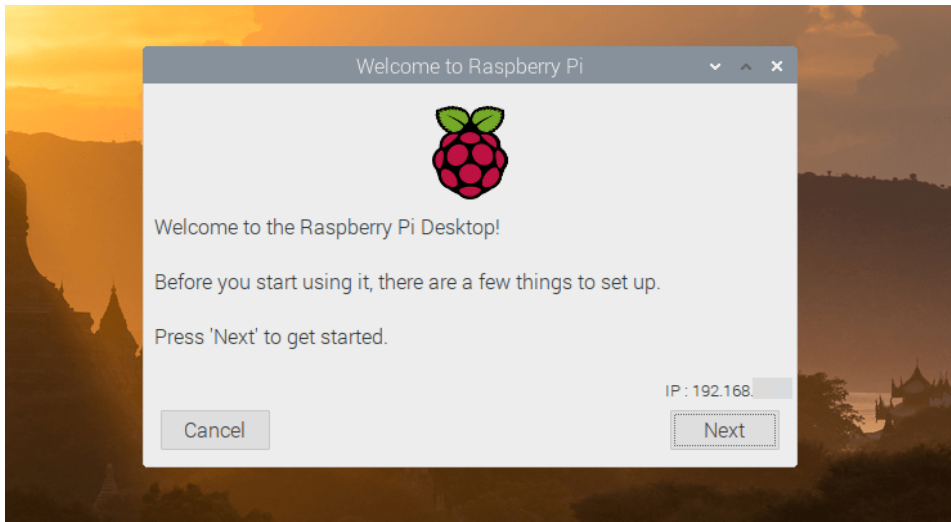


シート一覧

No.	シート名	内容	備考
1	【1】 ラズベリーパイ 初回起動時の設定手順	OSの初期設定手順	
2	【2】 WindowsPCからSSHでリモート接続	Windowsからラズベリーパイにリモート接続(SSH)	
3	【3】 SHから「raspi-config」でVNCを有効化手順	ラズベリーパイのVNCを有効化	
4	【4】 WindowsPCからVNCでGUI画面にリモート接続	Windowsからラズベリーパイにリモート接続(VNC)	
5	【5】 CO2センサ(mh_z19)を利用するまでの手順	CO2センサの接続から Pythonコマンドで測定値取得 1分ごとの測定値をCSVファイル出力 CSVデータをWebでトレンドグラフ表示	
6			
7			
8			
9			
10			

【1】ラズベリーパイ初回起動時の設定手順

①「NEXT」で次へ進む

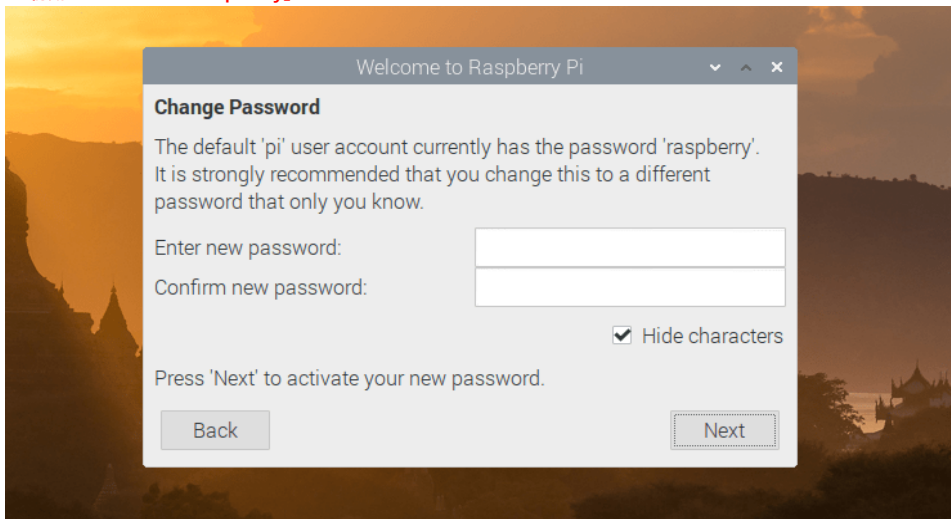


②Country「Japan」を選択して「NEXT」で次へ進む

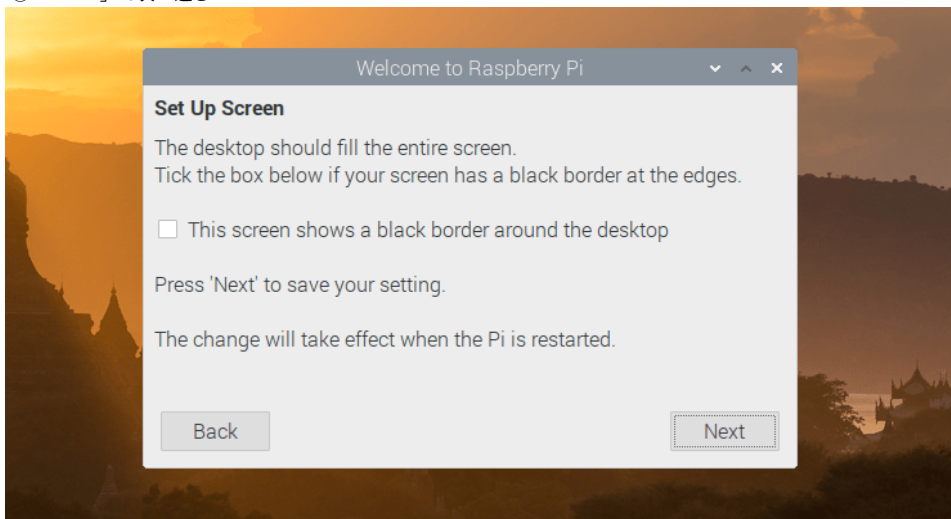


③新しいパスワードを入力して「NEXT」で次へ進む

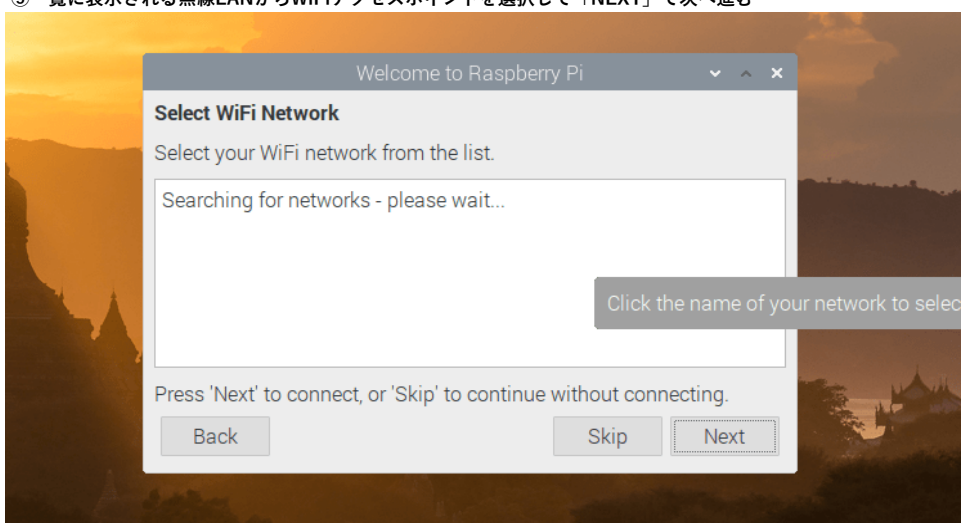
※初期パスワードは「raspberrry」



④「NEXT」で次へ進む

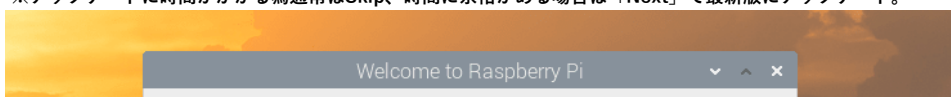


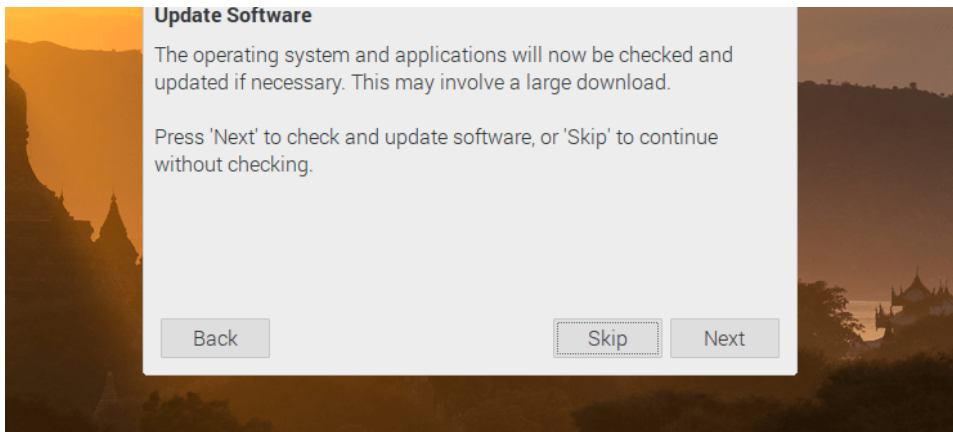
⑤一覧に表示される無線LANからWiFiアクセスポイントを選択して「NEXT」で次へ進む



