

# Raspberry Piを使用した オシロスコープの制御

アプリケーション・ノート



## はじめに

Raspberry Piは本来、学生にコンピュータ科学を教えるためのツールとして使用された、シングルボードの小型コンピュータです。その後、小型、低コスト、モジュール対応、オープン・デザインにより普及が進みました。Raspberry Piは、改訂のたびに機能が追加され、現在では教育以外の用途にも広く利用されています。

コンピュータ能力が限られているため、Raspberry Piは多くの分野で通常のPCに取って代わることはありません。しかし、コンパクトなサイズ、柔軟なI/Oインターフェース、低コスト、Pythonの組み込み対応により、ラボのテスト・ベンチや製造現場のテスト・ラックの自動化、計測器の制御、波形データや計測結果の取込み、計測器のデータ管理やリモート・アクセスのハブとして機能する、理想的なプラットフォームとなります。

このアプリケーション・ノートでは、Raspberry Piを迅速にセットアップし、テクトロニクスの2シリーズMSO（ミックスド・シグナル・オシロスコープ）を自動化し、リモート制御するための設定方法をご紹介します。セットアップ方法については、[ビデオ](#)でもご覧いただけます。

## Raspberry Piのセットアップ

ラボの作業ベンチのコントローラPCとしてRaspberry Piを使用するための設定はシンプルです。

### 基本要件とセットアップ

- Raspberry Pi 4とRaspberry Pi OS (旧Raspbian)
- Python 3.7以上
- PyUSB 1.2.1
- PyVISA 1.11.3
- PyVISA-py 0.5.2

オシロスコープとRaspberry Piの通信サポートは、pyVISAをベースとしています。

セットアップを始める前に、Raspberry Piのソフトウェア・コンポーネントが最新であることを確認します。最新でない場合は、Raspberry Piをネットワークに接続して、ソフトウェアのアップデートを行ってください。



図1. テクトロニクスの2シリーズMSOとRaspberry Piのセットアップ例

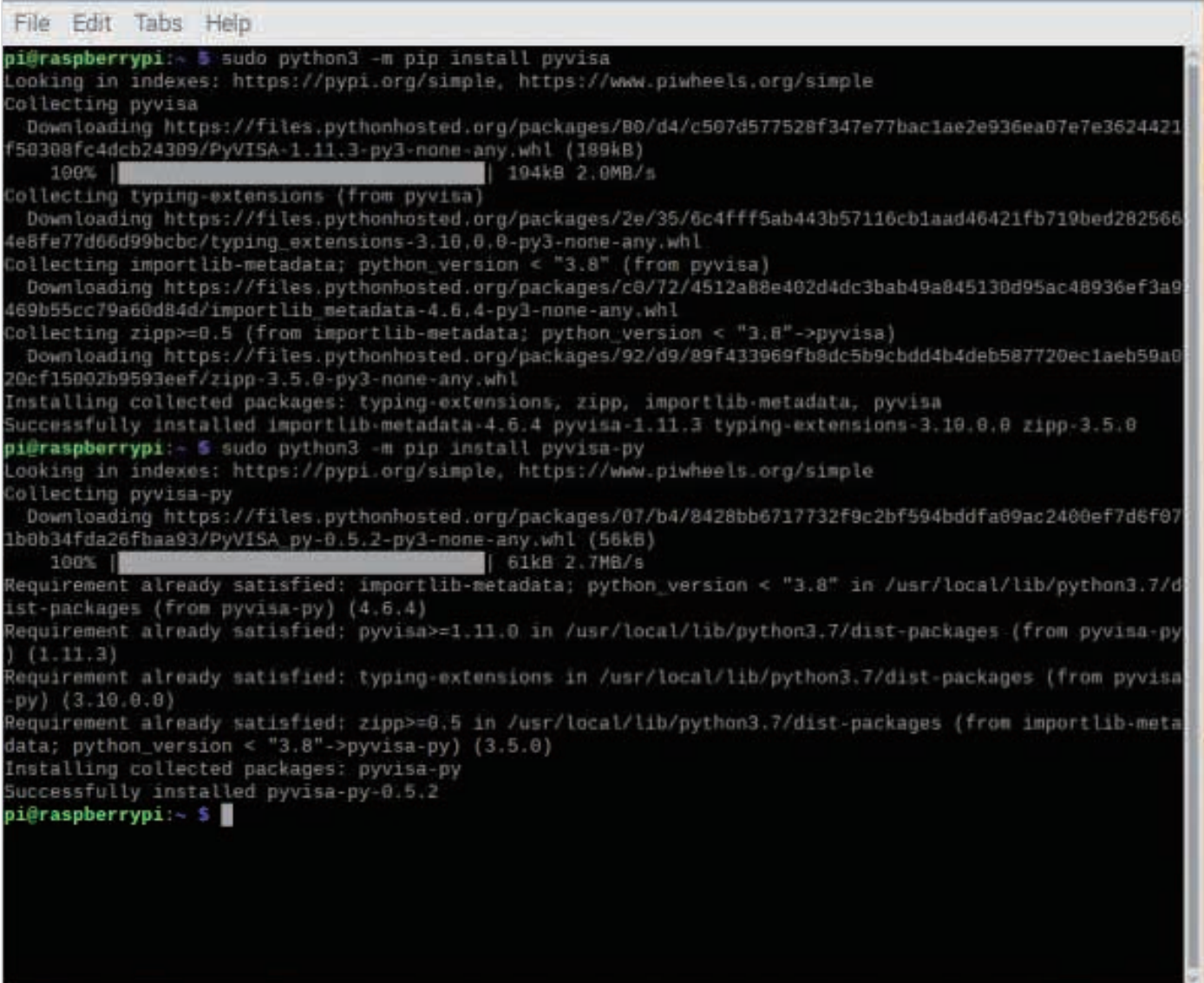
コマンド・プロンプトで以下のように入力します。

```
sudo apt update && sudo apt upgrade -y
```

