgrade (パッケージの更新)

sudo apt upgrade

パッケージを更新します。

このコマンドは管理者権限で行うので最初に sudo を付けます。

apt search (パッケージの検索)

apt search “キーワード”

例えば、game や japan などのキーワードを含むパッケージを検索します。検索から得たパッケージ名を使ってインストールや削除などに利用します。

apt show (パッケージの詳細表示)

apt show パッケージ名

パッケージのバージョン、サイズ、依存するパッケージ、パッケージの説明などが表示されます。

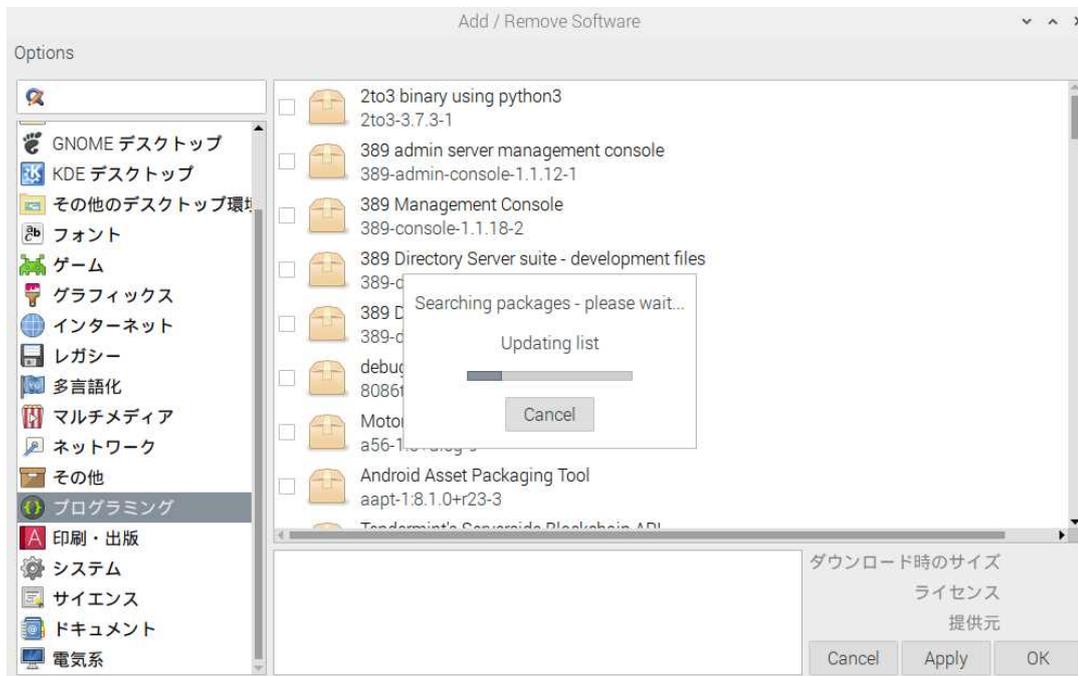
apt remove (パッケージの削除)

sudo apt remove パッケージ名

パッケージを削除します。このコマンドは管理者権限で行うので最初に sudo を付けます。

10 アプリケーションの追加と削除 (GUI 環境)

「メニュー」「設定」「Add/Remove Software」からでもパッケージの検索、インストール、削除が行えます。パッケージ情報の更新はオプションから選びます。ただし、コマンドで操作する場合と比べると、グラフィック処理をするためかなり時間がかかります。

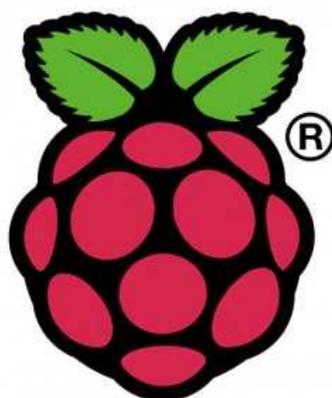


未来を創るリーダー育成

IoT 基礎

Sense Hat

センサ制御



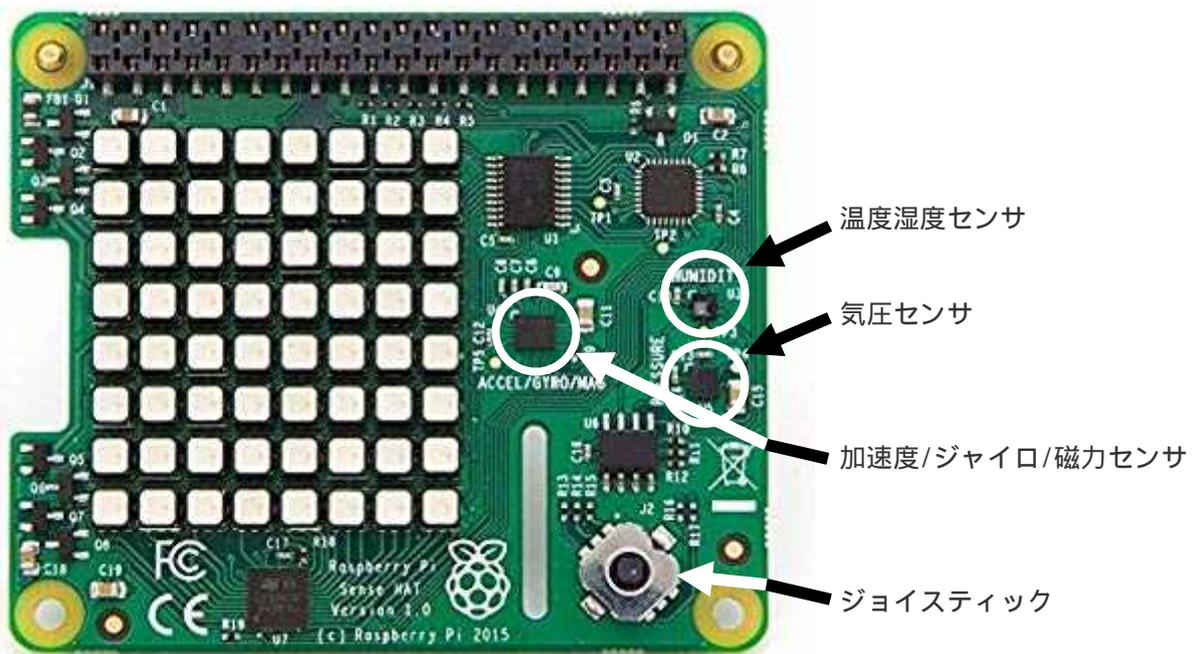
目 次

1 章	Sense Hat の導入	
1	Sense Hat とは	1
2	Sense Hat 用ライブラリのインストール	2
3	Python で Sense Hat を動かす(show message)	3
4	show message のパラメータ	7
5	演習	9
2 章	LEDディスプレイの操作	
1	一つの文字の表示	12
2	一つの文字の連続表示	13
3	一つの数字の表示	15
4	演習	17
5	イメージの表示	18
6	イメージの回転・反転	21
3 章	センサデータの取得	
1	気圧・温度・湿度センサデータの取得	23
2	姿勢動作方位データの基礎	24
3	姿勢動作方位データの取得	27
4 章	カメラ画像データの取得	
1	カメラモジュールの取り付けと設定	30
2	カメラの操作と撮影	32
3	カメラアプリの利用	36

1 章 Sense HAT の導入

1 Sense HAT とは

Sense HAT は、「Astro Pi」という宇宙ステーションでラズベリーパイを動かそうというプロジェクトのために作られた、ラズベリーパイの上に取り付けるセンサボードです。Sense Hat は、8×8 のRGB 3色 LED、前後左右下の5方向のジョイスティックと、ジャイロスコープ（角速度計）、アクセラレーター（加速度計）、マグネットメータ（磁力計）、温度計、気圧計、湿度計が組み込まれています。



Sense Hat をラズベリーパイに取り付け、キーボードとマウス、ディスプレイに接続をしてください。

電源を入れるとLEDが虹色に光ります。もし、LEDが光らない場合は、電源に異常がないか、取り付けが正常であるかを確認してください。また、他のラズベリーパイに装備して動作を確認するなど確認をしてください。



2 Sense HAT 用ライブラリのインストール

Raspberry pi OS に専用のライブラリが用意されているのでインストールします。

LX ターミナルを起動し、

```
sudo apt install sense-hat
```

を入力してください。パッケージリストの作成しインストールされているかどうかを調べます。インストールされてなければ、最新バージョンを自動でインストールします。

A terminal window titled 'pi@raspberrypi: ~' with a menu bar containing 'ファイル(F)', '編集(E)', 'タブ(T)', and 'ヘルプ(H)'. The terminal output shows the command 'sudo apt install sense-hat' being executed. The output text is: 'パッケージリストを読み込んでいます... 完了', '依存関係ツリーを作成しています', '状態情報を読み取っています... 完了', 'sense-hat はすでに最新バージョン (1.2) です。', 'アップグレード: 0 個、新規インストール: 0 個、削除: 0 個、保留: 216 個。', and finally 'pi@raspberrypi:~ \$' with a cursor.

```
pi@raspberrypi:~ $ sudo apt install sense-hat
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています
状態情報を読み取っています... 完了
sense-hat はすでに最新バージョン (1.2) です。
アップグレード: 0 個、新規インストール: 0 個、削除: 0 個、保留: 216 個。
pi@raspberrypi:~ $
```

画面はラズパイによって変わります。この画面はすでに最新バージョンがインストールされていた例です。

3 Python で Sense HAT を動かす (show message)

「メニュー」 - 「プログラム」 - 「Thonny Python IDE」を選び起動してください。

プログラムの中に「Thonny Python IDE」がない場合は、
「メニュー」 - 「設定」 - 「Main Menu Editor」を選び、Main Menu Editor を起動しま
す。

一番左の「アプリケーション」の中から「プログラミング」を選び、右に出てくる表示内容
の中から「Thonny Python IDE」にチェックをつけて「OK」を押してください。



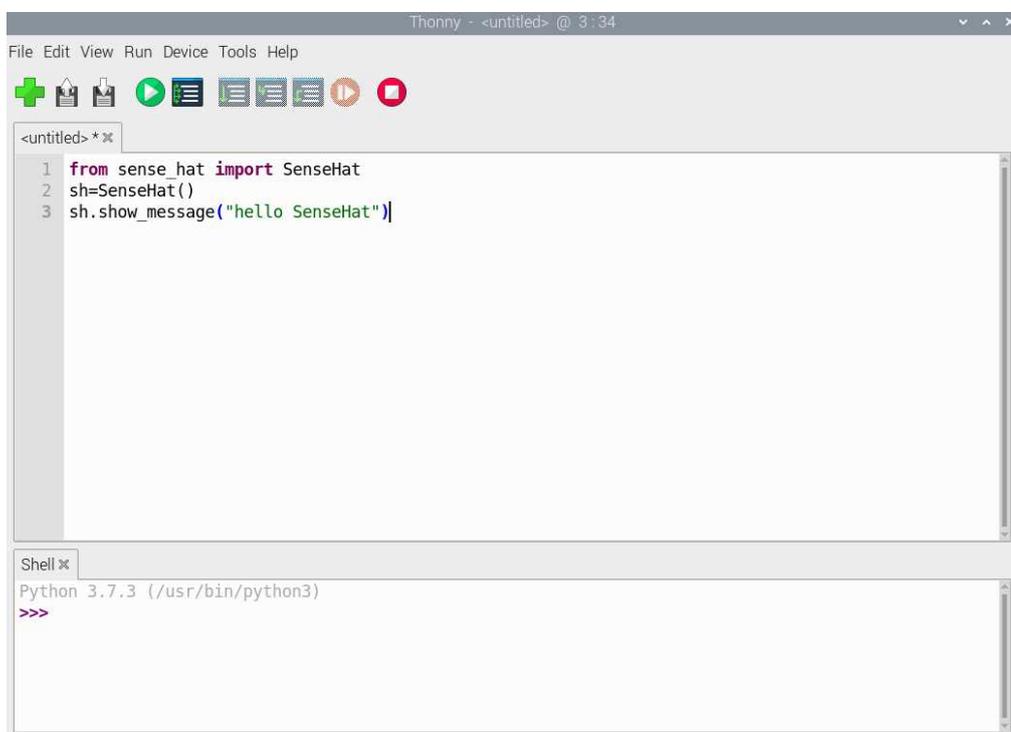
Thonny Python IDE は、プログラミング初心者向けに作られたパイソンの開発ツールです。起動すると上下に2つの画面が現れます。上の画面にソースを書き、「RUN」ボタンを押します。そうするとファイルの保存の画面が立ち上がります。ファイル名を記入すると保存が完了し、下のシェル画面に実行結果が表示されます。もちろん、ソースを書いたファイルを保存しておき、ファイルを呼び出して実行させることもできます。