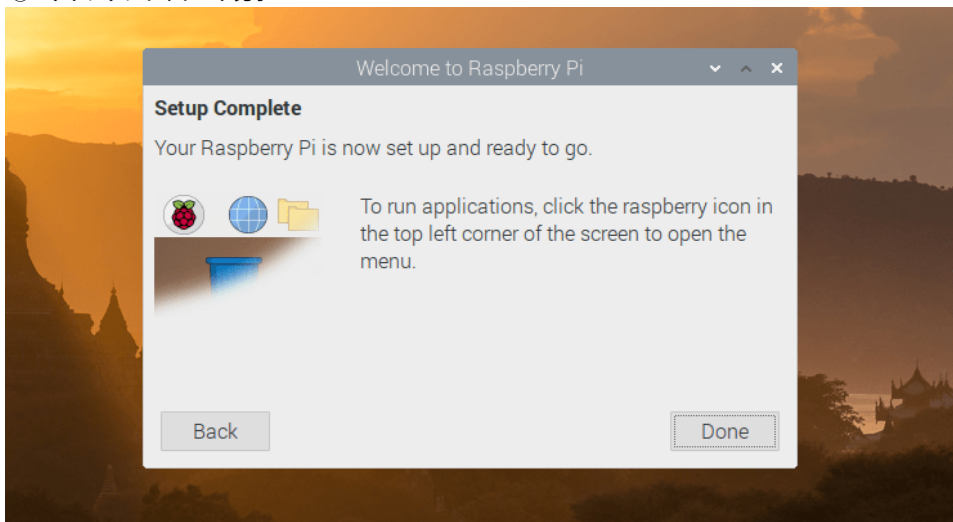
⑥「Skip」で次へ進む

※アップデートに時間がかかる為通常はSkip、時間に余裕がある場合は「Next」で最新版にアップデート。

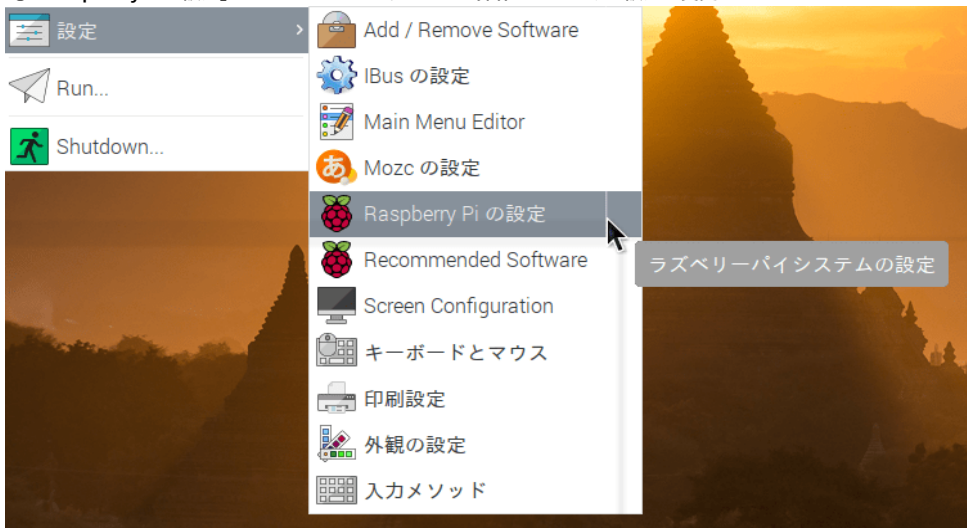




⑦セットアップウィザード完了



⑧「Raspberry Pi の設定」でWindowsPCからリモート操作できるように設定を変更



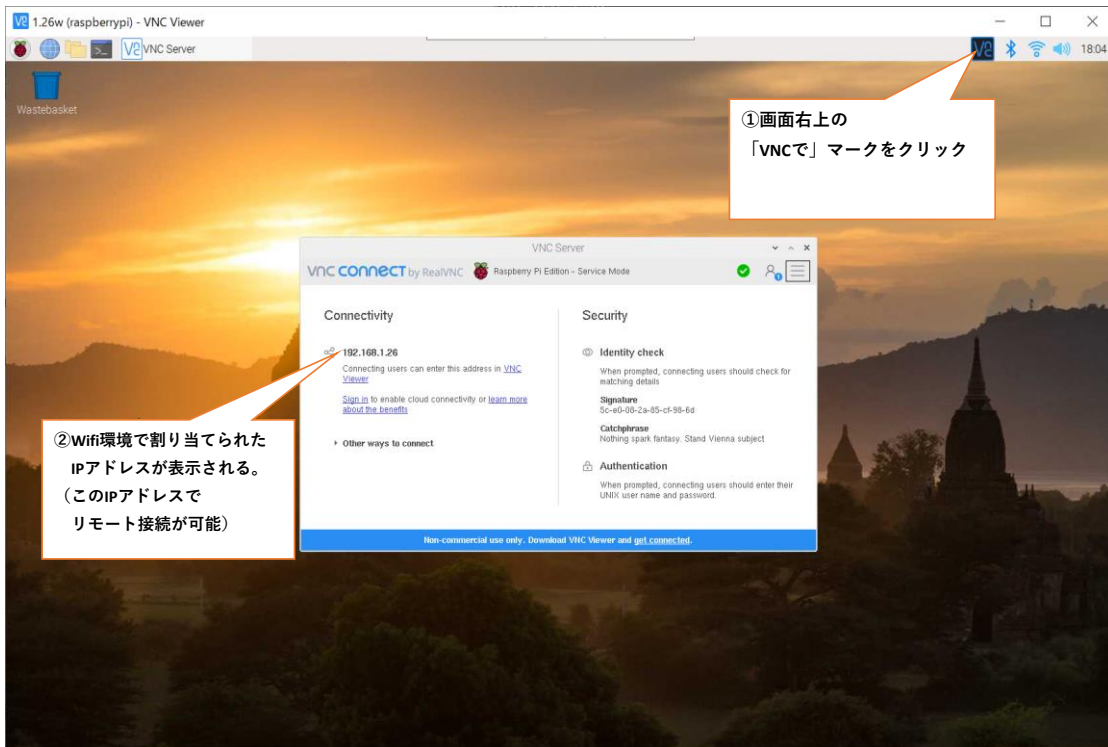
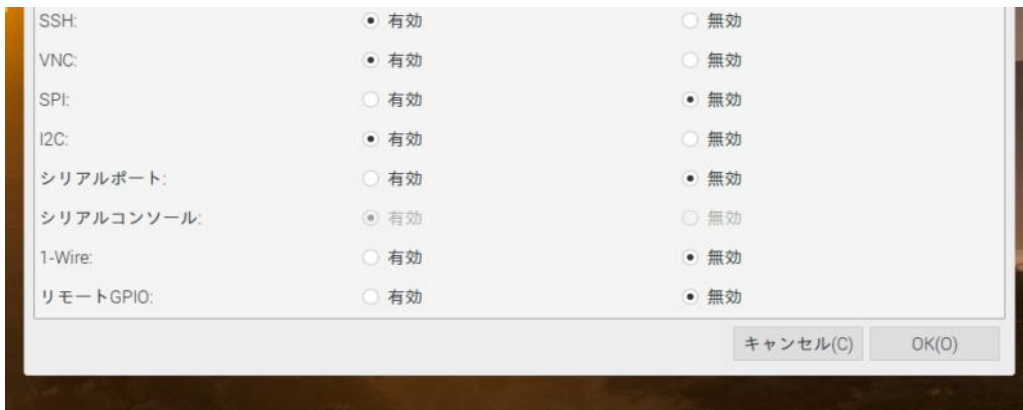
⑨「インターフェース」タブの