システムが最後にアップデートされたタイミングにより、アップデートに数分以上かかる場合があります。

セットアップには、Python 3.xのモジュールがいくつかが必要です。必要なモジュールすべてをインストールするため、コマンド・プロンプトで以下のアップデート用コマンドを入力します。

- `sudo python3 -m pip install pyvisa`
- `sudo python3 -m pip install pyvisa-py`
- `sudo python3 -m pip install PyUSB`



```
File Edit Tabs Help
pi@raspberrypi:~$ sudo python3 -m pip install pyvisa
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting pyvisa
  Downloading https://files.pythonhosted.org/packages/B0/d4/c507d577528f347e77bac1ae2e936ea07e7e3624421f50308fc4dcb24309/PyVISA-1.11.3-py3-none-any.whl (189kB)
    100% |#####| 104kB 2.0MB/s
Collecting typing-extensions (from pyvisa)
  Downloading https://files.pythonhosted.org/packages/2e/35/6c4fff5ab443b57116cb1aad46421fb719bed2825664e8fe77d66d99bcbc/typing_extensions-3.10.0.0-py3-none-any.whl
Collecting importlib-metadata; python_version < "3.8" (from pyvisa)
  Downloading https://files.pythonhosted.org/packages/c0/72/4512a88e402d4dc3bab49a845130d95ac48936ef3a9469b55cc79a60d84d/importlib_metadata-4.6.4-py3-none-any.whl
Collecting zipp>=0.5 (from importlib-metadata; python_version < "3.8"->pyvisa)
  Downloading https://files.pythonhosted.org/packages/92/d9/89f433969fb8dc5b9cbdd4b4deb587720ec1aeb59a020cf15002b9593eef/zipp-3.5.0-py3-none-any.whl
Installing collected packages: typing-extensions, zipp, importlib-metadata, pyvisa
Successfully installed importlib-metadata-4.6.4 pyvisa-1.11.3 typing-extensions-3.10.0.0 zipp-3.5.0
pi@raspberrypi:~$ sudo python3 -m pip install pyvisa-py
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting pyvisa-py
  Downloading https://files.pythonhosted.org/packages/07/b4/8428bb6717732f9c2bf594bddfa69ac2400ef7d6f071b0b34fda26fbaa93/PyVISA_py-0.5.2-py3-none-any.whl (56kB)
    100% |#####| 61kB 2.7MB/s
Requirement already satisfied: importlib-metadata; python_version < "3.8" in /usr/local/lib/python3.7/dist-packages (from pyvisa-py) (4.6.4)
Requirement already satisfied: pyvisa>=1.11.0 in /usr/local/lib/python3.7/dist-packages (from pyvisa-py) (1.11.3)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packages (from pyvisa-py) (3.10.0.0)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata; python_version < "3.8"->pyvisa-py) (3.5.0)
Installing collected packages: pyvisa-py
Successfully installed pyvisa-py-0.5.2
pi@raspberrypi:~$
```

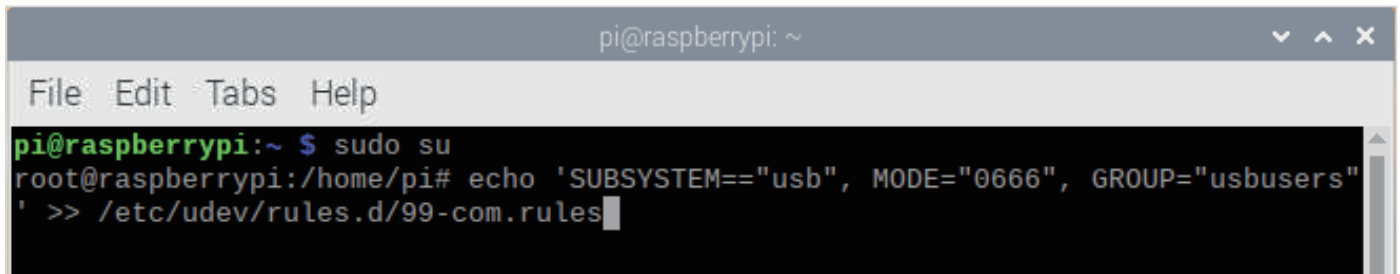
図2. 機器制御に必要なPython 3モジュール



Raspberry Piは、rootユーザしかUSBドライブにアクセスできない場合があります。すべてのユーザがアクセスできるように、Raspberry Piのルールを変更します。