上の画面に以下のソースを書いてください。

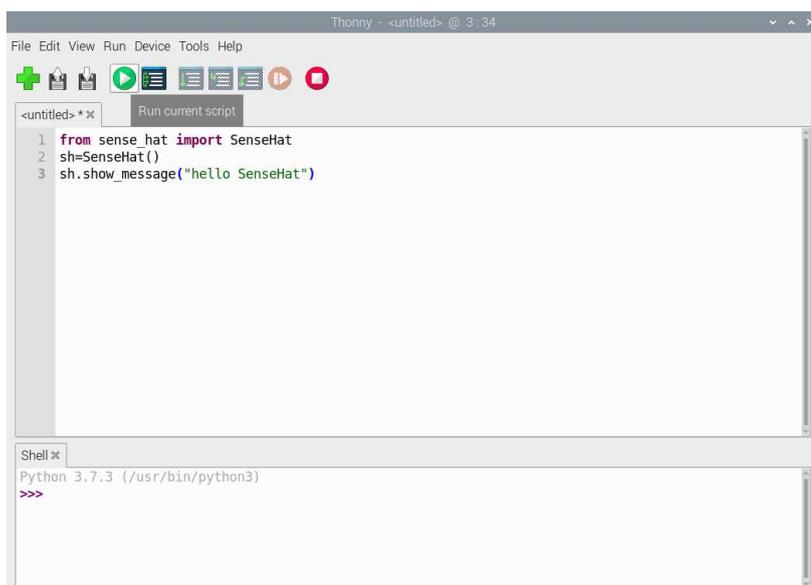
```
from sense_hat import SenseHat

sh=SenseHat()

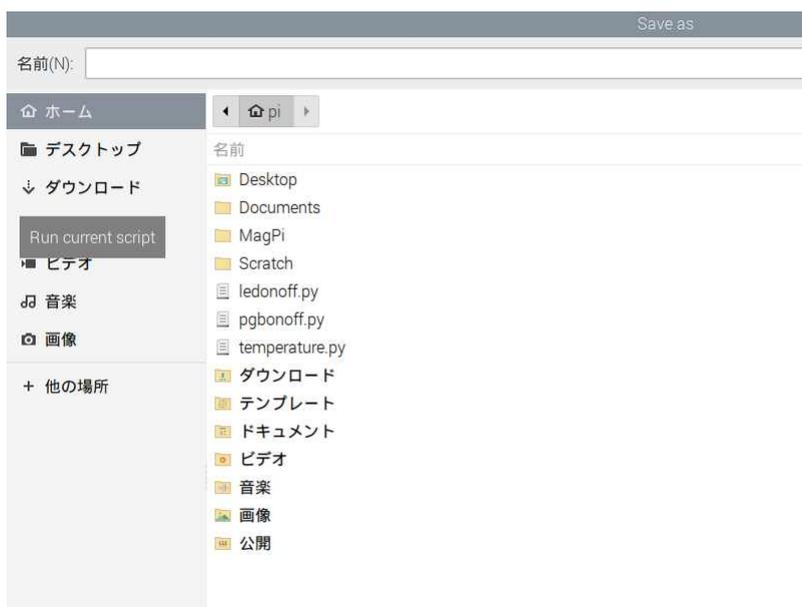
sh.show_message( " Hello SenseHat " )
```



次に、上の緑の「RUN(Run current script : 今あるファイルに書かれているソースコードを実行)」ボタンを押します。



打ち込んだソースを保存する画面が立ち上がります。



「Hello_SenseHat」と打ち、OKを押してください。Hello_SenseHat.pyのファイルが保存されます。保存と同時にソースが実行されLEDに文字が流れます。

(ソースの解説)

```
from sense_hat import SenseHat
```

sense_hat のパッケージから、SenseHat モジュールを呼び出します。

モジュールは「関数」や「クラス」を集めたファイルです。パッケージはモジュールを集めたものです。

「関数」は、データの操作・処理方法の「指示書」。「クラス」は、複数の「関数」の操作・処理方法をまとめた「設計図」が入っているものです。

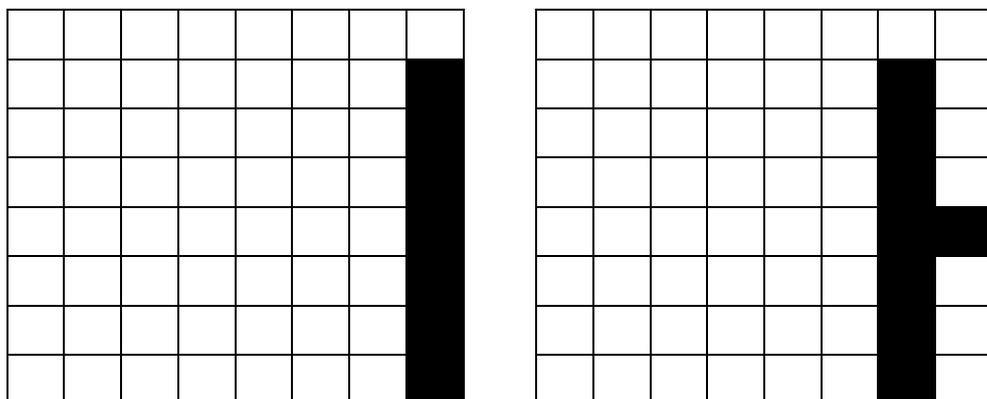
```
sh = SenseHat()
```

SenseHat モジュールの中にあるクラス SenseHat() をオブジェクト sh に保存します。

ここで、変数とオブジェクトの違いを説明します。sh に代入するのが「数字」や「文字」のみだけのときは「変数」、sh に代入するのが数字や文字が混在したものや関数などのときはオブジェクトになります。この場合は、sh に代入されるのが関数なのでオブジェクトになるわけです。

```
sh.show_message(" Hello SenseHat ")
```

show_message 関数は、メッセージを右から左に流す関数です。sh に収めた SenseHat() クラスの中の show_message 関数に「Hello SenseHat」という文字列を引数(パラメータ)として渡しています。show_message 関数は、もし H の文字が書かれている場合は、右から左に流れる文字を LED で表現するため、一番右の列の上から 2 ~ 8 番目を点灯させ、次に一番右の上から 5 番目と右から 2 番目の列 2 ~ 8 番目を点灯させ、次に・・・というようにプログラムされています。



他の文字についてもこの動作を連続して行い、あたかも文字が流れているように動作させています。街で見かける電光掲示板はこのような仕組みで動いています。

4 show message のパラメータ

(文字列以外のパラメータ(引数)の設定)

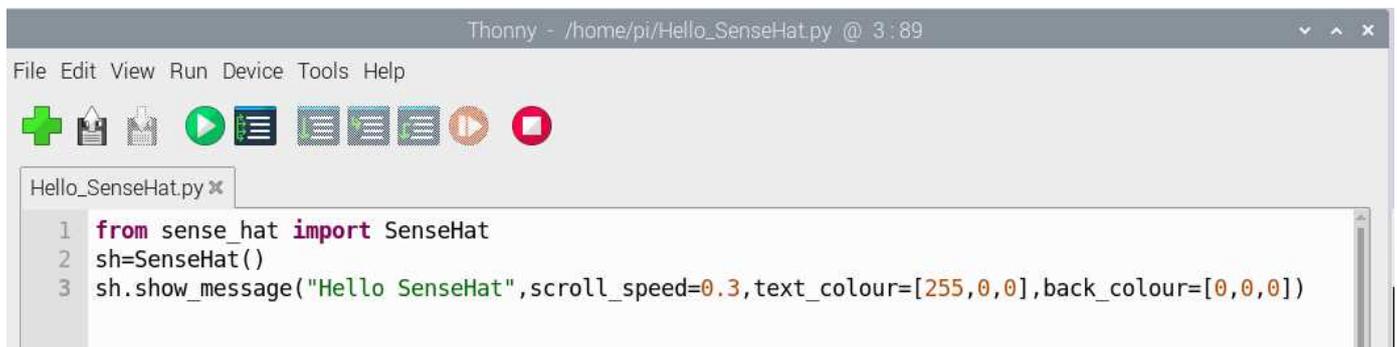
show_message 関数は文字列以外に、文字が流れる速度(スクロールスピード)、文字の色(テキストカラー)、背景の色(バックカラー)があります。

「Hello_SenseHat」ファイルの("Hello SenseHat")の中にソースを追加して実行してください。

(" Hello SenseHat " , scroll_speed=0.3 ,

text_colour=[255,0,0] , back_colour=[0,0,0])

colour のつづりに注意してください。カラーは米語(アメリカ英語)では color ですが、英語(イギリス英語)では colour です。



The screenshot shows the Thonny IDE interface. The title bar reads "Thonny - /home/pi/Hello_SenseHat.py @ 3:89". The menu bar includes "File", "Edit", "View", "Run", "Device", "Tools", and "Help". Below the menu bar is a toolbar with icons for opening files, saving, running, and stopping. The main editor window shows the following Python code:

```
Hello_SenseHat.py ✕
1 from sense_hat import SenseHat
2 sh=SenseHat()
3 sh.show_message("Hello SenseHat",scroll_speed=0.3,text_colour=[255,0,0],back_colour=[0,0,0])
```

(ソースの解説)

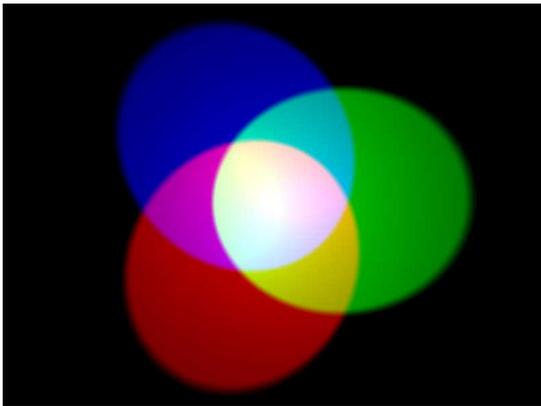
scroll_speed=0.3

文字が流れるスピードを秒単位で設定します。初期値(デフォルト)は 0.1 (秒)です。0.3 を設定すると、0.3 秒ごとに点灯する LED の位置が左に移動するようになります。

```
text_colour=[255,0,0]
```

```
back_colour=[0,0,0]
```

テキスト背景の色を設定します。ディスプレイ上の色は光の三原色RGBで表します。赤（RED）と緑（GREEN）と青（BLUE）の光の強さをそれぞれ0～255の256段階で表します。[255,0,0]は、R=255、G=0、B=0なので、赤が表示されます。初期値はテキスト白、背景黒です。



色名	R	G	B
黒	0	0	0
赤	255	0	0
緑	0	255	0
青	0		255
黄（イエロー）	255	255	0
空色（シアン）	0	255	255
赤紫（マゼンタ）	255	0	255
白	255	255	255

（練習）パラメータの文字や値を変えて動作を確認してください。

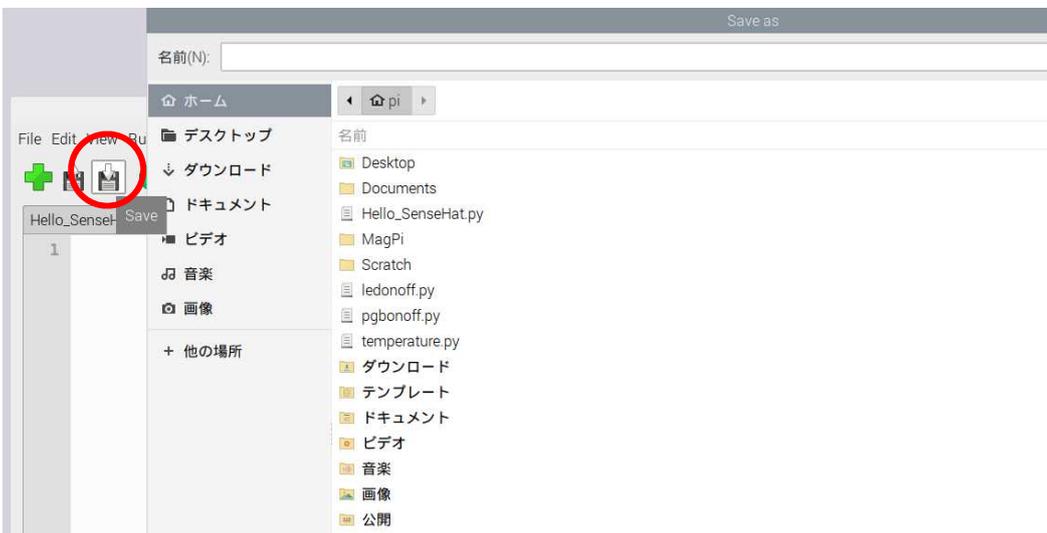
5 演習 次の要領に従って作業をしてください。

<Level 1>

新しいファイルを作成してください。（緑の十字アイコンを押すと新しいファイルが作成できます。）



新しいファイルの名前を「Hello_SenseHat2」にします。（左から3番目のアイコンが file メニューの save as をクリックして、保存画面を出し名前を記入して、OKを押します。）