【SSH】【VNC】【シリアルポート】を有効に変更

【シリアルコンソール】を無効に設定

「OK」で設定変更を有効化





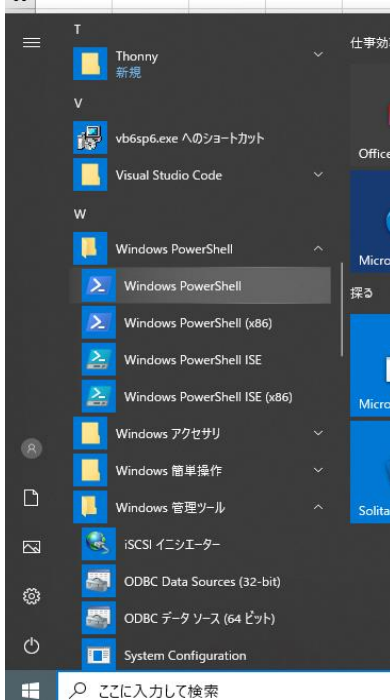
【2】WindowsPCからSSHでリモート接続

Windows標準機能の「PowerShell」でラズベリーパイに接続

※Pythonの実行やライブラリ追加作業はSSHでコマンド実行することで簡単に操作可能です。

CO2センサの操作するまでのセットアップもSSHのコマンド操作で可能です。

① スタートメニューの[Windows PowerShell]から「Windows PowerShell」を起動



② 「ssh pi@192.168.x.xx」SSHコマンドで接続先のIPアドレスを指定して接続

③ カレントフォルダのファイル確認

「ls」

④ カレントフォルダ位置の確認

「pwd」

※SSHからラズベリーパイを停止するコマンド「sudo halt」

```
pi@raspberrypi: ~
PS C:\Users\AdminX1> ssh pi@192.168.1.26
pi@192.168.1.26's password:
Linux raspberrypi 5.4.83+ #1379 Mon Dec 14 13:06:05 GMT 2020 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Mar 5 18:02:06 2021 from 192.168.1.10
pi@raspberrypi:~$ Connection to 192.168.1.26 closed by remote host.
Connection to 192.168.1.26 closed.
PS C:\Users\AdminX1> ssh pi@192.168.1.26
pi@192.168.1.26's password:
Linux raspberrypi 5.4.83+ #1379 Mon Dec 14 13:06:17 GMT 2020 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Mar 5 18:06:17 2021
pi@raspberrypi:~$ ls
Bookshelf Desktop Documents Downloads Music Pictures Public Templates Videos
pi@raspberrypi:~$ pwd
/home/pi
pi@raspberrypi:~$
```

- ① 「ssh pi@192.168.1.26」:接続先IP
- ② 「xxxx」 接続先パスワード
を入力すると
ラズベリーパイのlinuxコマンド
操作が可能になる。

※接続先の認証情報が登録済みの場合は、一度「known_hosts」ファイルを削除すると接続可能になります。

```
pi@raspberrypi: ~  
Connection to 192.168.1.33 closed.  
PS C:\Users\AdminX1> ssh pi@192.168.1.26  
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@  
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!    @  
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@  
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!  
Someone could be eavesdropping on you right now (man-in-the-  
It is also possible that a host key has just been changed.  
The fingerprint for the ECDSA key sent by the remote host is  
SHA256:qQFI18oYjs980jlu48pyd7Tg/EksH2Mb9EzK64SG/ak.  
Please contact your system administrator.  
Add correct host key in C:\Users\AdminX1\.ssh\known_hosts to get rid of this message.  
Offending ECDSA key in C:\Users\AdminX1\.ssh\known_hosts:1  
ECDSA host key for 192.168.1.26 has changed and you have requested strict checking.  
Host key verification failed.  
PS C:\Users\AdminX1> ssh pi@192.168.1.26  
The authenticity of host '192.168.1.26 (192.168.1.26)' can't be established.  
ECDSA key fingerprint is SHA256:qQFI18oYjs980jlu48pyd7Tg/EksH2Mb9EzK64SG/ak.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '192.168.1.26' to the list of known hosts.  
pi@192.168.1.26's password:  
Linux raspberrypi 5.4.83+ #1379  
The programs included with the Debian Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Fri Mar  5 00:33:43 2021  
pi@raspberrypi:~$
```

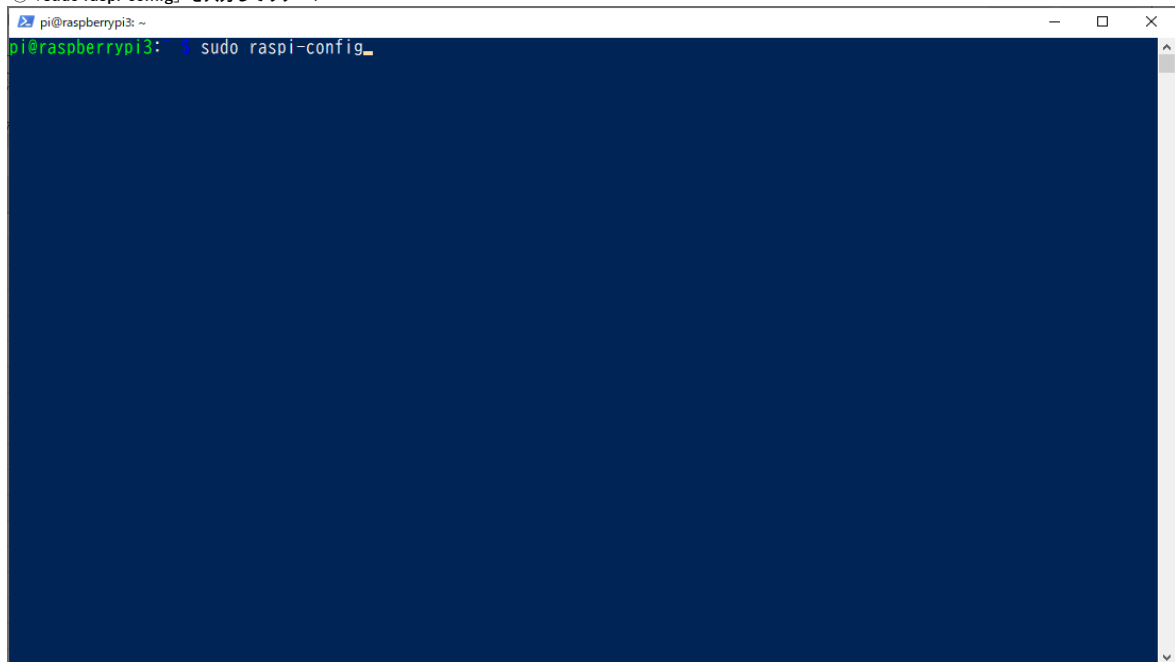
SSH接続時に警告メッセージが出る場合は、登録済みのssh設定ファイルを削除すると、接続可能になる。