コマンド・プロンプトで以下のように入力します。

- `sudo su`
- `echo 'SUBSYSTEM=="usb", MODE="0666", GROUP="usbusers" >> /etc/udev/rules.d/99-com.rules`
- `exit`



```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo su
root@raspberrypi:/home/pi# echo 'SUBSYSTEM=="usb", MODE="0666", GROUP="usbusers"
' >> /etc/udev/rules.d/99-com.rules
```

図3. すべてのユーザがUSBドライブにアクセスできるようにルールを変更する

変更を有効にするためにRaspberry Piを再起動します。コマンド・プロンプトで以下のように入力します。

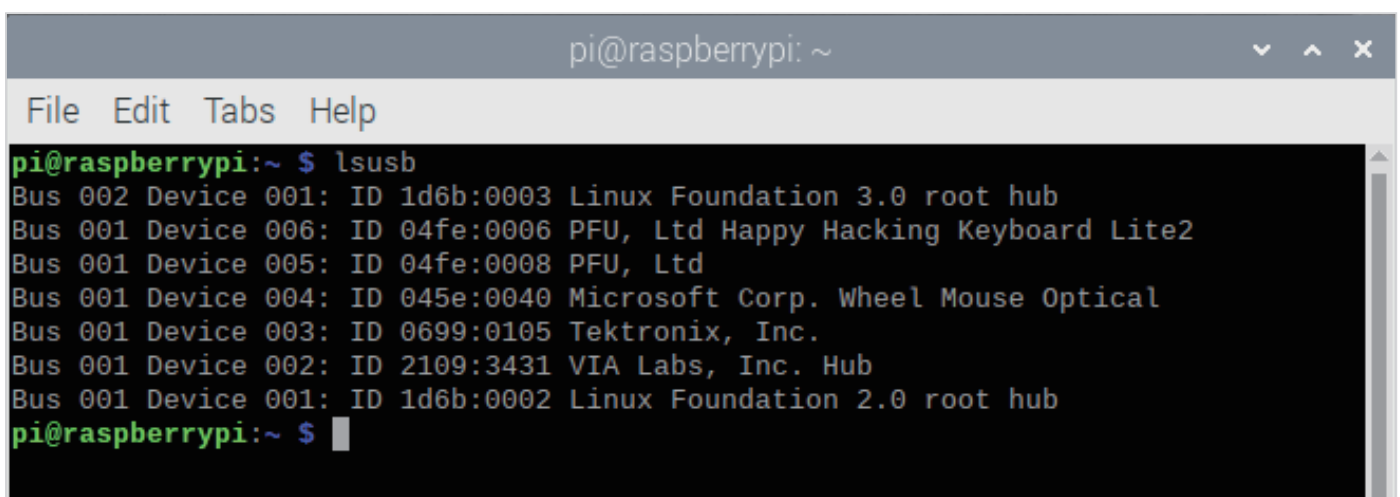
- `sudo reboot`

## テクトロニクス製の2シリーズMSOとの接続をセットアップします。

ほとんどのエントリーレベルのオシロスコープには、接続のためのUSBデバイスが装備されています。以下の手順でRaspberry Piと2シリーズMSOを接続します。

- オシロスコープ側面にあるUSBデバイス・ポートとRaspberry Piを接続します。
- Raspberry Piが2シリーズMSOを認識できるかチェックします。コマンド・プロンプトで以下のように入力します。

- `lsusb`



```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 006: ID 04fe:0006 PFU, Ltd Happy Hacking Keyboard Lite2
Bus 001 Device 005: ID 04fe:0008 PFU, Ltd
Bus 001 Device 004: ID 045e:0040 Microsoft Corp. Wheel Mouse Optical
Bus 001 Device 003: ID 0699:0105 Tektronix, Inc.
Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
pi@raspberrypi:~ $
```

図4. 接続されたUSBデバイスのベンダとして「Tektronix, Inc.」が表示されている

「Tektronix, Inc.」デバイスは、オシロスコープを意味しています。Raspberry Piがテクトロニクス製のデバイスを検出しない場合は、別のUSBポートまたはケーブルで同じ手順を繰り返します。