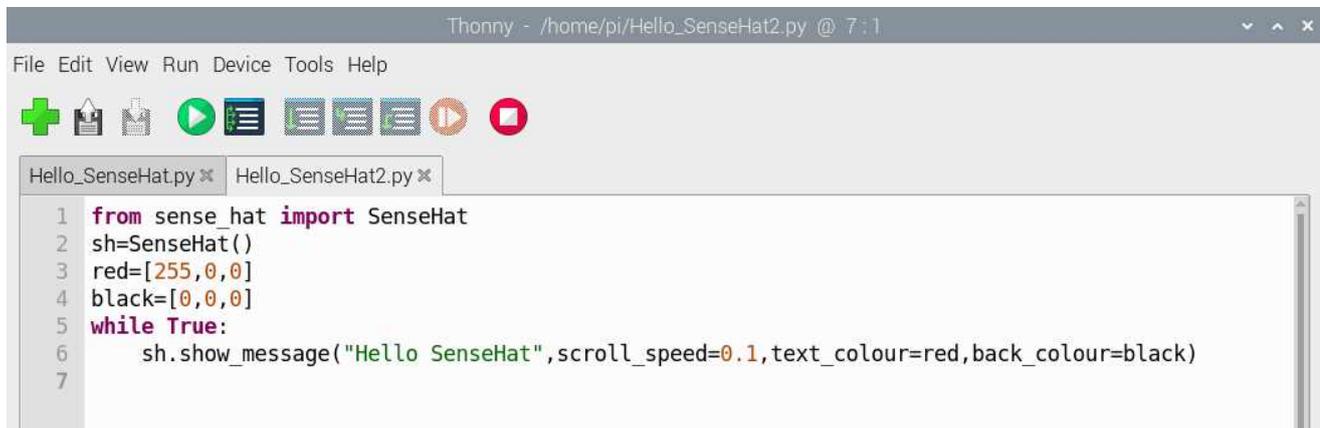
Hello_SenseHat.py のタブをクリックし、ソースをコピーします（「Ctrl + A」で全範囲選択した後、「Ctrl + C」でコピーすると便利です。）。次に新しく作成した Hello_SenseHat2.py のタブをクリックして貼り付けます（Hello_SenseHat2 の画面のどこかをクリックしてカーソルを点滅させてから「Ctrl + V」で貼り付けると便利です。）

sh.show.message の前に、色のオブジェクトを作ります。
red を[255,0,0]に、black を[0,0,0]にします。

```
red=[255,0,0]
```

```
black=[0,0,0]
```

sh.show.message の色番号の部分に で作ったオブジェクトを使用します。



```
Thonny - /home/pi/Hello_SenseHat2.py @ 7:1
File Edit View Run Device Tools Help
Hello_SenseHat.py x Hello_SenseHat2.py x
1 from sense_hat import SenseHat
2 sh=SenseHat()
3 red=[255,0,0]
4 black=[0,0,0]
5 while True:
6     sh.show_message("Hello SenseHat",scroll_speed=0.1,text_colour=red,back_colour=black)
7
```

<Level 2>

新しいファイルを作成し「Hello_SenseHat3」にします。「Hello_SenseHat2」のソースを「Hello_SenseHat3」にコピーしてください。

速さを 0.1 にして、表示する文字を「"Good!"」にします。さらに、yellow のオブジェクトを作成してテキストの色にします。

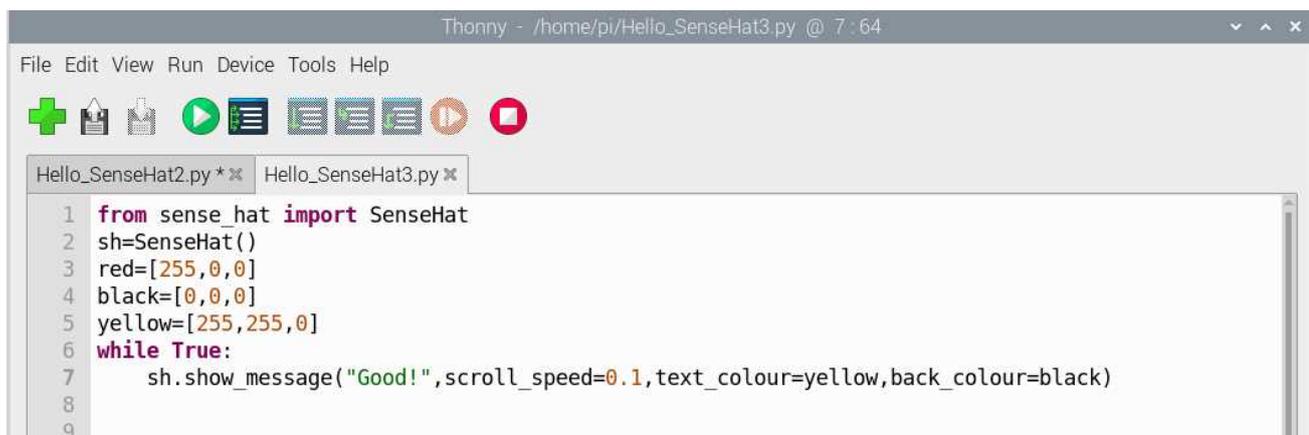
さらに、while を使って無限にループさせてください。

```
while True:
```

```
    sh.show_mes . . . . .
```

while True の後のコロんと、次の段のインデントを忘れないように。

上の赤の のアイコンを押すとプログラムがストップ（強制終了）します。



```
Thonny - /home/pi/Hello_SenseHat3.py @ 7:64
File Edit View Run Device Tools Help
Hello_SenseHat2.py *x Hello_SenseHat3.py x
1 from sense_hat import SenseHat
2 sh=SenseHat()
3 red=[255,0,0]
4 black=[0,0,0]
5 yellow=[255,255,0]
6 while True:
7     sh.show_message("Good!",scroll_speed=0.1,text_colour=yellow,back_colour=black)
8
9
```

<Level 3>

新しいファイルを作成し「Hello_SenseHat4」にします。「Hello_SenseHat3」のソースを「Hello_SenseHat4」にコピーしてください。

ループを 3 回繰り返したら終了するプログラムを書いてください。

プログラム例

```
i=0
while i<3:
    sh.show_mes . . . . .
    i=i+1      i+=1 でも可。
```



```
Thonny - /home/pi/Hello_SenseHat4.py @ 10:1
File Edit View Run Device Tools Help
Hello_SenseHat2.py *x Hello_SenseHat3.py x Hello_SenseHat4.py x
1 from sense_hat import SenseHat
2 sh=SenseHat()
3 red=[255,0,0]
4 black=[0,0,0]
5 yellow=[255,255,0]
6 i=0
7 while i<3:
8     sh.show_message("Good!",scroll_speed=0.1,text_colour=yellow,back_colour=black)
9     i=i+1
10
11
```

2 章 LEDディスプレイの操作

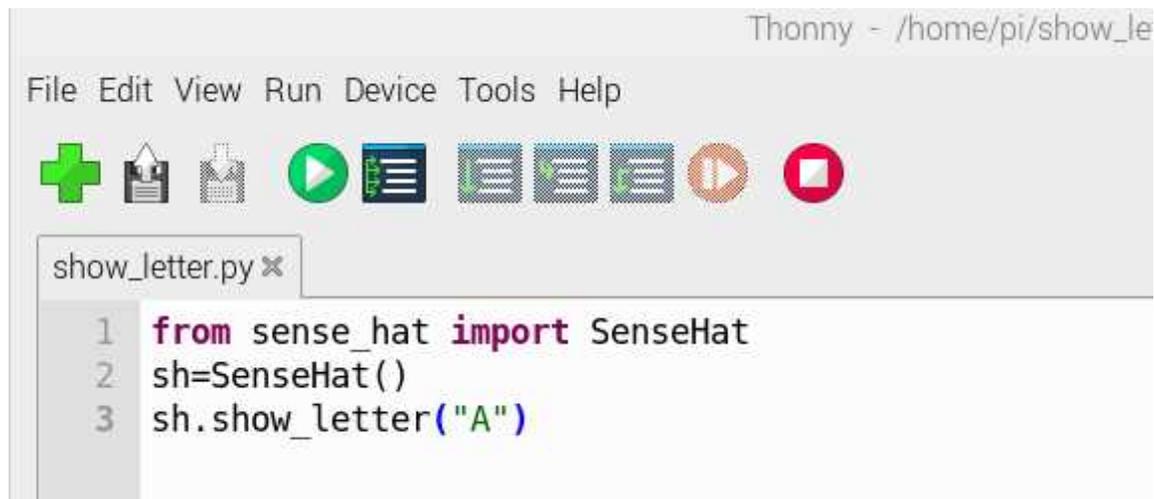
1 一つの文字の表示

1 章では文字をスクロールさせる `show.message` 関数を使いました。文字を表示する関数は、`show.letter()`です。

新しいファイル「`show_letter`」を作成してください。
以下のソースを書きます。

```
from sense_hat import SenseHat  
  
sh=SenseHat()  
  
sh.show_letter( " A " )
```

`show_letter` 関数には、テキストカラーと背景色の2つのパラメータしかありません。
(`show_message` 関数にあったスクロール速度は、スクロールをしないのでありません。)



The screenshot shows the Thonny IDE interface. The title bar reads "Thonny - /home/pi/show_le". The menu bar includes "File", "Edit", "View", "Run", "Device", "Tools", and "Help". Below the menu bar is a toolbar with icons for file operations and execution. The main editor window shows a file named "show_letter.py" with the following code:

```
1 from sense_hat import SenseHat  
2 sh=SenseHat()  
3 sh.show_letter("A")
```


次のソースで、文字の色を変えることもできます。また、Cを2秒表示させた後 sh.clear() をすると、SenseHat()モジュールが終了するので、Cの表示が消えます。

```
from sense_hat import SenseHat
from time import sleep
sh=SenseHat()

red=[255,0,0]

green=[0,255,0]

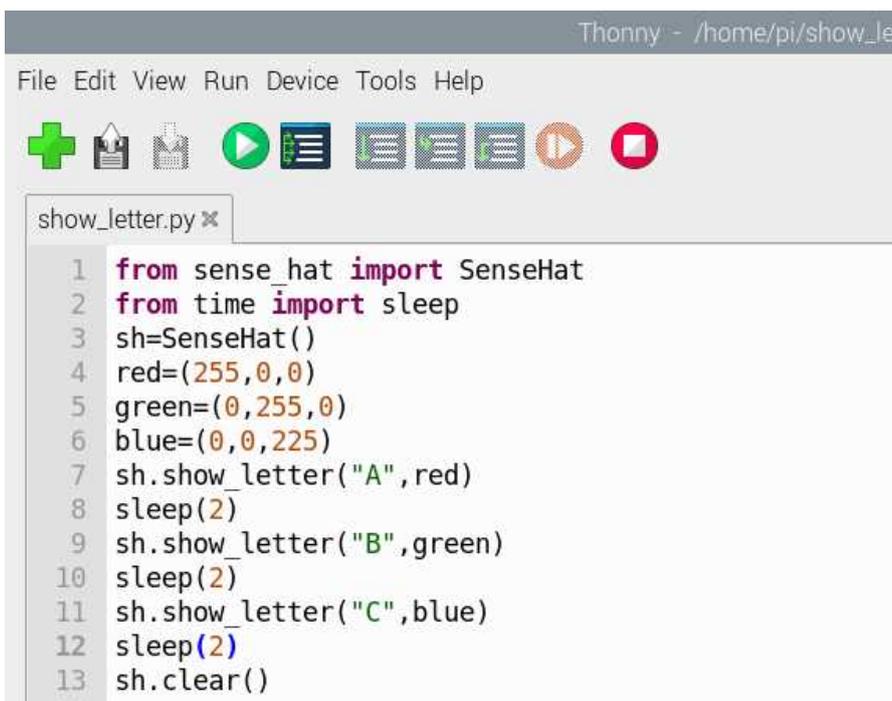
blue=[0,0,225]

sh.show_letter(" A ",red)
sleep(2)

sh.show_letter(" B ",green)
sleep(2)

sh.show_letter(" C ",blue)
sleep(2)

sh.clear()
```

A screenshot of the Thonny IDE interface. The title bar shows 'Thonny - /home/pi/show_le'. The menu bar includes 'File', 'Edit', 'View', 'Run', 'Device', 'Tools', and 'Help'. Below the menu bar is a toolbar with icons for a new file, open file, save file, run, and stop. The main editor window shows a file named 'show_letter.py' with the following Python code:

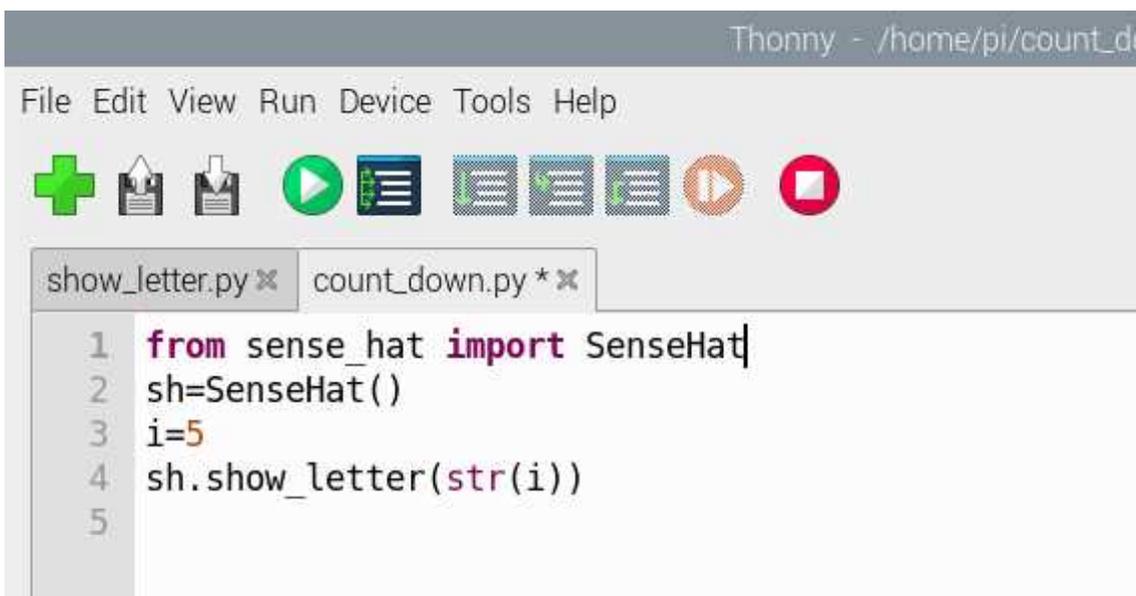
```
1 from sense_hat import SenseHat
2 from time import sleep
3 sh=SenseHat()
4 red=(255,0,0)
5 green=(0,255,0)
6 blue=(0,0,225)
7 sh.show_letter("A",red)
8 sleep(2)
9 sh.show_letter("B",green)
10 sleep(2)
11 sh.show_letter("C",blue)
12 sleep(2)
13 sh.clear()
```

3 一つの数字の表示

新しいファイル「show_letter2」を作成してください。
以下のソースを書きます。

```
from sense_hat import SenseHat  
  
sh=SenseHat()  
  
i=5  
  
sh.show_letter(str(i))
```

変数*i*の中の数値を文字列（string）として表示させています。文字の数字はLEDディスプレイの発光場所が決められていますが、値は決められてないのでstrをつけないと表示できません。

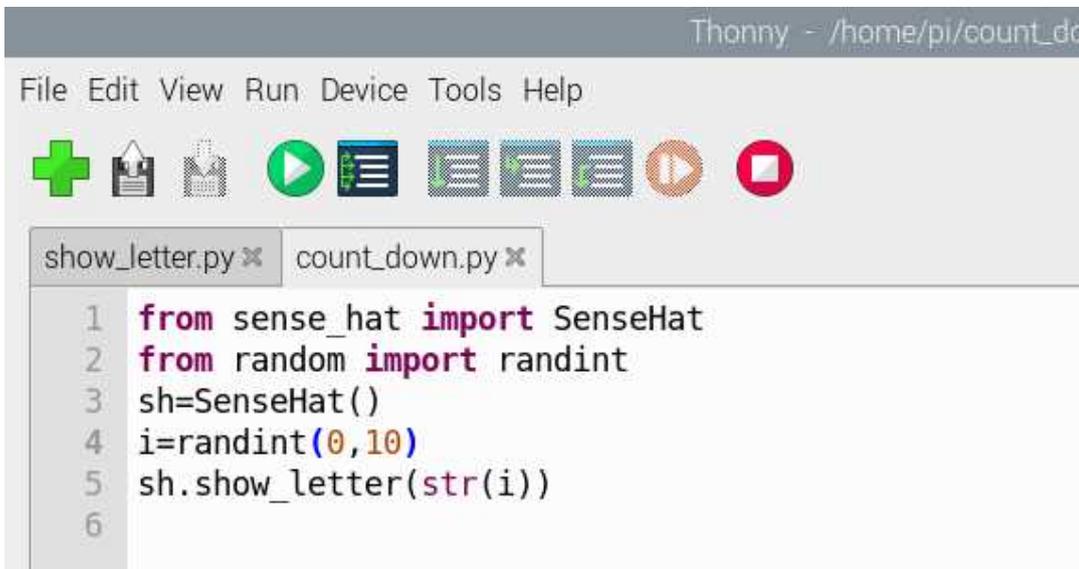


```
Thonny - /home/pi/count_d...  
File Edit View Run Device Tools Help  
+ [Icons]  
show_letter.py * count_down.py *  
1 from sense_hat import SenseHat  
2 sh=SenseHat()  
3 i=5  
4 sh.show_letter(str(i))  
5
```

このiの変数にランダムな数値を代入することもできます。
新しいファイル「show_letter3」を作成し、以下のソースを書きます。

```
from sense_hat import SenseHat  
  
from random import randint  
  
sh=SenseHat()  
  
i=randint(0,10)  
  
sh.show_letter(str(i))
```

random オブジェクトから整数のランダムな値を返す randint メソッドをインポートして、0 以上 10 未満の整数の乱数を発生させています。



```
Thonny - /home/pi/count_d  
File Edit View Run Device Tools Help  
+ [Icons]  
show_letter.py x count_down.py x  
1 from sense_hat import SenseHat  
2 from random import randint  
3 sh=SenseHat()  
4 i=randint(0,10)  
5 sh.show_letter(str(i))  
6
```

4 演習

「show_letter4」に次のソースを書き実行し、ソースの解説を作ってください。

```
from sense_hat import SenseHat
from time import sleep
sh=SenseHat()
for i in reversed(range(0,10)):
    sh.show_letter(str(i))
    sleep(1)
sh.clear()
```

from sense_hat import SenseHat

(解説) _____

from time import sleep

(解説) _____

sh=SenseHat()

(解説) _____

for i in reversed(range(0,10)):

(解説) _____

 sh.show_letter(str(i))

(解説) _____

 sleep(1)

(解説) _____

 sh.clear()

(解説) _____

解説例) sense_hat ライブラリから SenseHat モジュールを呼び出す

time ライブラリから sleep モジュールを呼び出す

SenseHat()の入った sh オブジェクトを作る

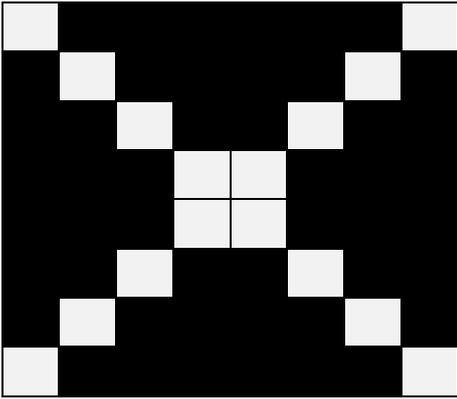
0以上10未満の整数を作り、大きいものからiという変数に格納する

値を文字列に強制変換し、表示させる

1秒間表示させた後、forをiが0になるまで繰り返す。

繰り返しが終わったら、表示をクリアする

5 イメージの表示



LEDひとつひとつを指定して点灯させて、上のような図を作ります。

どのLEDを点灯させるかを指定していきます。LEDの位置の座標は(x、y)で表します。

		x 軸							
y 軸		0	1	2	3	4	5	6	7
0	A								C
1		B							
2									
3						F			
4				E					
5									
6									
7	D								

値が0からスタートすることに注意してください。

AのLEDは(0,0)

BのLEDは(1,1)

CのLEDは(7,0)

DのLEDは(0,7)

EのLEDは(3,4)

FのLEDは(4,3)

になります。

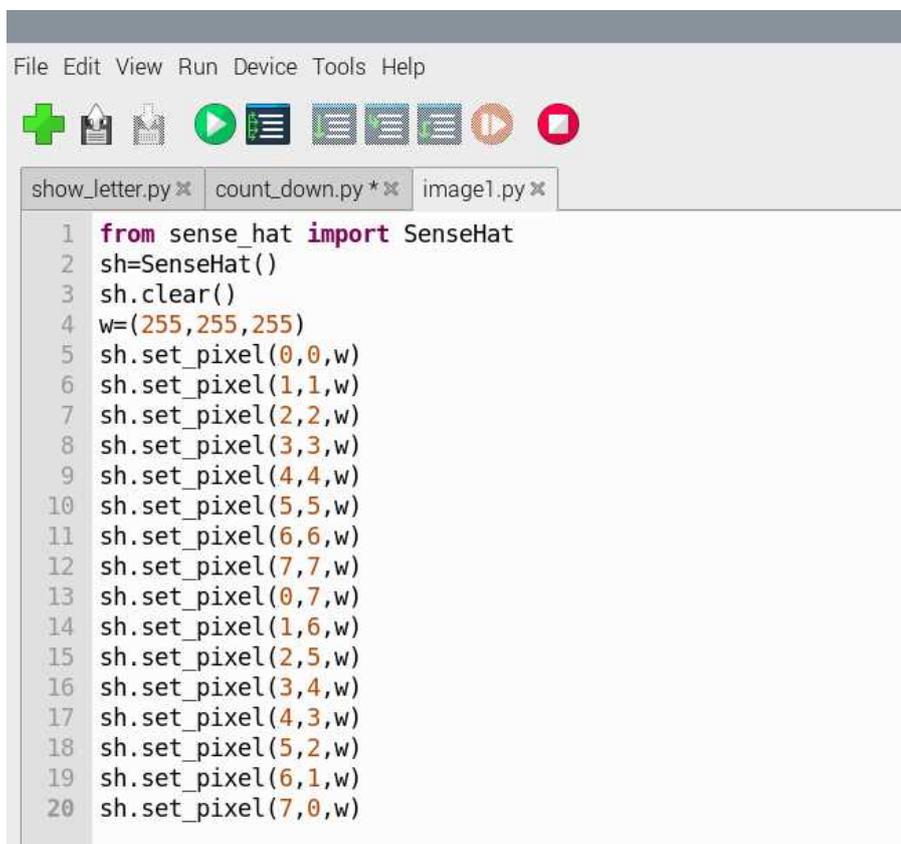
点灯させるLEDの座標をひとつずつ指定していきます。

また、座標の次に色を指定します。今回はすべて白(255,255,255)で作成します。

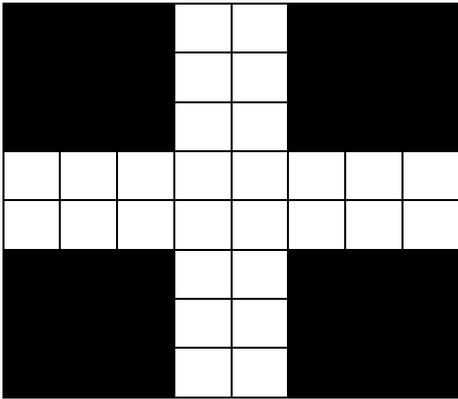
直前のLEDの情報が残っている場合があるので、最初にクリアをしておきます。

Image1 のファイルを作成して、次のソースを書いて実行してください。

```
from sense_hat import SenseHat
sh=SenseHat()
sh.clear()
w=(255,255,255)
sh.set_pixel(0,0,w)
sh.set_pixel(1,1,w)
sh.set_pixel(2,2,w)
sh.set_pixel(3,3,w)
sh.set_pixel(4,4,w)
sh.set_pixel(5,5,w)
sh.set_pixel(6,6,w)
sh.set_pixel(7,7,w)
sh.set_pixel(0,7,w)
sh.set_pixel(1,6,w)
sh.set_pixel(2,5,w)
sh.set_pixel(3,4,w)
sh.set_pixel(4,3,w)
sh.set_pixel(5,2,w)
sh.set_pixel(6,1,w)
sh.set_pixel(7,0,w)
```



The image shows a screenshot of a Python IDE window. The window title bar includes 'File Edit View Run Device Tools Help'. Below the title bar is a toolbar with icons for a file explorer, a play button, a list icon, a refresh icon, a play button, and a stop button. The main area of the window shows a code editor with three tabs: 'show_letter.py', 'count_down.py', and 'image1.py'. The code in the editor is identical to the code block above, with line numbers 1 through 20 on the left side of the editor.