- ① 「ssh pi@192.168.1.26」:接続先IP
- ② 「xxxx」 接続先パスワード

を入力すると
ラズベリーパイのlinuxコマンド
操作が可能になる。

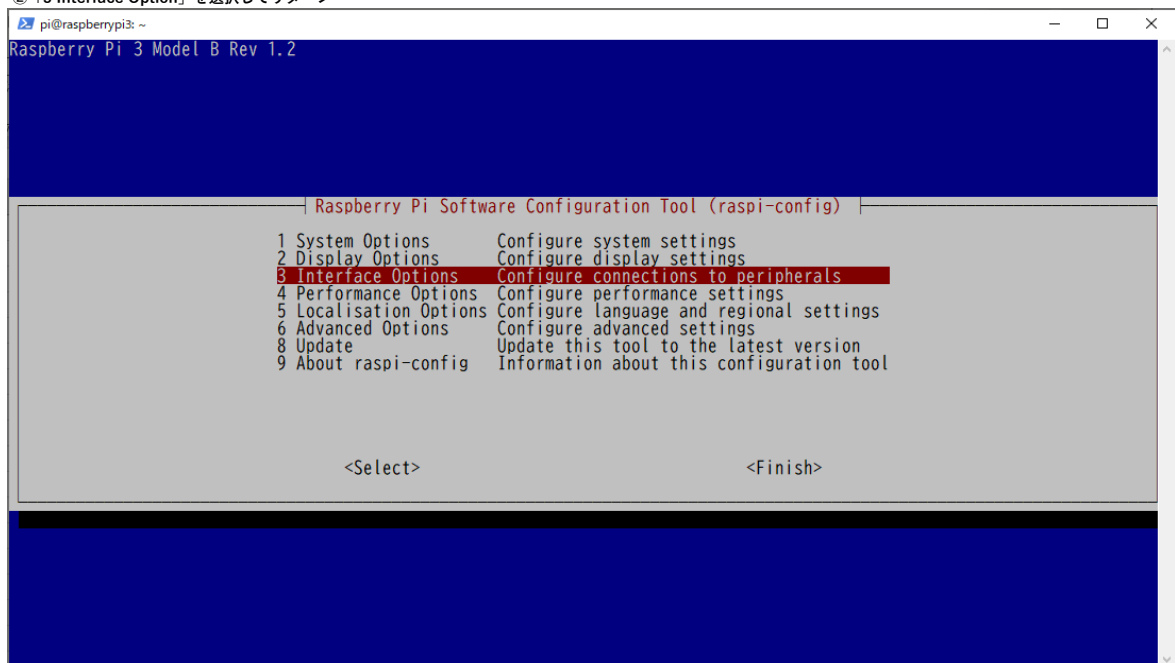
【3】SSHから「raspi-config」でVNCを有効化

① 「sudo raspi-config」を入力してリターン



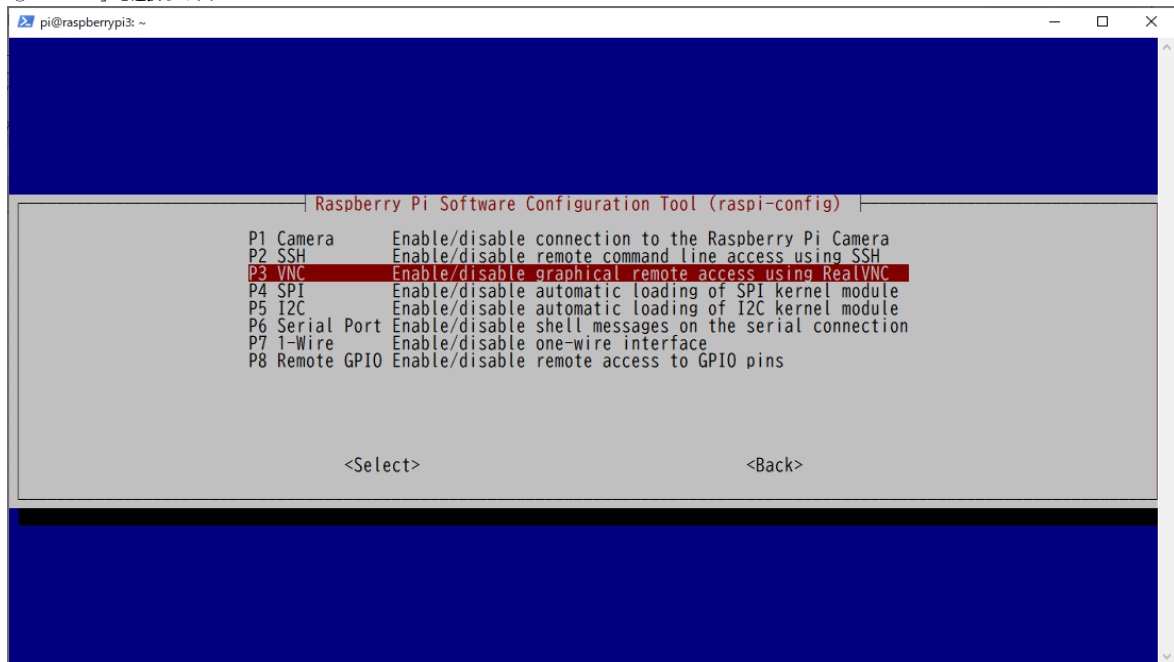
```
pi@raspberrypi3: ~  
pi@raspberrypi3:~$ sudo raspi-config_
```

② 「3 Interface Option」を選択してリターン

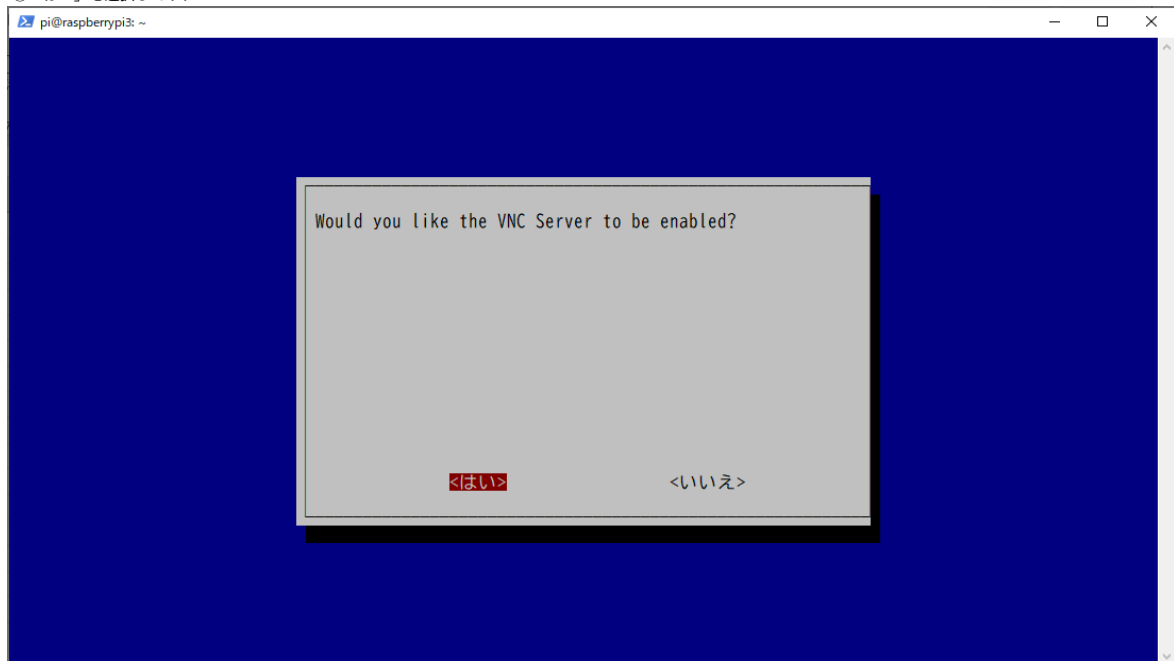


```
pi@raspberrypi3: ~  
Raspberry Pi 3 Model B Rev 1.2  
  
Raspberry Pi Software Configuration Tool (raspi-config)  
1 System Options      Configure system settings  
2 Display Options     Configure display settings  
3 Interface Options   Configure connections to peripherals  
4 Performance Options Configure performance settings  
5 Localisation Options Configure language and regional settings  
6 Advanced Options    Configure advanced settings  
8 Update              Update this tool to the latest version  
9 About raspi-config  Information about this configuration tool  
  
<Select> <Finish>
```