Raspberry Piとオシロスコープの通信を確認するために、Python 3.0を起動します。コマンド・プロンプトで以下のように入力します。

- `python3`

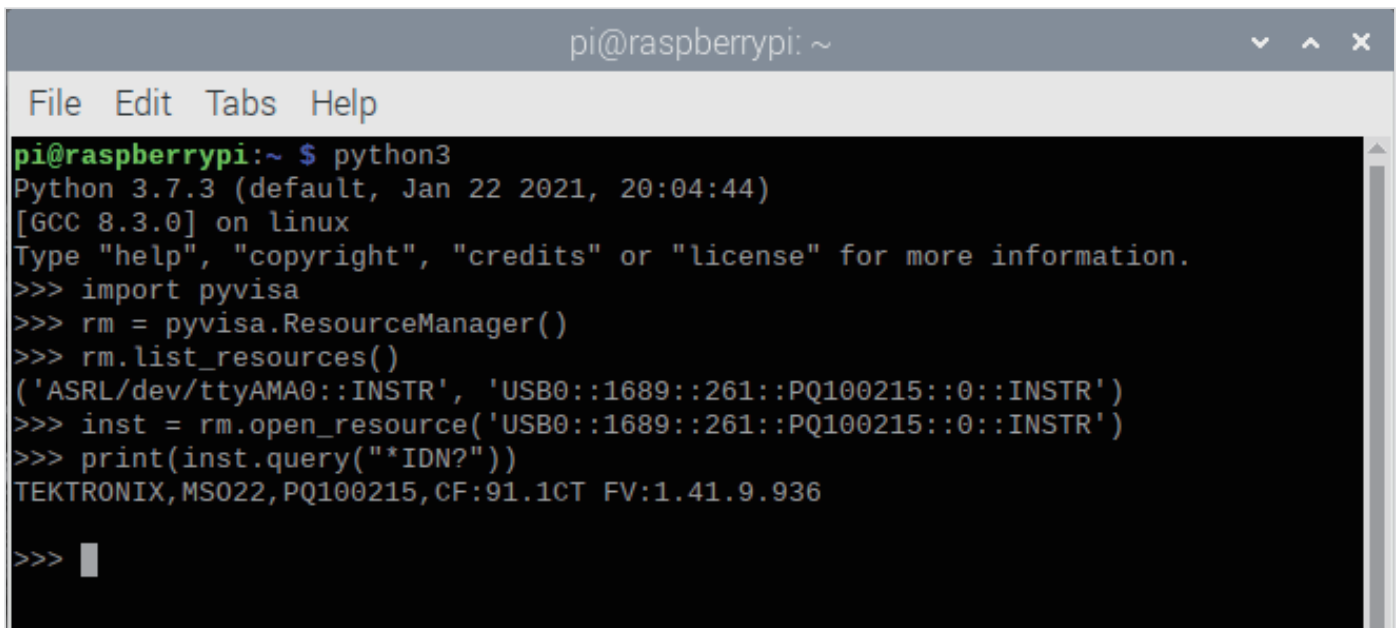
次に、以下を入力してオシロスコープのVISAディスクリプタをチェックします。

- `>>> import pyvisa`
- `>>> rm = pyvisa.ResourceManager()`
- `>>> rm.list_resources()`
- `('ASRL/dev/ttyAMA0::INSTR', 'USB0::1689::261::PQ100125::0::INSTR')`
- `>>> inst = rm.open_resource('USB0::1689::261::PQ100125::0::INSTR')`
- `>>> print(inst.query("*IDN?"))`

`rm.list_resources()`には、VISAディスクリプタが表示されています。正しく表示されていることを確認して以下のように入力します。

- `inst = rm.open_resource(<VISA descriptor>)` と入力してRaspberry Piとオシロスコープを接続します。

通信を確認するために\*IDN?クエリを入力します。リターン・ストリングで正しい型名、シリアル番号が表示されていれば、Raspberry Piがオシロスコープと通信できたことを示しています。(以下の図5参照)



```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ python3
Python 3.7.3 (default, Jan 22 2021, 20:04:44)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import pyvisa
>>> rm = pyvisa.ResourceManager()
>>> rm.list_resources()
('ASRL/dev/ttyAMA0::INSTR', 'USB0::1689::261::PQ100215::0::INSTR')
>>> inst = rm.open_resource('USB0::1689::261::PQ100215::0::INSTR')
>>> print(inst.query("*IDN?"))
TEKTRONIX,MSO22,PQ100215,CF:91.1CT FV:1.41.9.936
>>> █
```

図5. \*IDN?クエリ・コマンドで通信を検証する

2シリーズMSOだけでなく、テクトロニクスのTBS2000Bシリーズ、TBS1000Cシリーズ・デジタル・オシロスコープなどのエントリーレベルのオシロスコープもRaspberry Piに対応しています。