イメージを作成するもうひとつの方法です。
 LEDは8 × 8の2元配列でなっているので、
 そのままの位置で色を指定するやり方です。

Image2 のファイルを作って、以下のソースを書いて実行してください。

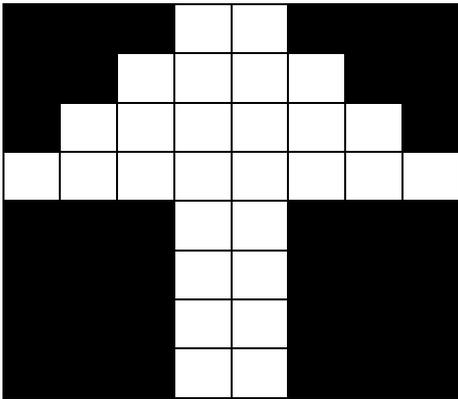
```
from sense_hat import SenseHat
sh=SenseHat()
w=(255,255,255)
r=(255,0,0)

image=[
r,r,r,w,w,r,r,r,
r,r,r,w,w,r,r,r,
r,r,r,w,w,r,r,r,
w,w,w,w,w,w,w,w,
w,w,w,w,w,w,w,w,
r,r,r,w,w,r,r,r,
r,r,r,w,w,r,r,r,
r,r,r,w,w,r,r,r,
]
sh.set_pixels(image)
```

set_pixels ですべてのピクセルに色のリストをLEDの座標に当てはめます。すべてのピクセルに色指定をするので、sh.clear()をする必要がありません。

6 イメージの回転・反転

練習) ファイル image3 を作り、以下のイメージのプログラムを組んでください。作成はすべてのLEDの色を当てはめる方法で行ってください。ただし、背景は黒。白い部分は白にします。



```
from sense_hat import SenseHat
sh=SenseHat()
w=(255,255,255)
K=(0,0,0)
image=[
k,k,k,w,w,k,k,k,
k,k,w,w,w,w,k,k,
k,w,w,w,w,w,w,k,
w,w,w,w,w,w,w,w,
k,k,k,w,w,k,k,k,
k,k,k,w,w,k,k,k,
k,k,k,w,w,k,k,k,
k,k,k,w,w,k,k,k,
k,k,k,w,w,k,k,k,
]
sh.set_pixels(image)
```

このイメージを 90 度、180 度、270 度回転させます。
一番下に次のソースを書いてください。

```
sh.set_rotation(180)
```

このように、イメージを回転させることができます。回転角度は、90 度、180 度、270 度です。

次に垂直方向に反転（ひっくり返し）します。反転は flip を使います。以下のソースを書いてください。

sh.flip_v()

垂直（vertical）方向にひっくり返ります。このイメージでは 180 度回転したときと同じになります。

水平（horizontal）方向にひっくり返すときは、flip_h を使います。ただしこの矢印のイメージを水平方向にひっくり返しても同じイメージになるので効果がわかりません。



```
Thonny - /home/pi/image3.py @ 18:10
File Edit View Run Device Tools Help
+ [Icons]
image2.py * [x] image3.py [x]
1 from sense_hat import SenseHat
2 sh=SenseHat()
3 w=(255,255,255)
4 k=(0,0,0)
5
6 image=[
7 k,k,k,w,w,k,k,k,
8 k,k,w,w,w,w,k,k,
9 k,w,w,w,w,w,w,k,
10 w,w,w,w,w,w,w,w,
11 k,k,k,w,w,k,k,k,
12 k,k,k,w,w,k,k,k,
13 k,k,k,w,w,k,k,k,
14 k,k,k,w,w,k,k,k,
15 ]
16 sh.set_pixels(image)
17 sh.set_rotation(180)
18 sh.flip_v()
```

3 章 センサデータの取得

1 気圧・温度・湿度センサデータの取得

sensor1 ファイルを作り、以下のソースを書いて実行してください。

```
from sense_hat import SenseHat

sh=SenseHat()

sh.clear()

print(sh.get_pressure())
```

気圧センサのデータを取得できます。単位はミリバール（ヘクトパスカル）です。

```
1 from sense_hat import SenseHat
2 sh=SenseHat()
3 sh.clear()
4 print(sh.get_pressure())
```

続いて温度と湿度データの取得をします。温度の単位は摂氏（：セルシウス度）、湿度は%です。

温度・湿度センサはラズパイのボードの発熱の影響を大きく受けます。室温とはかけ離れたデータになるので、ラズパイの温度と考えてください。室温を計測する場合は、センサをラズパイから離します。

```
print(sh.get_temperature())

print(sh.get_humidity())
```

```
1 from sense_hat import SenseHat
2 sh=SenseHat()
3 sh.clear()
4 print(sh.get_pressure())
5 print(sh.get_temperature())
6 print(sh.get_humidity())
```

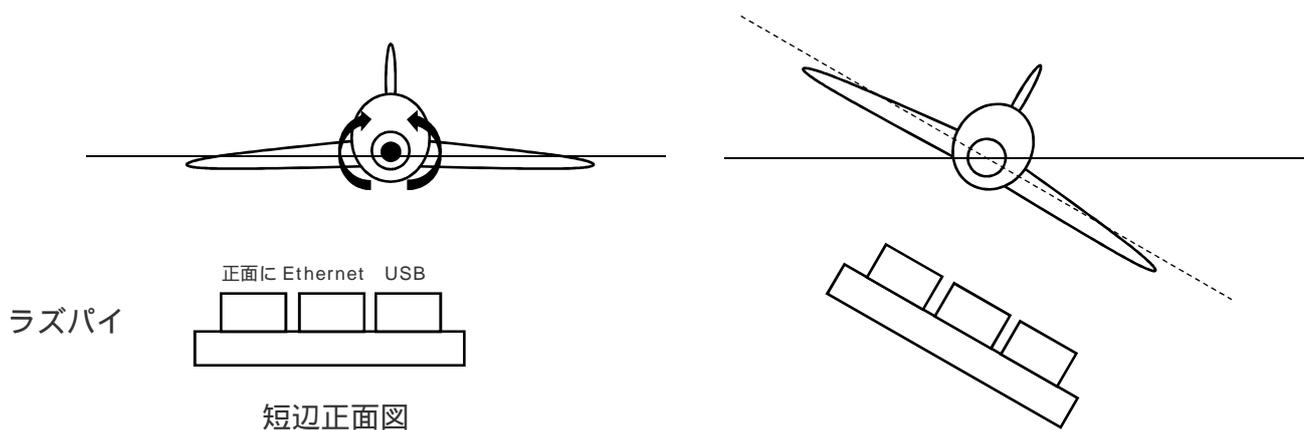
2 姿勢動作方位（ジャイロスコープ・加速度計・磁力計）データの基礎

ジャイロスコープは、物体の角度や角速度を測定するセンサです。カメラの手振れ補正やカーナビ、スマホ、ゲーム機などに用いられています。傾きや回転する物体の計測が得意です。加速度計は加速度を測定するセンサで、物体の傾きや衝撃の程度を計測します。地震計や歩数計、スマホ、ドローンやロボットの姿勢制御などに用いられています。傾きと動きを計測するのが得意です。磁力計は、方位磁石（コンパス）に用いられます。カーナビやスマホの地図に用いられています。方位を計測します。

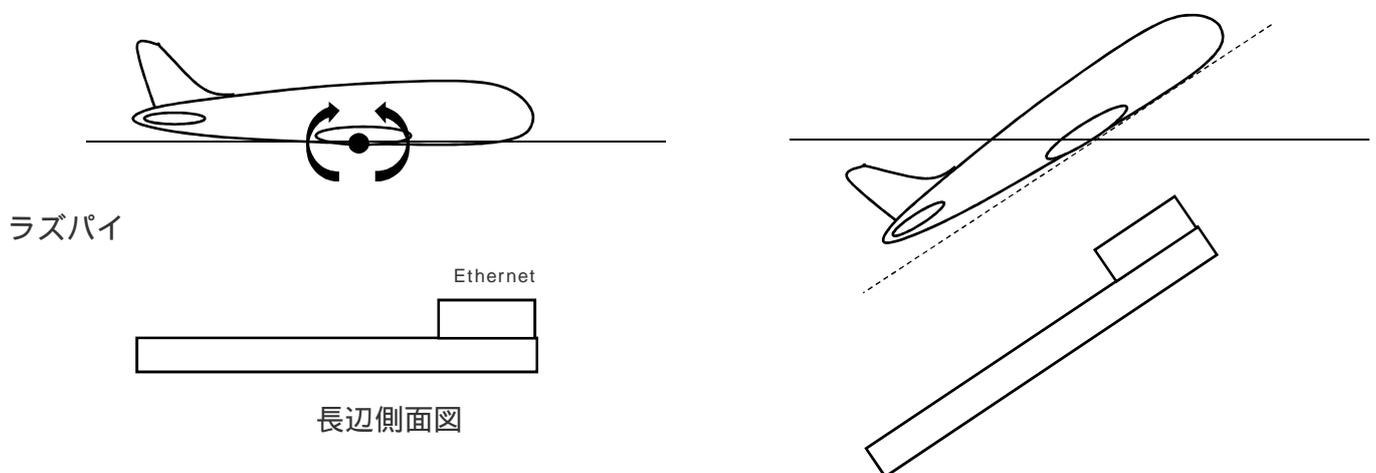
これらのセンサはそれぞれ、x軸、y軸、z軸方向（3軸）のデータを取得します。3つのセンサから得られたデータを基に、方位と姿勢の情報（オイラー角）を得ます。

オイラー角 Roll（ロール）、Pitch（ピッチ）、Yaw（ヨー）

（Roll（ロール））前後の軸に対して傾く。飛行機の旋回態勢。オートバイで回転の内側に傾斜するのもローリング。

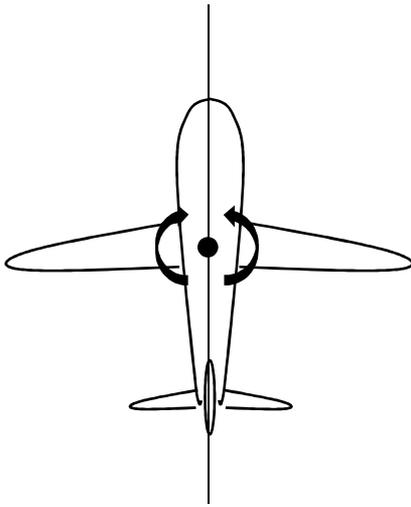


（Pitch（ピッチ））左右の軸に対して傾く。飛行機の離陸態勢。

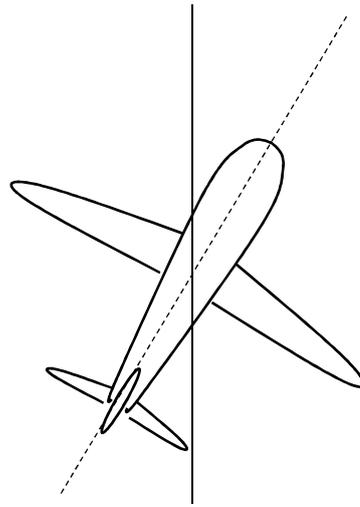


(Yaw (ヨー)) 上下の軸に対して傾く。水平方向の回転。

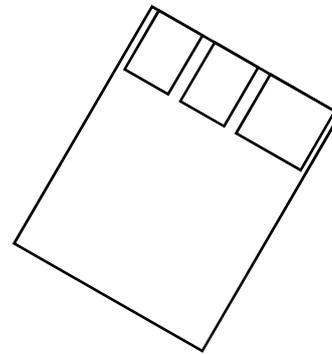
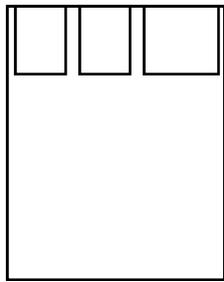
北



東にヨーイングを行った



ラスパイ



平面図 (上から見た図)

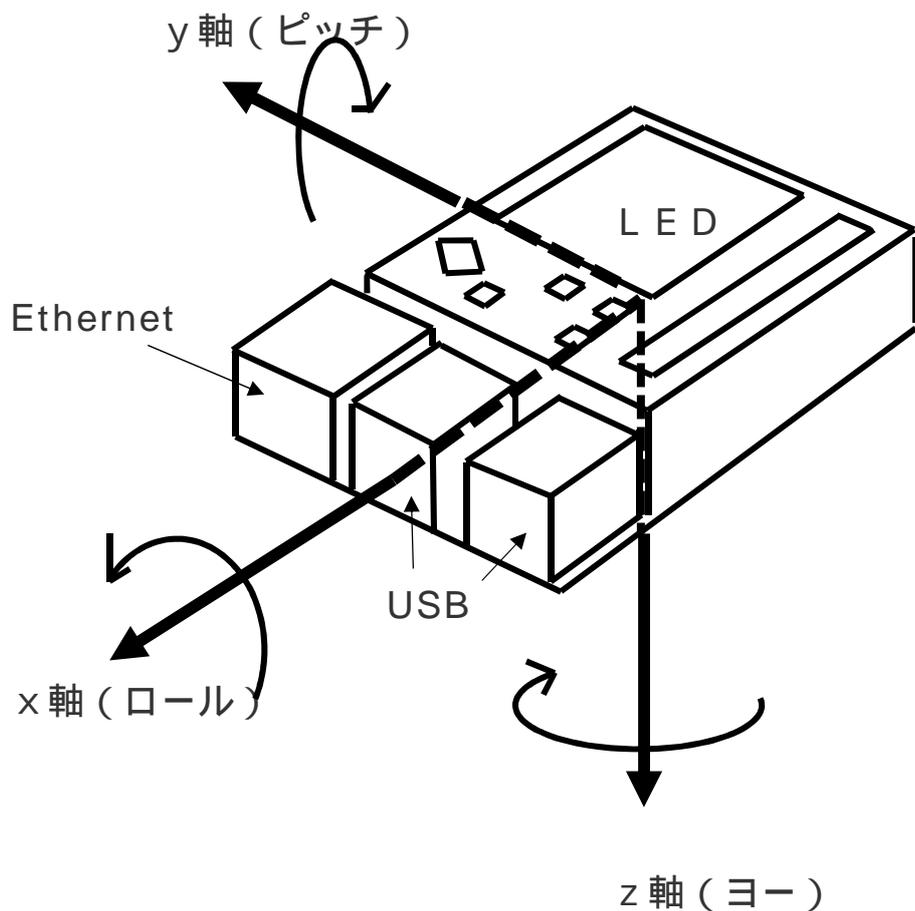
センスハットのロール、ピッチ、ヨー

Sense HAT version 1.0 の場合、それぞれの軸について重心に向かって反時計回りで 0 から 360 度まで値が変化します。

ロールの基準値 水平時 0度

ピッチの基準値 水平時 0度

ヨーの基準値 磁北 0度



3 姿勢動作方位データの取得

sensor2 ファイルを作り、以下のソースを書いて実行してください。

```
from sense_hat import SenseHat

sh=SenseHat()

print(sh.get_compass())

print(sh.get_gyroscope())

print(sh.get_gyroscope_raw())

print(sh.get_accelerometer())

print(sh.get_accelerometer_raw())
```