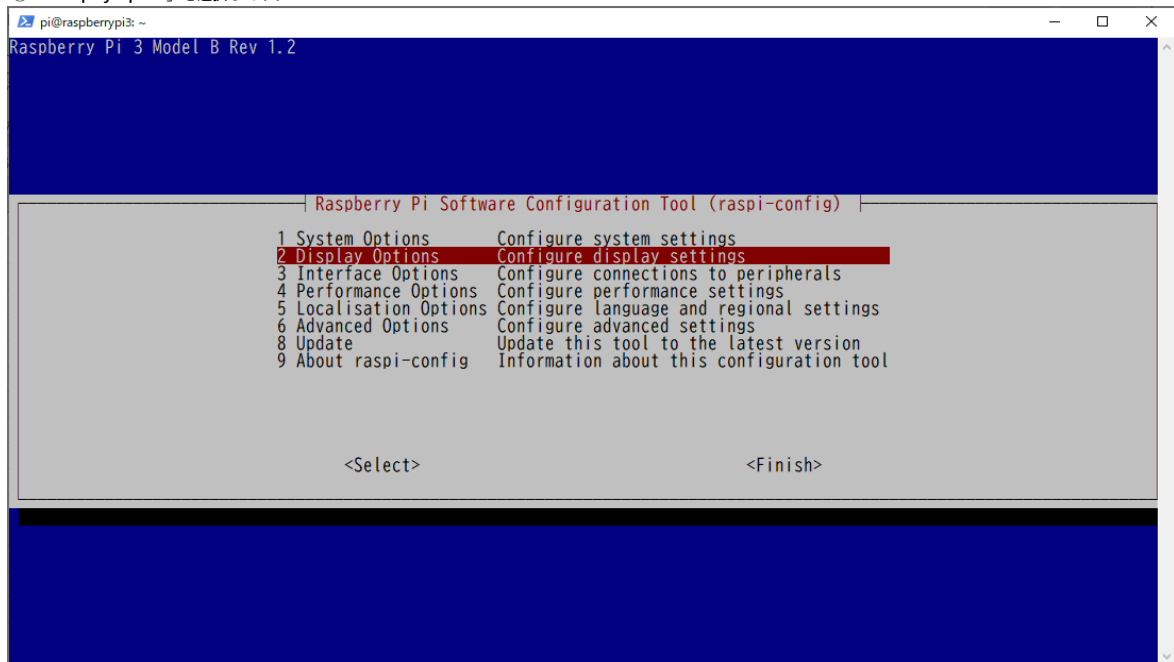
③ 「P3 VNC」を選択してリターン



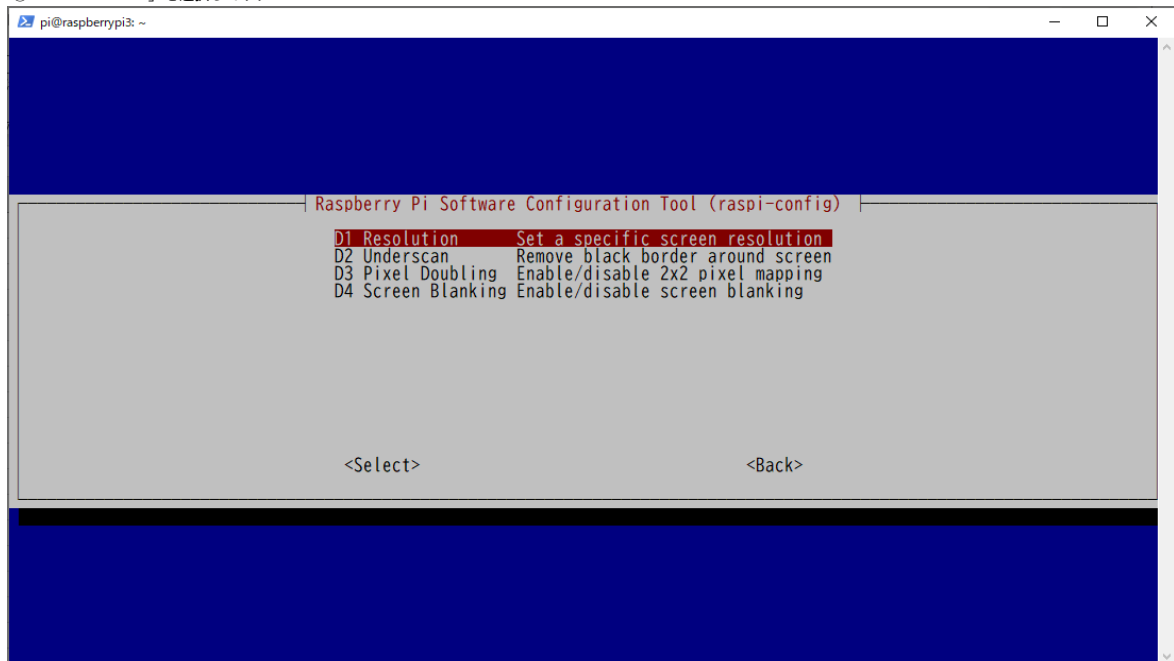
④ 「はい」を選択してリターン



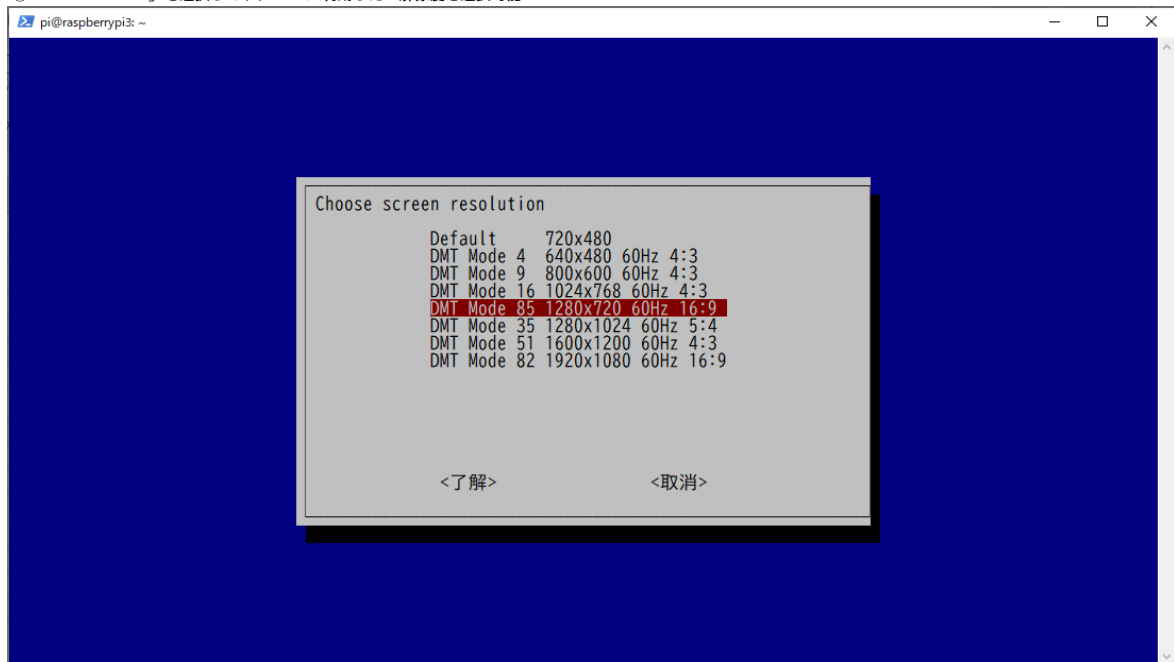
⑤ 「2 Display Option」を選択してリターン



⑥ 「D1 Resolution」を選択してリターン



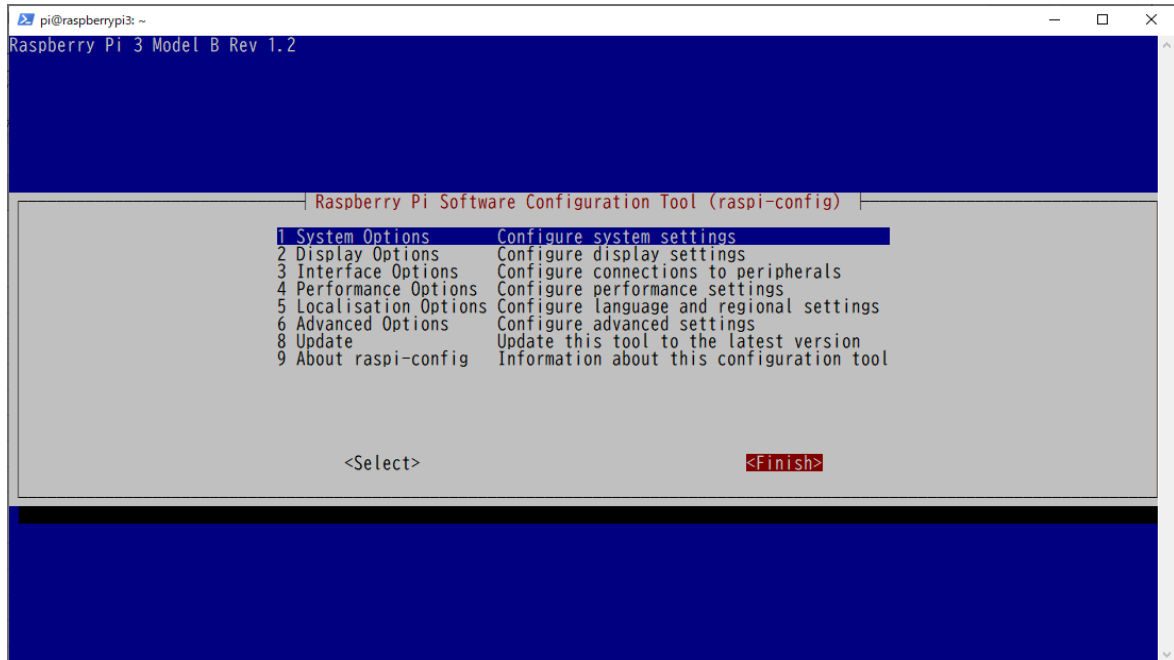
⑦ 「DTM Mode 85」を選択してリターン ※利用したい解像度を選択可能



⑧ 「了解」を選択してリターン



⑨ 「<Finish>」を選択してリターン



【4】WindowsPCからVNCでGUI画面にリモート接続

※WiFiの接続先変更やWiFiパスワードの入力はVNC画面での操作が分かりやすい。

WiFi接続で割り当てられた自分のIPアドレスの確認はGUIでの確認をお勧めします。

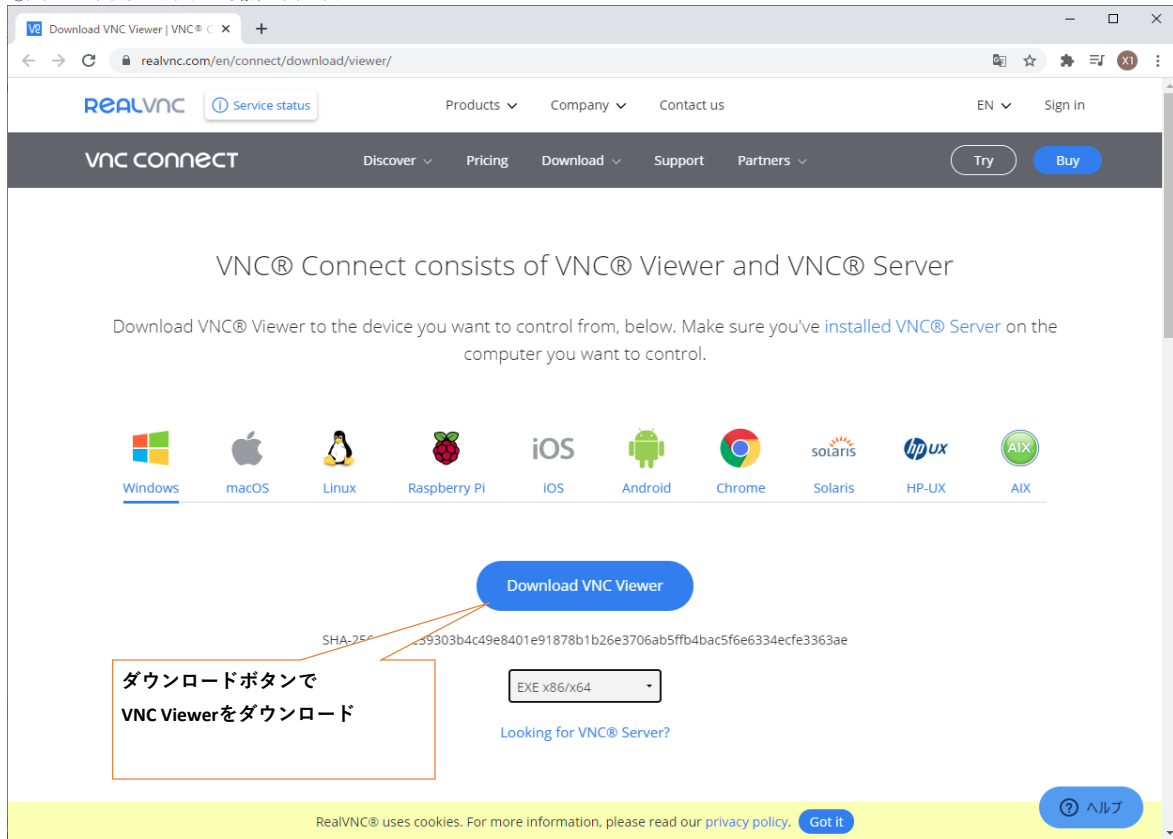
参照URL：

VNC Viewerダウンロードページ

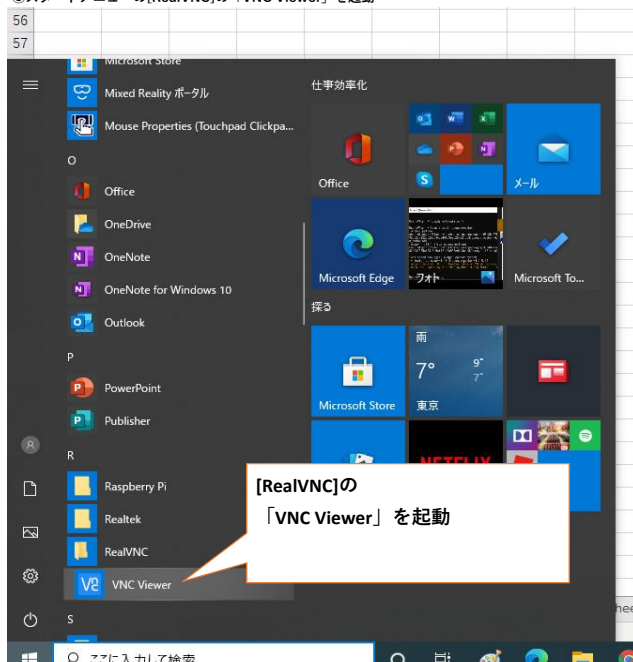
「<https://www.realvnc.com/en/connect/download/viewer/>」