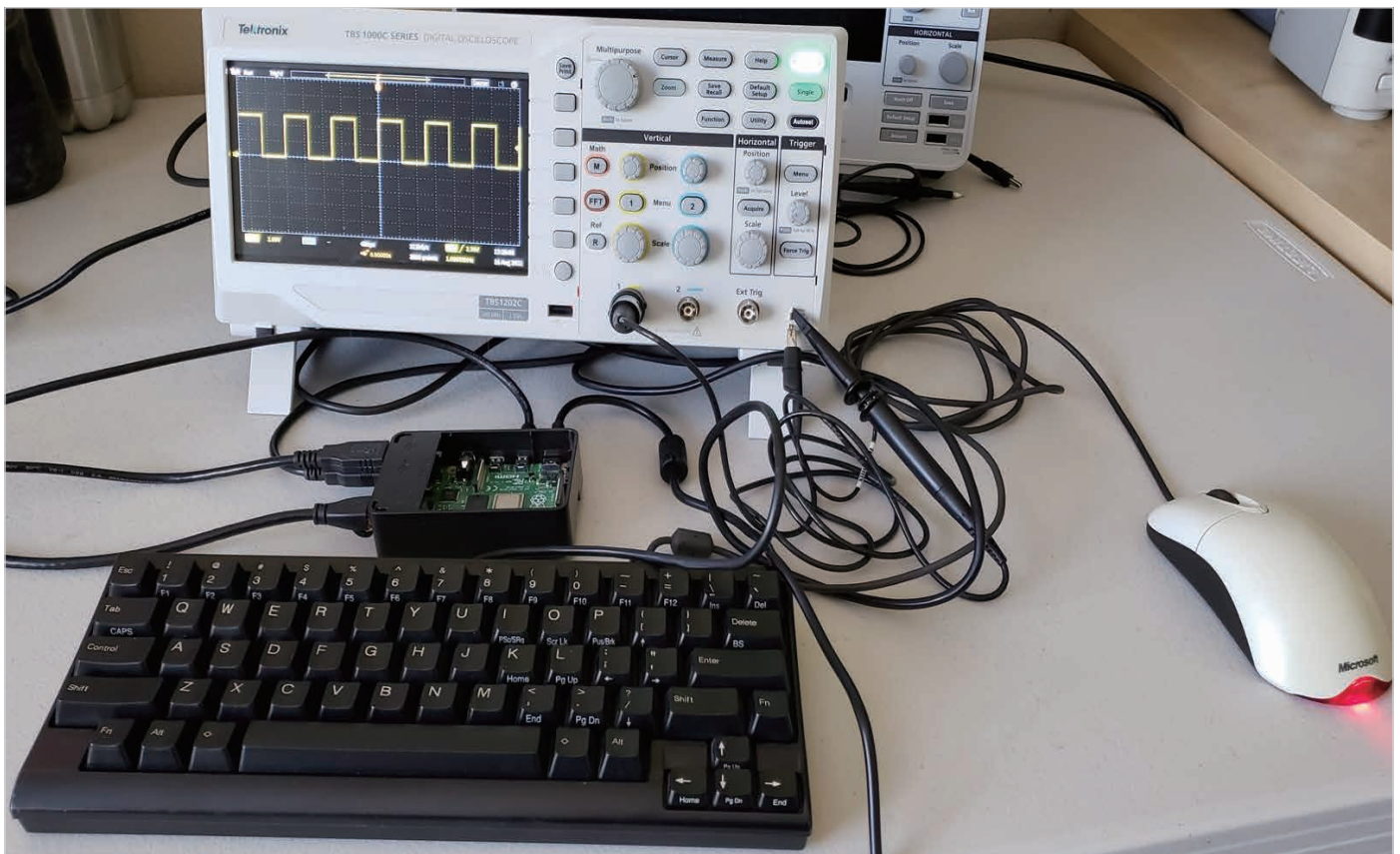


図6. テクトロニクスのTBS1000Cシリーズ・デジタル・オシロスコープとの接続例

### サンプル・スクリプト

以下は、波形データを取得し、プロットするPythonのスクリプト例です。このサンプル・スクリプトは、example\_script.txtというファイル名でこのPDFファイルに添付されているので、ダウンロードしてコピーすることもできます。

```
import time # std module
import pyvisa as visa # http://github.com/hgrecco/pyvisa - pyvisa for connectivity
import matplotlib.pyplot as plt # http://matplotlib.org/ - for plotting
import numpy as np # http://www.numpy.org

# VISA descriptor to identify the test and measurement device
# Please update the VISA descriptor from the query result from pyvisa
visa_address = 'USB0::1689::261::Q300209::0::INSTR'

rm = visa.ResourceManager()
scope = rm.open_resource(visa_address)
scope.timeout = 10000 # ms
scope.encoding = 'latin_1'
scope.read_termination = '\n'
scope.write_termination = None
scope.write('*cls') # clear ESR
scope.write('header OFF')
```

```
# acquisition
scope.write('acquire:state OFF') # stop
scope.write('acquire:stopafter SEQUENCE;state ON') # single
r = scope.query('*opc?')

# curve configuration
scope.write('data:encdg SRIBINARY') # signed integer
scope.write('data:source CH1')
scope.write('data:start 1')
acq_record = int(scope.query('horizontal:recordlength?'))
scope.write('data:stop {}'.format(acq_record))
scope.write('wfmmoutpre:byt_n 1') # 1 byte per sample

# data query
bin_wave = scope.query_binary_values('curve?', datatype='b', container=np.array, chunk_size =
1024**2)

# retrieve scaling factors
wfm_record = int(scope.query('wfmmoutpre:nr_pt?'))
pre_trig_record = int(scope.query('wfmmoutpre:pt_off?'))