(ソースの解説)

```
print(sh.get_compass())
```

磁力計をコンパスとして利用し、磁北を0度で表します。ただし、磁石が示す磁北と方位の真北^{しんぼく}は異なります。地磁気は常に変化しています。2015年現在北海道は約9度、東京は約7度、沖縄は約5度のずれがあります。(国交省国土地理院・磁気図「地理院地図版」参照 <https://www.gsi.go.jp/index.html>)

```
print(sh.get_gyroscope())
```

ジャイロスコープのデータです。オイラー角のロール、ピッチ、ヨーで角度を表しています。

```
print(sh.get_gyroscope_raw())
```

同じくジャイロスコープのデータですが、ロールの軸をx、ピッチの軸をy、ヨーの軸をzとしラジアンで表示しています。ラジアンは、180度を1としたときの値です。

```
print(sh.get_accelerometer())
```

加速度計のデータです。オイラー角のロール、ピッチ、ヨーでセンサに生じている加速度などから傾きを表しています。

```
print(sh.get_accelerometer_raw())
```

同じく加速度計のデータですが、ロールの軸をx、ピッチの軸をy、ヨーの軸をzとし標準重力加速度(9.8m/s²)を1として表します。水平に静止して置いている場合の理論値は、x軸(ロール)、y軸(ピッチ)向きには重力加速度は発生しないので0。z軸(ヨー)方向のみ重力加速度が発生しているので1になります。

```

1 from sense_hat import SenseHat
2 sh=SenseHat()
3
4 print(sh.get_compass())
5 print(sh.get_gyroscope())
6 print(sh.get_gyroscope_raw())
7 print(sh.get_accelerometer())
8 print(sh.get_accelerometer_raw())

```

Shell ✕

>>> %Run sensor2.py

```

0.18835320523474586
{'roll': 0.00042064102388229103, 'pitch': 0.0006252267683612648, 'yaw': 0.18491615265877495}
{'x': -0.00032226089388132095, 'y': 0.0008600112050771713, 'z': -0.0026257820427417755}
{'roll': 0.01685656669615429, 'pitch': 0.002227871887212286, 'yaw': 0.1825278276050919}
{'x': -0.0019746851176023483, 'y': 0.01383917685598135, 'z': 0.9986696839332581}

```

Shell のデータはラズパイによって異なります。

一番上の 0.188・・・はコンパスです。磁北から 0.188 度東に傾いていることを表しています。

二番目はジャイロスコープの計測値です。ロールは 0 度が水平なのでほぼ水平です。ピッチも 0 度が水平なのでほぼ水平です。ヨーは磁北のずれです。磁力計では 0.188 度のずれでしたが、ジャイロスコープの磁北では 0.185 度のずれを計測しています。0.003 度の誤差が生じています。

三番目はジャイロスコープの計測値をラジアンで表したものです。

四番目はアクセロメータの計測値です。ロールもピッチもほぼ水平です。磁力計は 0.188 度、アクセロメータは 0.183 度で、0.005 度の誤差が生じています。

五番目はアクセロメータの計測値を重力 G で表したものです。重力の向きを示す z 軸は重力 (G) = 1 に近い値になっています。

次に、1秒ごとにアクセロメータからデータを得てセンサにかかる重力加速度を計測するプログラムを作ります。Sensor3 ファイルを作り、以下のソースを書いて実行してください。

```
from sense_hat import SenseHat
sh=SenseHat()

import time

while True:

    print(sh.get_accelerometer_raw())

    time.sleep(1)
```

(ソースの解説)

```
print(sh.get_accelerometer_raw())
```

加速度計から3軸の重力加速度データを取得します。

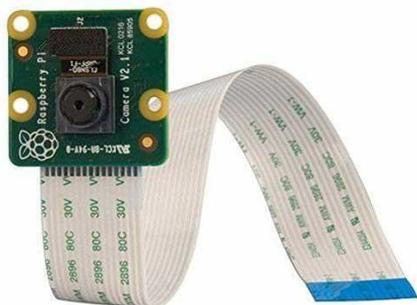
```
time.sleep(1)
```

1秒間プログラムを停止させます。

ラズパイを縦方向に垂直、横方向に垂直、水平方向に垂直など動かして、アクセロメータが計測する重力加速度の向きを確認してみてください。

4 章 カメラ画像データの取得

1 カメラモジュールの取り付けと設定



ラズパイにカメラを取り付けます。
ラズパイ用に用意されているものを使用します。

写真は Raspberry Pi Camera V2

露光・ホワイトバランス・輝度などを自動制御します。

Sense HAT が取り付けられているときは、取り外して作業をします。



Raspberry Pi 4 B



Raspberry Pi 3 B +

この部分がカメラを取り付けるコネクタです。コネクタの両端にある爪を上を持ちあげます。そうすると USB や Ethernet ポート側にすきまができます。

ケーブルの青いテープが貼ってある側を USB ポートに向けてケーブルを差し込みます。

この時、Sense HAT にあるカメラケーブル用のスリット（通し穴）にケーブルを通すのを忘れないように。爪を下におろせば完成です。

この後、Sense HAT を取り付けてください。

カメラの台座の組み立ては特に必要ありませんが、参考のために台座の組み立ての動画を作成しています。台座を組み立てる場合は、表面の保護紙をはがしてください。動画では、保護紙をはがした状態から説明しています。

ラズパイ側の設定を行います。

メニューから「設定」 - 「Raspberry Pi の設定」をクリックします（左図）。

右のウィンドウが出てきたら、インターフェイスのタブをクリックします。

一番上の「カメラ」のチェックボタンを有効にして、OKを押します。



2 カメラの操作と撮影

L Xターミナルを立ち上げ、次のソースを書きます。

```
raspistill -d
```

5 秒間カメラ映像が表示されます。

(ソースの解説)

raspistill 静止画を撮影するコマンドです。

-d カメラのデモモードです。

画像のコントラストを上げてみます。

```
raspistill -d -co 100
```

画像のコントラストを上げたまま、画像のスケッチ処理をします。

```
raspistill -d -co 100 -ifx sketch
```

このように、画像のコントラストや輝度、ホワイトバランス(ナイトモード・逆光モード・花火モードなど)、露出(晴天モード・くもりモード・フラッシュモード・日没モードなど)、画像効果(水彩・油彩・日の出・ポスター・アニメなど)等デジタルカメラで使われているような撮影条件の設定ができます。

では、次に撮影をして画像を保存します。

```
raspistill -o test.jpg
```

画像が出て 5 秒後に撮影されます。画像ファイルは、タスクバーの「ファイルマネージャ」をクリックして出てくるフォルダに「test.jpg」として保存されています。ファイルをクリックして開いてみてください。

(ソースの解説)

raspistill 静止画を撮影するコマンドです。

-o test.jpg 「-o」は、output で、出力ファイル名を指定するパラメータです。この場合は、「test.jpg」として画像データが保存されます。指定しない時は保存されません。

もし、画像の上下(天地)が逆になっていたり、90度傾いていたりした場合は、画像を回転させて撮影します。次の例では、天地が逆になっている場合のソースです。

```
raspistill -o test.jpg -rot 180
```

(ソースの解説)

-rot 180 「-rot」は、rotation で、画像を 0 ~ 359 度回転させるパラメータです。

このように、raspistill コマンドの後ろにパラメータを付けることで様々な設定や効果を得ることができます。パラメータは複数使うことができます。

カメラのコマンドとして、3種類用意されています。

raspistill	ファイルを圧縮した静止画コマンド
raspiyuv	ファイルを圧縮しない静止画コマンド
raspivid	ビデオ動画撮影コマンド

それぞれのコマンドのパラメータは以下の表のとおりです。パラメータを変えてどのようになるか試してみてください。

< raspistill (静止画コマンド) パラメータ >

-w	width	画像の幅指定
-h	height	画像の高さ指定
-q	quality	画像の質指定 (0~100)
-o	output	出力ファイル名。指定しない場合保存されない。
-t	timeout	撮影までの時間をミリ秒で指定。3秒後は3000。指定しない場合は5秒。
-d	demo	撮影をせず、デモモードを実行する。
-e	encoding	出力ファイルのエンコード (jpg,bmp,gif,png) の指定
-tl	timelapse	ミリ秒で指定した間隔で連続撮影する (コマ撮りモード)。

< raspiyuv (ファイルを圧縮しない静止画コマンド) パラメータ >

-w	width	画像の幅指定
-h	height	画像の高さ指定
-o	output	出力ファイル名。指定しない場合保存されない。
-t	timeout	撮影までの時間をミリ秒で指定。3秒後は3000。指定しない場合は5秒。

< raspivid (ビデオコマンド) パラメータ >

-w	width	画像の幅指定
-h	height	画像の高さ指定
-b	bitrate	ビットレートの指定。1Mbits/s は -b 1000000
-o	output	出力ファイル名。
-t	timeout	撮影時間をミリ秒で指定。1分間は60000。指定しない場合は5秒。0は撮影なし。
-d	demo	撮影をせず、デモモードを実行する。
-f	framerate	フレームレートを指定。フレーム/秒。

< raspivid、raspistill、raspiyuv 共通パラメータ >

(プレビューパラメータ) ラズパイをディスプレイに接続している時のプレビューの制御

-f	fullscreen	フルスクリーンでプレビュー
-op	opacity	プレビューの不透明度指定 (0~255)。
-n	nopreview	プレビューしない。

(イメージパラメータ) 撮影条件の制御

-sh	sharpness	シャープネス(輪郭強調)(-100~100)	
-co	contrast	コントラスト(明暗比)(-100~100)	
-br	brightness	ブライツネス(輝度)(0~100)	
-rot	rotation	画像の回転(0~359)	
-hf	hflip	水平反転	
-vf	vflip	垂直反転	
-ex	exposure 露出モード	off	自動露出オフ
		auto	自動露出オン
		night	ナイトモード
		backlight	逆光モード
		spotlight	スポットライト
		sports	スポーツモード
		snow	スノーモード
		antishake	ぶれ防止
		fireworks	花火
-awb	auto white balance ホワイトバランスモード	off	自動ホワイトバランスオフ
		auto	自動ホワイトバランスオン
		sun	晴天モード
		cloud	くもりモード
		shade	日陰モード
		tungsten	タングステン照明モード
		fluorescent	蛍光灯照明モード
		incandescent	白熱電球モード
		flash	フラッシュモード
		horizon	日没モード
-ifx	lmaxfx 画像効果	non	画像効果なし
		negative	反転効果
		solarize	日の出効果
		sketch	スケッチ効果
		denoise	ノイズ除去効果
		emboss	エンボス効果
		oilpaint	油彩効果
		pastel	パステル効果
		watercolor	水彩効果
		film	フィルム効果
		blur	ぼやけ効果
		posterise	ポスタリゼーション(階調変更)効果
		cartoon	アニメ効果

3 カメラアプリ (picamera) の利用

Thonny Python IDE を起動して camera1 ファイルを作り、以下のソースを書いて実行してください。

```
import picamera

cmr=picamera . Pi Camera()

cmr . start_preview()
```

常にカメラが画面に映し出されます。

カメラ画面を消す時は、コントロールキーを押しながら Q のキーを押します。もし Q が効かない時は、D を押してみてください。(OS のバージョンによってキーボードショートカットが異なります)

動作が確認できたら、10 秒カメラの画面を表示した後、もとに戻すプログラムを追加します。

```
import picamera

import time

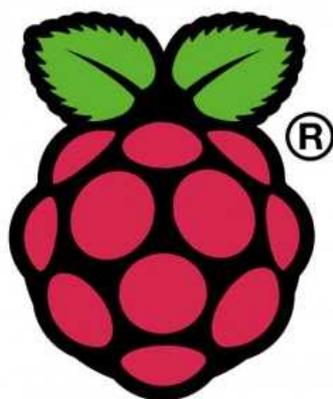
cmr=picamera . Pi Camera()
cmr . start-preview()

time . sleep(10)

cmr . stop_preview()
```

IoT 基礎

Raspberry pi Remote



目 次

1 章	Raspberry Pi + Sense Hat と P C の接続	
1	Raspberry の設定	1
2	V N C Viewer のインストールと設定	3
3	WinSCP のインストールと設定	6
2 章	演習	
1	演習	9

1 章 Raspberry Pi + Sense HAT と P C の接続

1 ラズパイの設定

センサハットとカメラを搭載したラズパイから、「温湿度」「圧力」「姿勢」「衝撃振動」「画像」のデータを得ることができます。これらのデータを P C に取り込み各種分析を行うためラズパイと P C を接続します。