①ダウンロードボタンでVNC Viewerをダウンロード

②ダウンロードしたEXEファイルから標準セットアップ



③スタートメニューの[RealVNC]の「VNC Viewer」を起動



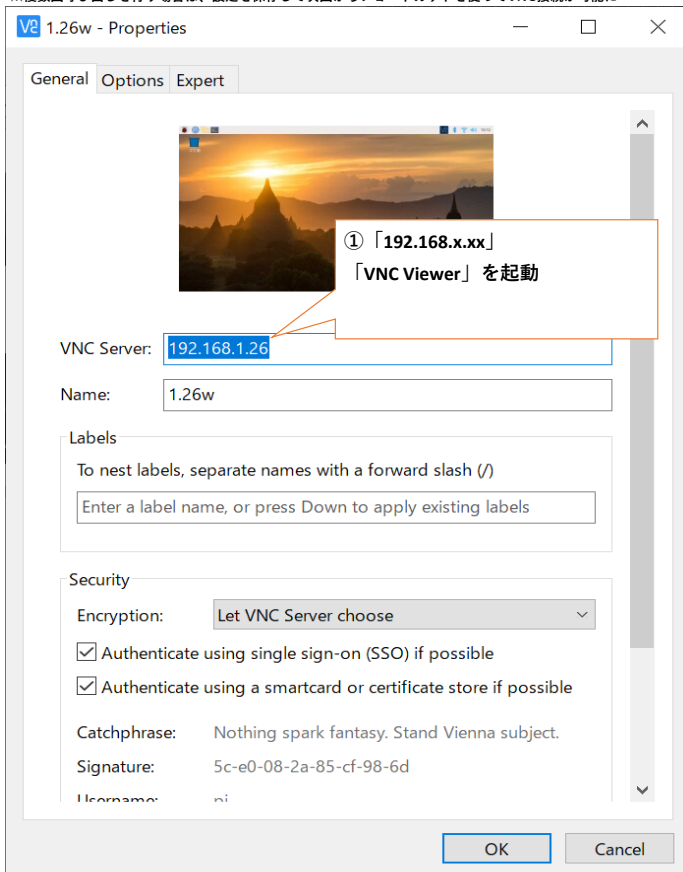
④接続するraspberrypiの「IPアドレス」入力



⑤接続するraspberrypiの「パスワード」を入力



※複数回呼び出しを行う場合は、設定を保存して次回からショートカットを使ってVNC接続が可能に

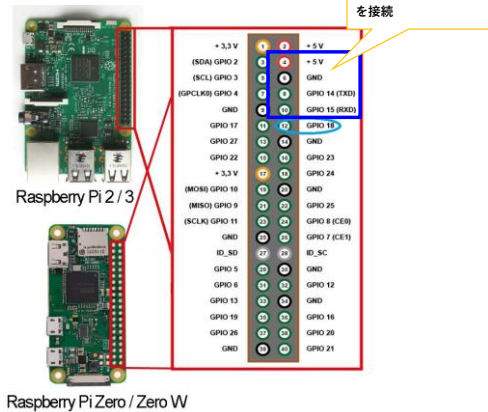


【5】CO2センサ(mh_z19)を利用するまでの手順

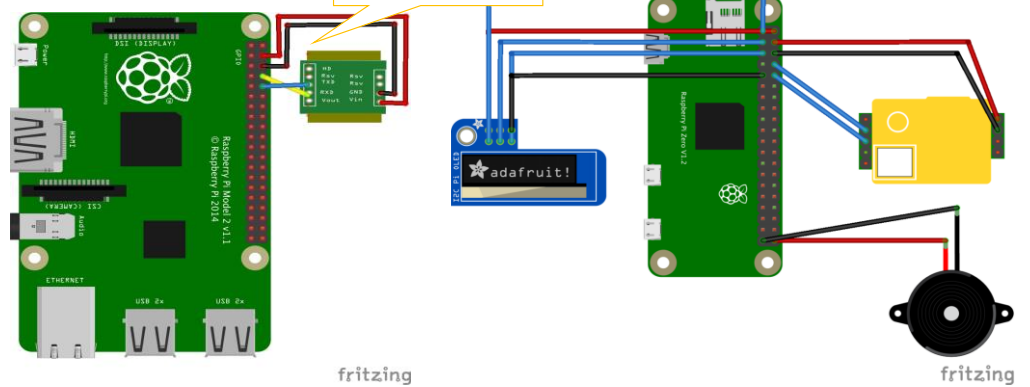
(センサをGPIO接続して測定結果をグラフ表示するまでの手順)

1. CO2センサ(mh_z19)とラズベリーパイのGPIOを接続
2. Pythonライブラリの追加と実行
3. 取得結果可能用のライブラリの追加
4. 取得結果のファイル出力、Web表示用のファイルをラズベリーパイにコピー
5. ファイル送信したバッチコマンドファイルを実行権限を設定して、スケジュール実行設定を登録
6. Pythonの簡易Webサーバ機能でco2、温度のトレンドグラフを表示