t_scale = float(scope.query('wfmmoutpre:xincr?'))
t_sub = float(scope.query('wfmmoutpre:xzero?')) # sub-sample trigger correction
v_scale = float(scope.query('wfmmoutpre:ymult?')) # volts / level
v_off = float(scope.query('wfmmoutpre:yzero?')) # reference voltage
v_pos = float(scope.query('wfmmoutpre:yoff?')) # reference position (level)

# disconnect
scope.close()
rm.close()

# create scaled vectors
# horizontal (time)
total_time = t_scale * wfm_record
t_start = (-pre_trig_record * t_scale) + t_sub
t_stop = t_start + total_time
scaled_time = np.linspace(t_start, t_stop, num=wfm_record, endpoint=False)
# vertical (voltage)
unscaled_wave = np.array(bin_wave, dtype='double') # data type conversion
scaled_wave = (unscaled_wave - v_pos) * v_scale + v_off

# plotting
plt.plot(scaled_time, scaled_wave)
plt.title('channel 1') # plot label
plt.xlabel('time (seconds)') # x label
plt.ylabel('voltage (volts)') # y label
print("look for plot window...")
```

```
plt.show()
```

## TightVNCのセットアップ(オプション)

これは、Raspberry PiでVNCを設定し、リモートで接続したいユーザのためのオプションです。

最新のバージョンにアップデートするため、コマンド・プロンプトで次のように入力します。

- `sudo apt update && sudo apt upgrade -y`

次に、コマンド・プロンプトで次のように入力してVNCサーバをインストールします。

- `sudo apt install tightvncserver`

コマンド・プロンプトで次のように入力してVNCサーバを初期セットアップします。

- `vncserver`

初期セットアップであるため、コマンド・プロンプトでパスワードを求められます。8文字のパスワードを入力します。パスワードは自動的に8文字に短縮されます。

確認のために再度パスワードを入力します。

Viewerのみに適用するパスワードかを聞かれた場合は、Noを選択します。

PCにおいて、[tightvnc.com](http://tightvnc.com)でTightVNCクライアントをインストールします。

インストールしたならばTightVNC Viewerを起動します。接続ウィンドウで、Raspberry PiのIPアドレスとデフォルトのポート番号(5901)を入力します。