P C とラズパイを接続する方法は、S S H 接続と V N C 接続があります。S S H (Secure Shell : セキュアシェル) 接続は、暗号や認証技術を利用して、安全にリモートコンピュータと通信するための通信手段 (プロトコル) です。ラズパイと P C の双方が S S H でやりとりをする取り決めを行い、通信が暗号化されます。主に、C L I (Command Line Interface) 環境の接続に使います。

V N C (Virtual Network Computing) 接続は、ネットワーク上の離れたコンピュータを遠隔操作するプロトコルを使用します。主に G U I (Graphical User Interface) 環境の接続に使います。

まず、ラズパイの設定から行います。

メニューから「設定」-「Raspberry Pi の設定」をクリックします。

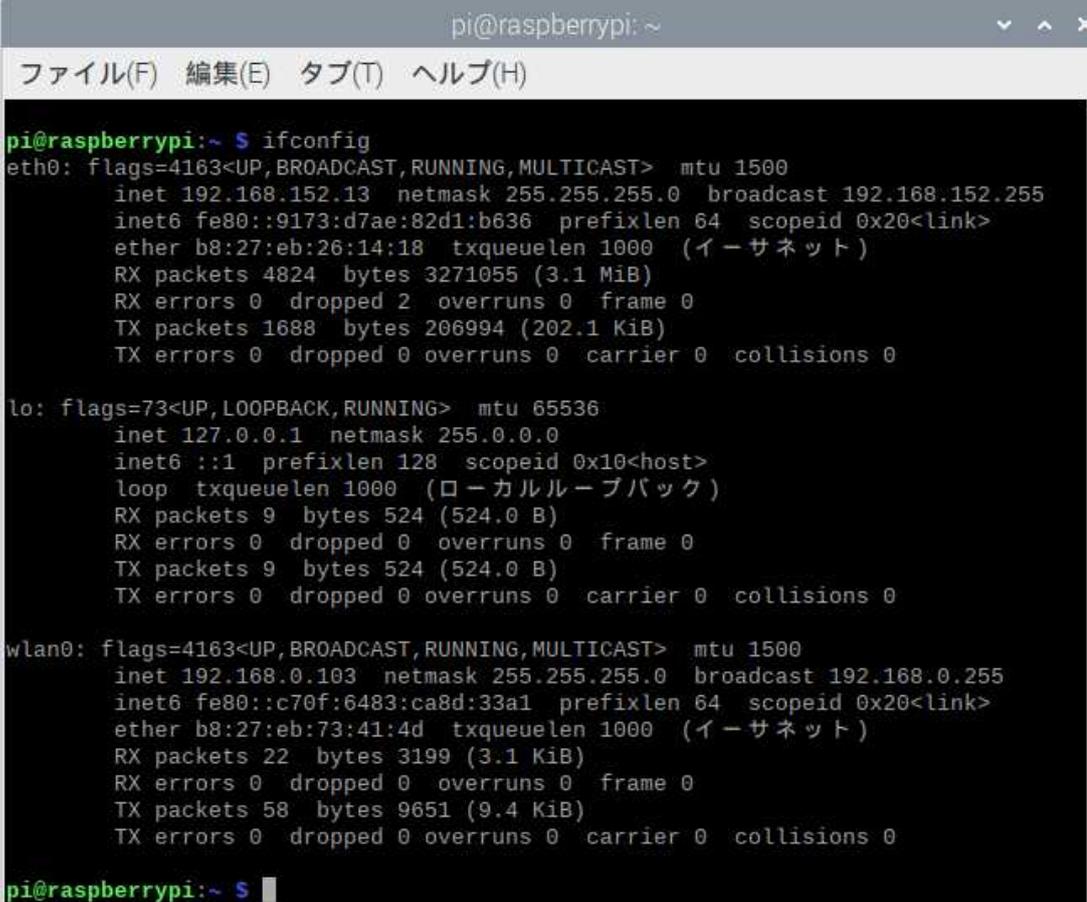
「インターフェイス」タブをクリックして、S S H と V N C のチェックボタンをつけて O K を押しします。再起動をするか聞いてくるので、すぐに再起動 (reboot) をしてください。再起動後ユーザー名やパスワードについて確認画面がでますが O K を押しします。



次に、ラズパイのタスクバーから L X ターミナルを開いて次のコマンドを書きます。

ifconfig

これは、ネットワーク環境の状態を確認・設定するコマンドです。



```
pi@raspberrypi:~ $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.152.13 netmask 255.255.255.0 broadcast 192.168.152.255
    inet6 fe80::9173:d7ae:82d1:b636 prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:26:14:18 txqueuelen 1000 (イーサネット)
    RX packets 4824 bytes 3271055 (3.1 MiB)
    RX errors 0 dropped 2 overruns 0 frame 0
    TX packets 1688 bytes 206994 (202.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (ローカルループバック)
    RX packets 9 bytes 524 (524.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 9 bytes 524 (524.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.103 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::c70f:6483:ca8d:33a1 prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:73:41:4d txqueuelen 1000 (イーサネット)
    RX packets 22 bytes 3199 (3.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 58 bytes 9651 (9.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@raspberrypi:~ $
```

画面のデータ表示はラズパイによって異なります。

上の画面では大きく3つの部分に分かれています。上は「eth0」、真ん中は「lo」、下は「wlan0」です。それぞれがインタフェースの名前で、「eth0」はイーサネット0（ゼロ）、「lo」はローカルループバック、「wlan0」はワイヤレスラン0（ゼロ）です。イーサネットはケーブル接続、ワイヤレスランは無線ラン（Wifi）接続のことです。「lo」はネットワークテスト用なので使うことはありません。

接続に必要なのは、二行目のinetのIPアドレス（Internet Protocol Address）です。これは、ラズパイの住所を表しています。この画面の例では、イーサネット接続で192.168.152.13、wifi接続で192.168.0.103の住所になっています。

無線LANでPCとラズパイを接続する場合は「wlan0」のIPアドレスが、LANケーブルでPCとラズパイを接続をするときは、「eth0」のIPアドレスが必要になります。それぞれメモしておいてください。

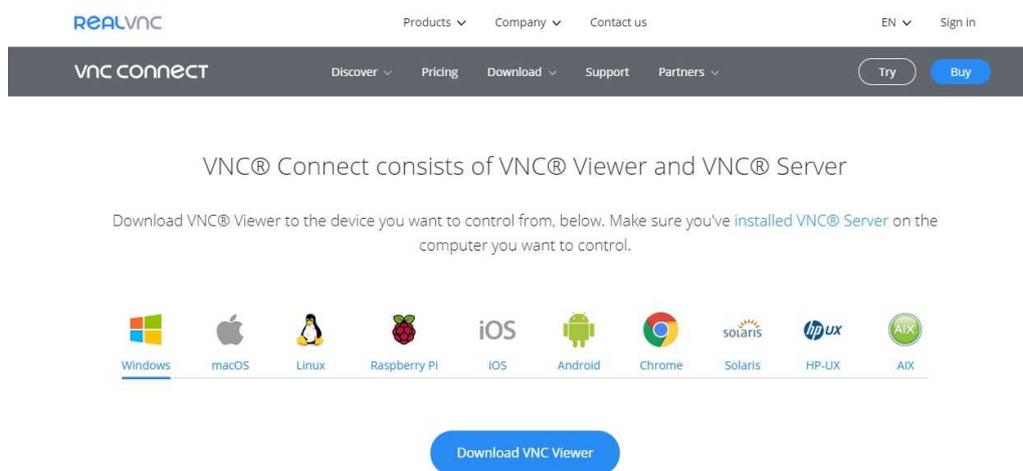
ちなみに、RXはデータがまとめてあるパケット（小包）をいくつ受信（Receiver）したか、TXはいくつ送信（Transmitter）したかの結果が表示されています。

2 「VNC Viewer」のインストールと設定

PCに「VNC Viewer」をインストールします。「VNC (Virtual Network Computing) Viewer」は、GUI通信プロトコルに対応した無料のリモートデスクトップソフトです。インストールされているパソコンのOSに関わらず通信することができます。また、パソコンだけでなく、スマホ・タブレット端末から離れた場所にあるPCを操作することもできます。

「VNC Viewer」を検索すると、ダウンロードページが出てくるのでクリックします。

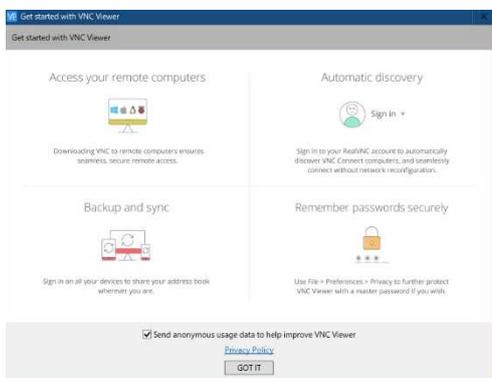
例) Download VNC Viewer for Windows | VNC Connect



ダウンロードのボタンを押してダウンロードをして、インストールします。



インストーラは日本語がないので、英語にします。インストールが終わったら、exe ファイルをクリックしてください。インストーラが立ち上がりインストールされます。

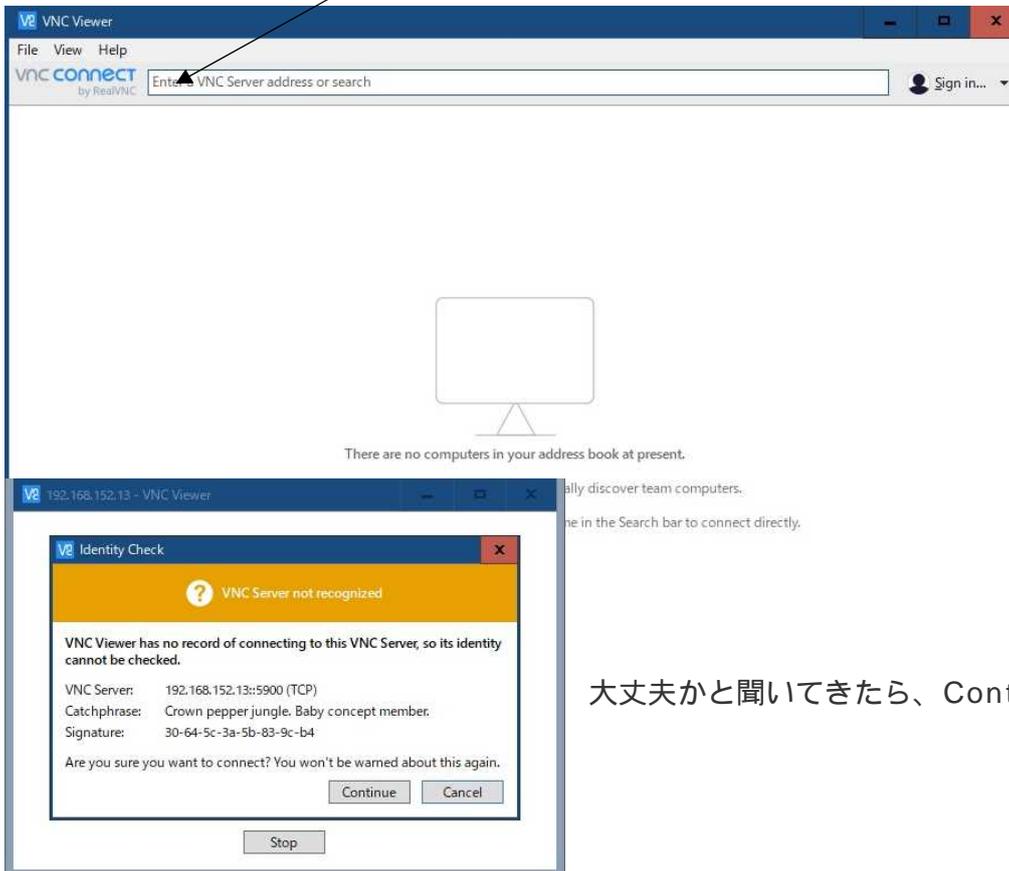


インストールが終わって次の画面がでたら、Got It (わかった、了解した) を押します。

インストールが終わったら、ウィンドウズの「スタートボタン」-「プログラム」の中から、「RealVNC」の右の を押し「VNCのアイコン」を右クリックして「その他」-「タスクバーにピン留め」や、アイコンをデスクトップにドラックしてデスクトップにショートカットを作ってください。

そして、アイコンをクリックしてプログラムを立ち上げます。

このような画面になるので、サーチのところにラズパイのIPアドレスを書いてクリック。



大丈夫かと聞いてきたら、Continue（続ける）をクリック。

次に、ラズパイのユーザー名とパスワードを聞いてくるので、



username は、 pi
password は、 raspberry
と入力し、Remember password（パスワードを覚えておく）にチェックを入れておきます。

ラズパイの画面が表示されます。

では、遠隔操作ができるか確認をしてみます。

L Xターミナルを立ち上げ、以下のソースを書いて実行してください。

```
raspistill -o test.jpg
```

画像が出て 5 秒後に撮影されます。画像ファイルは、タスクバーの「ファイルマネージャ」をクリックして出てくるフォルダに「test.jpg」として保存されています。ファイルをクリックして開いてみてください。今、撮影された画像を見ることができます。