1. CO2センサ(mh_z19)とラズベリーパイのGPIOを接続



※ラズベリーパイ3、4の場合



2. Pythonライブラリの追加と実行

①CO2センサ(mh_z19)用のPython3用ライブラリ追加

「sudo pip3 install mh-z19」

②測定結果を取得

「sudo python3 -m mh_z19」

co2の測定値取得

「sudo python3 -m mh_z19 --all」

co2と温度の測定値取得

```

pi@raspberrypi: ~
Last login: Fri Mar  5 18:20:38 2021
Linux raspberrypi 5.4.83+ #1379 Mon Dec 14 13:06:05 GMT 2020 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Mar  5 18:20:38 2021
pi@raspberrypi:~$ sudo pip3 install mh-z19
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting mh-z19
  Downloading https://www.piwheels.org/simple/mh-z19/mh_z19-3.0.1-py3-none-any.whl
Requirement already satisfied: RPi.GPIO in /usr/lib/python3/dist-packages (from mh-z19) (0.7.0)
Collecting getrpimodel (from mh-z19)
  Downloading https://www.piwheels.org/simple/getrpimodel/getrpimodel-0.1.19-py3-none-any.whl
Requirement already satisfied: pyserial in /usr/lib/python3/dist-packages (from mh-z19) (3.4)
Collecting argparse (from mh-z19)
  Downloading https://files.pythonhosted.org/packages/f2/94/3af39d34be01a24a6e65433d19e107099374224905f1e0cc6b
belfd22a2f/argparse-1.4.0-py2.py3-none-any.whl
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (from mh-z19) (2.21.0)
Installing collected packages: getrpimodel, argparse, mh-z19
Successfully installed argparse-1.4.0 getrpimodel-0.1.19 mh-z19-3.0.1
pi@raspberrypi:~$ sudo python3 -m mh_z19
{"co2": 864}
pi@raspberrypi:~$ sudo python3 -m mh_z19 --all
{"co2": 851, "temperature": 50, "TT": 90, "SS": 0, "UHuI": 0}
pi@raspberrypi:~$ sudo apt install -y jq
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... 0%

```

3. 取得結果可能用のライブラリの追加

①jqをインストールする

JSON抽出コマンド。

「sudo apt install -y jq」

4. 取得結果のファイル出力、Web表示用のファイルをラズベリーパイにコピー

【パッチ処理用ファイル】

log2.sh

2021-03-05.log

2021-03-05_single.log

【Web表示用ファイル】

index.html

index_single.html

mychart.js

mychart_single.js

Chart.min.js

Python コマンドを実行して取得された測定結果の「co2」と「Temperature」の値をCSV形式でファイル出力する。

1 日分のco2測定結果をファイル出力

co2測定結果の直近 2 0 回分をファイル出力

1 日分のco2測定結果をWebでグラフ表示

直近20回分のco2測定結果をWebでグラフ表示

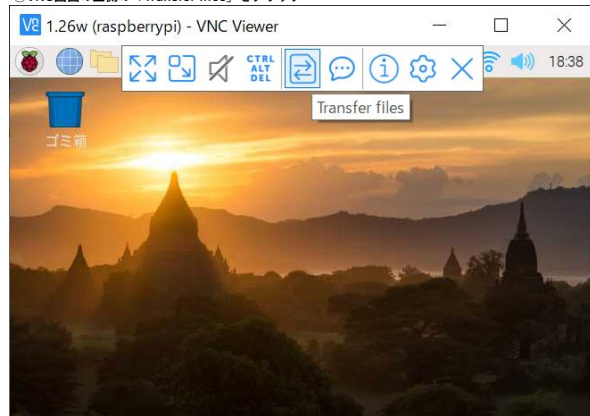
1 日分のco2測定結果グラフ表示用JavaScript

直近20回分のco2測定結果グラフ表示用JavaScript（自動更新機能あり）

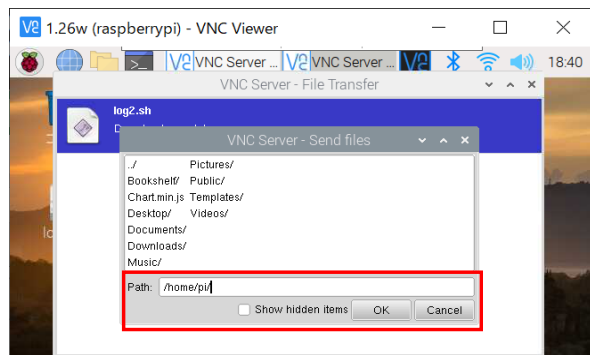
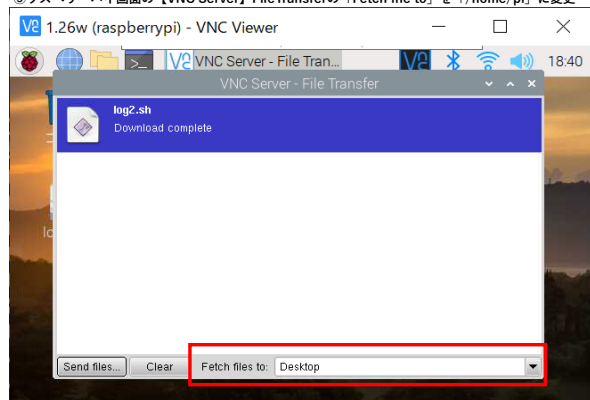
測定結果グラフ表示用JavaScriptライブラリ

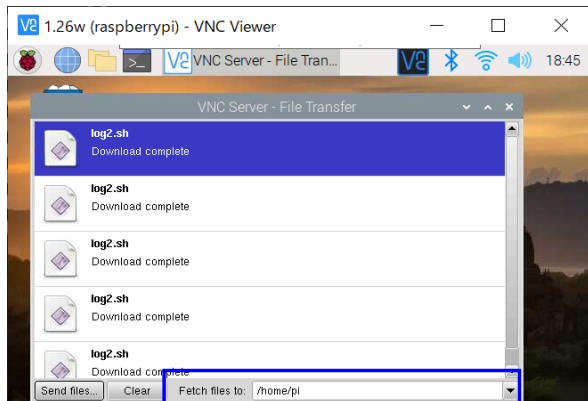
①VNCでラズベリーパイに接続

②VNC画面の上側の「Transfer files」をクリック

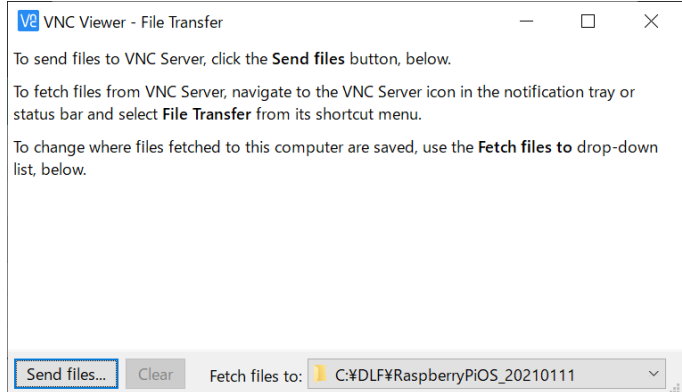


③ラズベリーパイ画面の【VNC Server】FileTransferの「Fetch file to」を「/home/pi」に変更

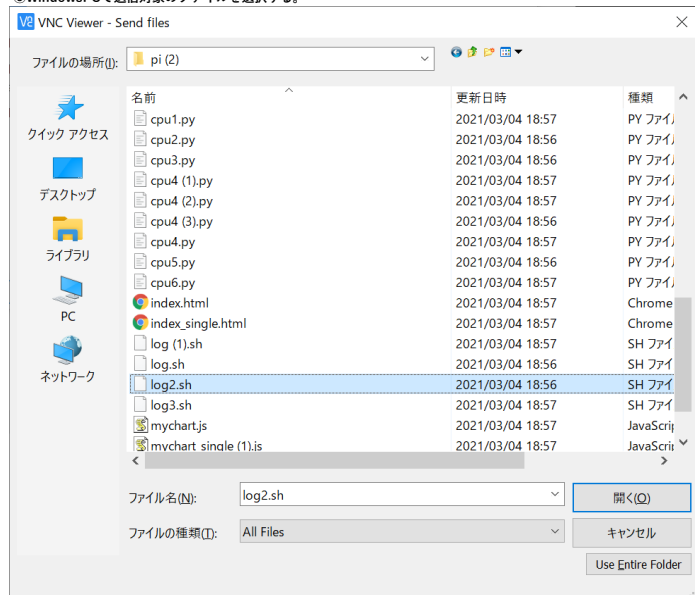




④WindowsPCの【VNC Viewer】FileTransferの「Send files」をクリック



⑤WindowsPCで送信対象のファイルを選択する。



5. ファイル送信したバッチコマンドファイルを実行権限を設定して、スケジュール実行設定を登録

【バッチ処理用ファイル】

log2.sh Python コマンドを実行して取得された測定結果の「co2」と「Temperature」の値をCSV形式でファイル出力する。

#!/bin/sh	
echo "hello world"	※動作確認用のメッセージ
dateA=`date '+%Y-%m-%d %H:%M:%S'`	※年月日時分秒を取得
dateB=`date '+%Y-%m-%d.log'`	※年月日.logファイル名をセット（1日分記録用ファイル）
dateS=`date '+%Y-%m-%d_single.log'`	※年月日_single.logファイル名をセット(部分画面用ファイル)
result=`sudo python3 -m mh_z19   jq '.co2'`	Pythonコマンドでco2の測定結果を取得
result2=`sudo python3 -m mh_z19 --all   jq '.temperature'`	Pythonコマンドでco2、温度の測定結果を取得
echo "\${dateA},\${result},\${result2}" >> "\${dateB}"	※1日分記録用ファイルにco2、気温のデータを追記
tail "\${dateB}" -n 20 > "\${dateS}"	※年月日_single.logファイル名をセット(部分画面用ファイル)
echo "\${dateA},\${result},\${result2}"	