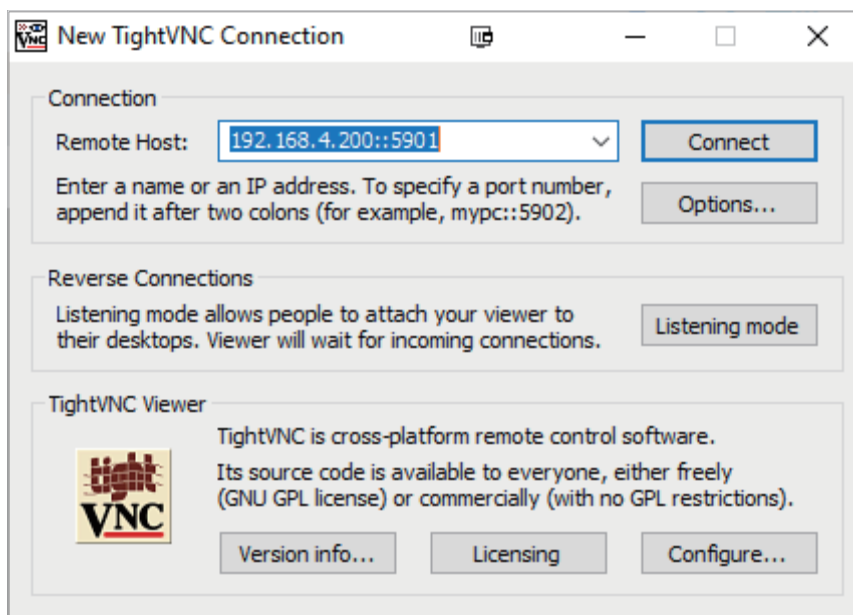
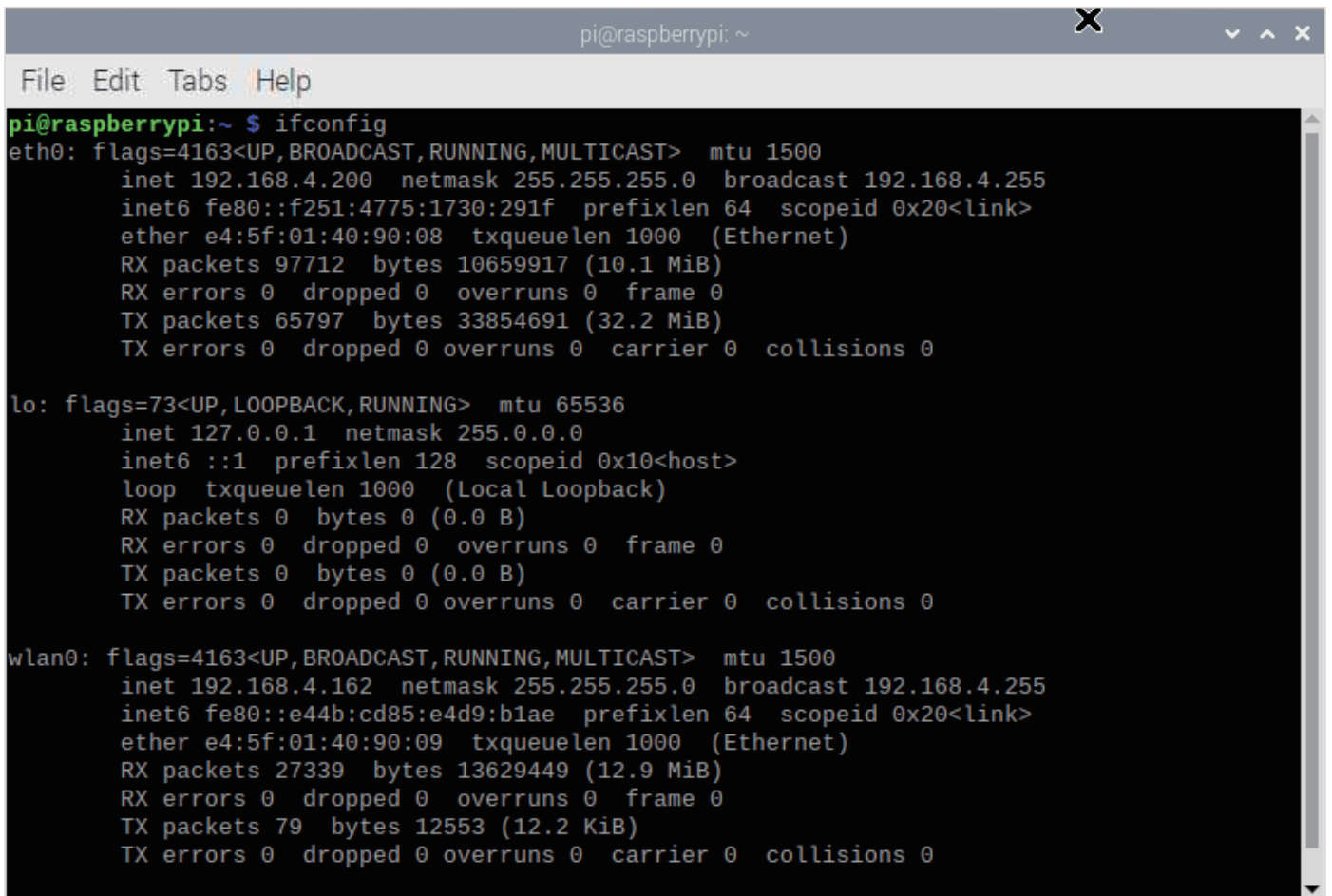


図7. TightVNCビューアの接続ウィンドウ



Raspberry PiのIPアドレスを調べるには、**ifconfig**コマンドを使用します。



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
  inet 192.168.4.200 netmask 255.255.255.0 broadcast 192.168.4.255  
  inet6 fe80::f251:4775:1730:291f prefixlen 64 scopeid 0x20<link>  
  ether e4:5f:01:40:90:08 txqueuelen 1000 (Ethernet)  
  RX packets 97712 bytes 10659917 (10.1 MiB)  
  RX errors 0 dropped 0 overruns 0 frame 0  
  TX packets 65797 bytes 33854691 (32.2 MiB)  
  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 (Local Loopback)  
  RX packets 0 bytes 0 (0.0 B)  
  RX errors 0 dropped 0 overruns 0 frame 0  
  TX packets 0 bytes 0 (0.0 B)  
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
  inet 192.168.4.162 netmask 255.255.255.0 broadcast 192.168.4.255  
  inet6 fe80::e44b:cd85:e4d9:b1ae prefixlen 64 scopeid 0x20<link>  
  ether e4:5f:01:40:90:09 txqueuelen 1000 (Ethernet)  
  RX packets 27339 bytes 13629449 (12.9 MiB)  
  RX errors 0 dropped 0 overruns 0 frame 0  
  TX packets 79 bytes 12553 (12.2 KiB)  
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