このように、ラズパイ本体で行っていた「L Xターミナル」「Thonny Python」などの操作が、P Cからも遠隔操作できるようになっています。ただし、全ての操作が遠隔でできるわけではなく、ラズパイからでないといけない操作もあります。

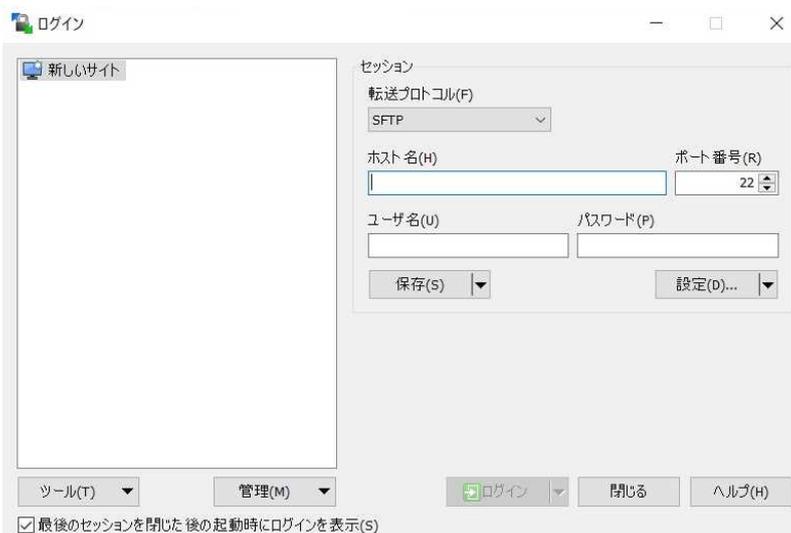
3 PCとラズベリーパイのフォルダ共有化（「WinSCP」のインストール）

PCに「WinSCP」をインストールします。「WinSCP（Windows Secure Copy）」は、遠隔操作をするPCとラズパイの間で安全なファイルのコピーができるようになります。

「winSCP」を検索すると、「WinSCP プロジェクト日本語トップページ」が出てきます。このページに行き、最新の exe ファイルをダウンロードします。ダウンロードが終わったらインストールをします。

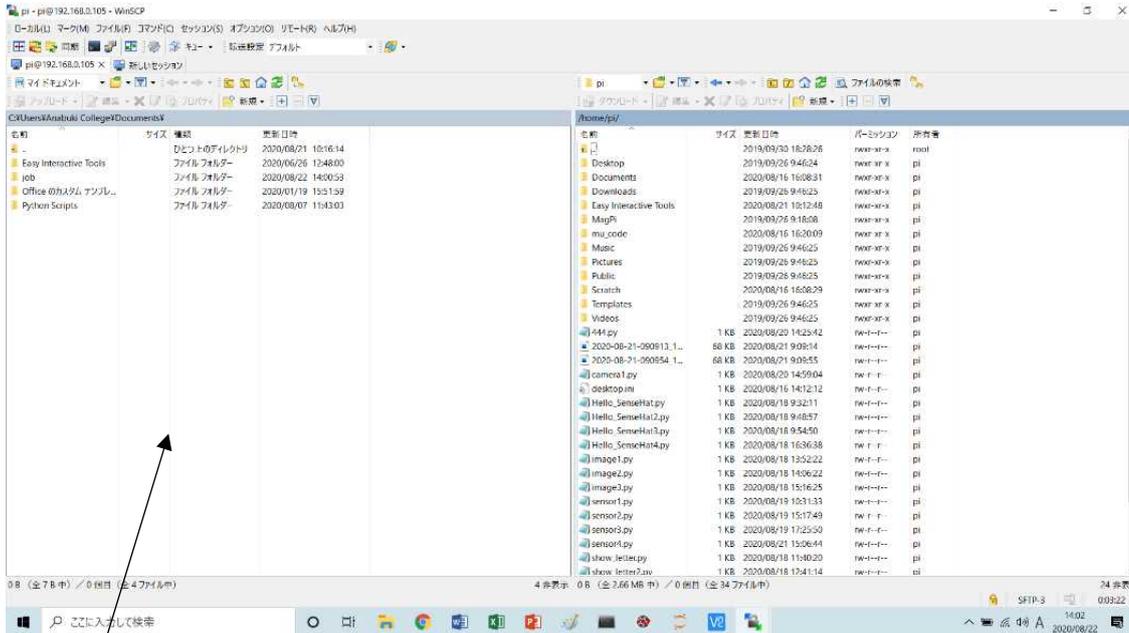


インストールが終わったら、プログラムを起動します。
ホスト名にはラズベリーパイのIPアドレスを入力します。
ラズベリーパイのユーザー名とパスワードを入力して、ログインを押します。

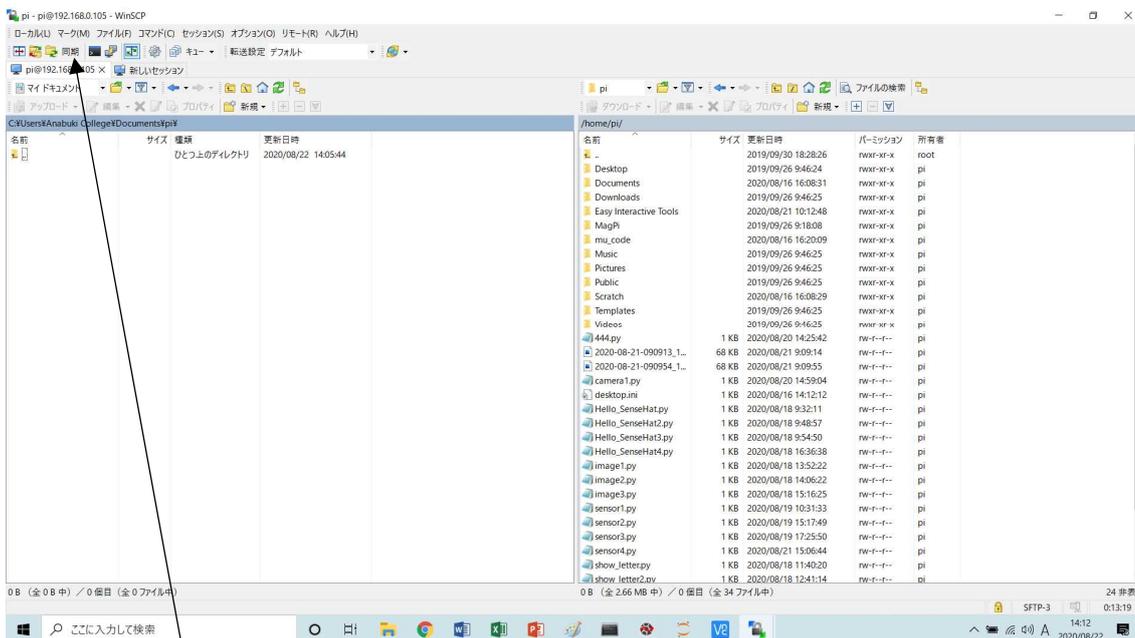


不明なサーバーに接続し、そのホスト鍵をキャッシュに追加しますか？という警告がでるので、「はい」を押します。次回からはこの画面はでなくなります。

ログインすると、左にPC（ローカル）、右にラズベリーパイ（リモート）のフォルダが表示されます。

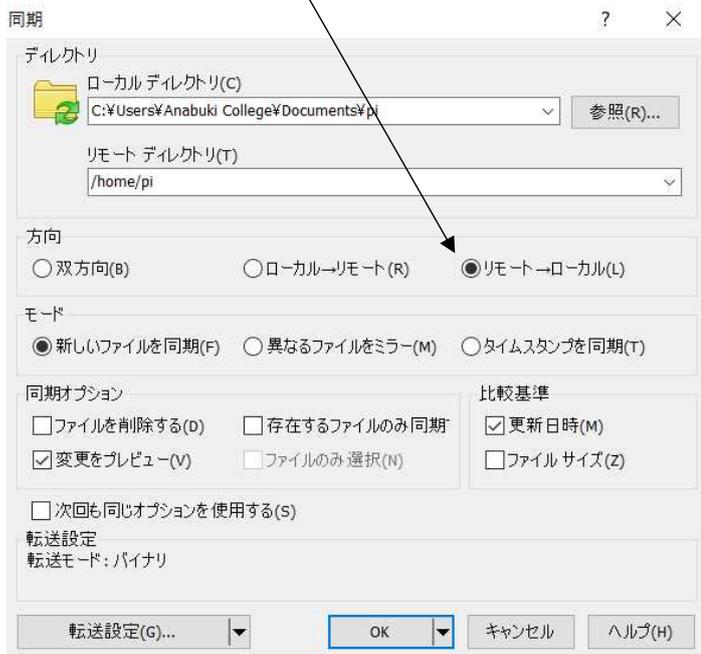


PC（ローカル）に、ラズパイ（リモート）のデータを同期させます。まず、左の画面のどこかで、右クリックして「新規」-「フォルダの作成」を押して新しいフォルダを作り、名前を「pi」にしてください。そして、「pi」をクリックします。

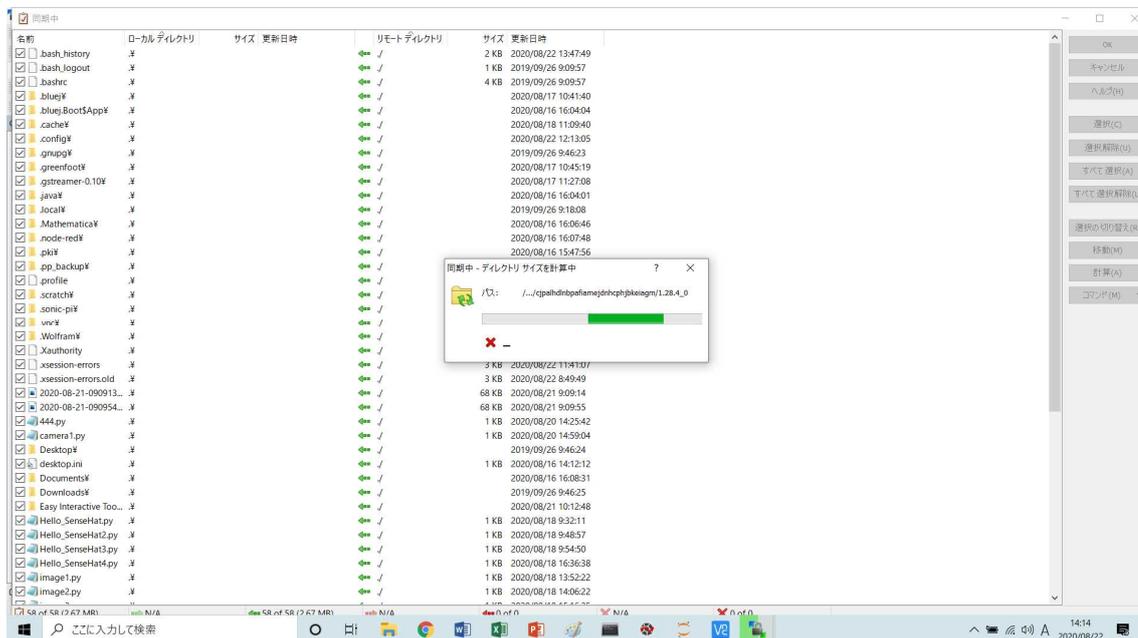


次に、上の「同期」をクリックしてください。

同期の方向で、リモート ローカルを選び、OKを押します。



同期が終わると、PCの pi フォルダがラズパイの pi フォルダと中身がじになります。このようにして、ラズパイのデータをPCに取り込み、操作や分析などを行います。



2章 演習

準備：ラズパイをVNCで遠隔操作します。PCからラズパイにアクセスしてください。

プログラムの仕様 ファイル名「sensor4」

ラズパイからアクセロメータのラジアンデータを10秒ごとに得る。

ラズパイが傾いた時だけ異常を警告する。

重力加速度が0.5より小さい場合「warning」

重力加速度が0.9より小さく0.5以上の場合「alert」を表示

警告のあと重力加速度を表示する。さらに、感知した年月日と時刻を表示する。

表示例) warning 0.385454565 2020/08/22,15:13:12

(level 1) ラズパイのセンスハットのアクセロメータからz軸のデータのみを取得し、表示させます。

アクセロメータからの重力加速度をラジアンで取得するソース例

```
raw=SenseHat().get_accelerometer_raw()
```

アクセロメータからz軸のデータのみを取得するソース

```
( "{z}" .format(**raw) )
```

ここで、得た数字は文字列になるので、浮動小数点に変換する必要があります。

(level 2) アクセロメータからのデータ取得を10秒おきに設定し、表示します。

time モジュールをインポートし sleep を設定します。

(level 3) データによって分岐をつくり warning と alert と異常値を表示します。

level 1 で作ったデータ表示は削除します。

(level 4) 異常が計測された年月日、時間(時分秒)を表示します。

from datetime import datetime でモジュールや関数のインポートを行います。

現在時刻は、datetime.now().strftime('%Y/%m/%d,%X')で取得します。