① 「log2.sh」 に実行権限を設定

```
「sudo chmod 755 log2.sh」
```

② 「log2.sh」 がスケジュール実行されるようにcron設定登録

```
「crontab -e」
```

※ ↑ sodoを付けない

③ 実行設定を追加

```
「*/1 * * * * bash ./log2.sh」
```

```
「@reboot bash ./webstart.sh」
```

```
「@reboot sudo nohup python3 -m run_server.py >> /home/pi/web_https.log 2>&1 &」
```

を最終行に貼り付け。

```
「ctrl」 + 「o」
```

```
enter
```

```
「ctrl」 + 「x」
```

```
pi@raspberrypi: ~
pi@raspberrypi:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether b8:27:eb:13:4c:6f brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.26/24 brd 192.168.1.255 scope global dynamic noprefixroute wlan0
        valid_lft 84539sec preferred_lft 73739sec
    inet6 fe80::68e:d227:d903:c000/64 scope link
        valid_lft forever preferred_lft forever
pi@raspberrypi:~$ nohup python3 -m http.server --bind 192.168.1.26 8080 &
[1] 1150
pi@raspberrypi:~$ nohup: 入力を見出し、出力を 'nohup.out' に追記します
pi@raspberrypi:~$ sudo chmod 755 log2.sh
pi@raspberrypi:~$ ls
2021-03-05.log      Chart.min.js  Downloads  Public      index.html    mychart.js
2021-03-05_single.log Desktop      Music      Templates  index_single.html mychart_single.js
Bookshelf          Documents    Pictures    Videos     log2.sh       nohup.out
pi@raspberrypi:~$ crontab -e
no crontab for pi - using an empty one
Select an editor. To change later, run 'select-editor'.
 1. /bin/nano      <---- easiest
 2. /usr/bin/vim.tiny
 3. /bin/ed
Choose 1-3 [1]: 1_
```

「1」 Enter

```
pi@raspberrypi: ~
GNU nano 3.2 /tmp/crontab.50Kzxp/crontab 変更済み

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
*/1 * * * * bash ./log2.sh
```

右クリックで
「*/1 * * * * bash ./log2.sh」を最終行に貼り付け。

「ctrl」+「o」で書き込み

```
pi@raspberrypi: ~
2021-03-05.log Chart.min.js Downloads Public index.html mychart.js
2021-03-05_single.log Desktop Music Templates index_single.html mychart_single.js
Bookshelf Documents Pictures Videos log2.sh nohup.out
pi@raspberrypi:~$ crontab -e
no crontab for pi - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano <----- easiest
 2. /usr/bin/vim.tiny
 3. /bin/ed

Choose 1-3 [1]: 1
crontab: installing new crontab
pi@raspberrypi:~$ ls
2021-03-05.log Chart.min.js Downloads Public index.html mychart.js
2021-03-05_single.log Desktop Music Templates index_single.html mychart_single.js
Bookshelf Documents Pictures Videos log2.sh nohup.out
pi@raspberrypi:~$ cat 2021-03-05.log
2021-03-05 18:46:50,715,50
2021-03-05 18:47:40,715,50
2021-03-05 18:47:48,719,50
2021-03-05 18:57:01,685,38
pi@raspberrypi:~$ date
2021年 3月 5日 金曜日 18:57:26 JST
pi@raspberrypi:~$ cat 2021-03-05.log
2021-03-05 18:46:50,715,50
2021-03-05 18:47:40,715,50
2021-03-05 18:47:48,719,50
2021-03-05 18:57:01,685,38
2021-03-05 18:58:01,688,37
pi@raspberrypi:~$
```

スケジュール出力された
時間、co2、温度のcsvデータ

6. Pythonの簡易Webサーバ機能でco2、温度のトレンドグラフを表示

【Web表示用ファイル】

index.html

1 日分のco2測定結果をWebでグラフ表示

index_single.html

直近20回分のco2測定結果をWebでグラフ表示

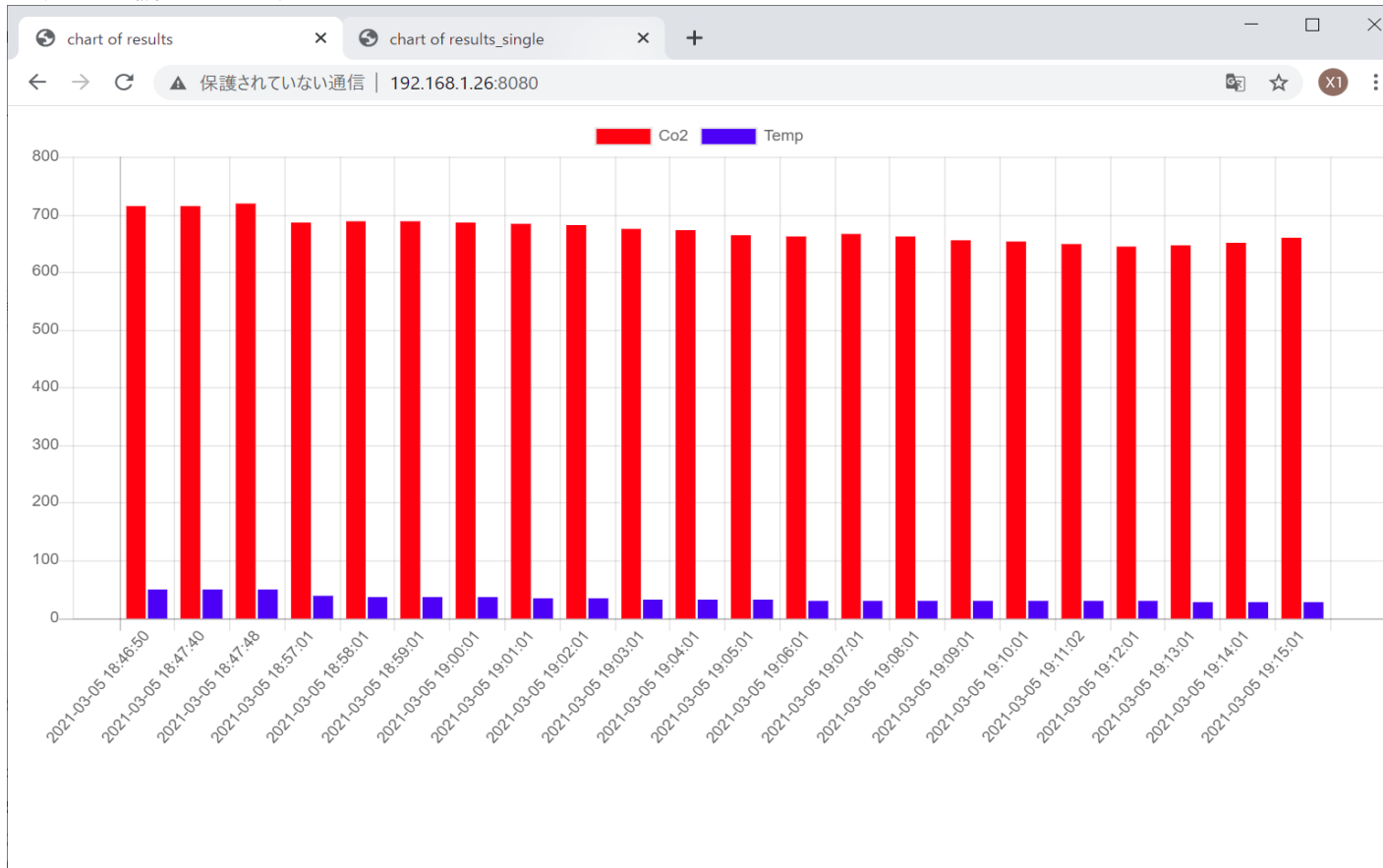
①WebサーバプロセスをIPアドレスを指定して実行

```
「nohup python3 -m http.server --bind 192.168.1.26 8080 &」
```

Enter

※IPアドレスにはラズベリーのIPアドレスをセット

1 日分のco2測定結果をWebでグラフ表示



直近20回分のco2測定結果をWebでグラフ表示