図8. ifconfigコマンドによるIPアドレスの確認

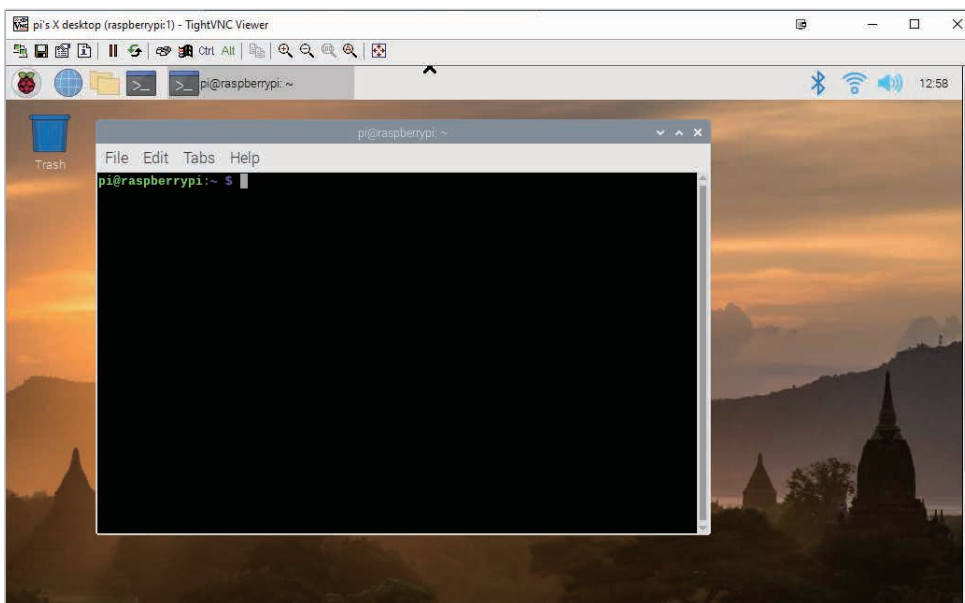


図9. リモートPC上のTightVNCビューアー

## お問い合わせ先：

オーストラリア 1 800 709 465  
オーストリア 00800 2255 4835  
バルカン諸国、イスラエル、南アフリカ、その他ISE諸国 +41 52 675 3777  
ベルギー 00800 2255 4835  
ブラジル +55 (11) 3530 8901  
カナダ 1 800 833 9200  
中央／東ヨーロッパ、バルト海諸国 +41 52 675 3777  
中央ヨーロッパ／ギリシャ +41 52 675 3777  
デンマーク +45 80 88 1401  
フィンランド +41 52 675 3777  
フランス 00800 2255 4835  
ドイツ 00800 2255 4835  
香港 400 820 5835  
インド 000 800 650 1835  
インドネシア 007 803 601 5249  
イタリア 00800 2255 4835  
日本 81 (3) 6714 3086  
ルクセンブルク +41 52 675 3777  
マレーシア 1 800 22 55835  
メキシコ、中央／南アメリカ、カリブ海諸国 52 (55) 88 69 35 25  
中東、アジア、北アフリカ +41 52 675 3777  
オランダ 00800 2255 4835  
ニュージーランド 0800 800 238  
ノルウェー 800 16098  
中国 400 820 5835  
フィリピン 1 800 1601 0077  
ポーランド +41 52 675 3777  
ポルトガル 80 08 12370  
韓国 +82 2 565 1455  
ロシア +7 (495) 6647564  
シンガポール 800 6011 473  
南アフリカ +41 52 675 3777  
スペイン 00800 2255 4835  
スウェーデン 00800 2255 4835  
スイス 00800 2255 4835  
台湾 886 (2) 2656 6688  
タイ 1 800 011 931  
イギリス、アイルランド 00800 2255 4835  
アメリカ 1 800 833 9200  
ベトナム 12060128

2022年2月現在



[www.tek.com/ja](http://www.tek.com/ja)

テクトロニクス／ケースレイインストルメンツ

各種お問い合わせ先：<https://www.tek.com/ja/contact-tek>

技術的な質問、製品の購入、価格・納期、営業への連絡、修理・校正依頼  
〒108-6106 東京都港区港南2-15-2 品川インターシティB棟6階

記載内容は予告なく変更することがありますので、あらかじめご了承ください。

Copyright © 2023, Tektronix. All rights reserved. TEKTRONIX およびTEK はTektronix, Inc. の登録商標です。  
記載された製品名はすべて各社の商標あるいは登録商標です。

2022年2月 48Z-73971-0