

レッスン⑥… UDPソケット通信の受信

国野 亘

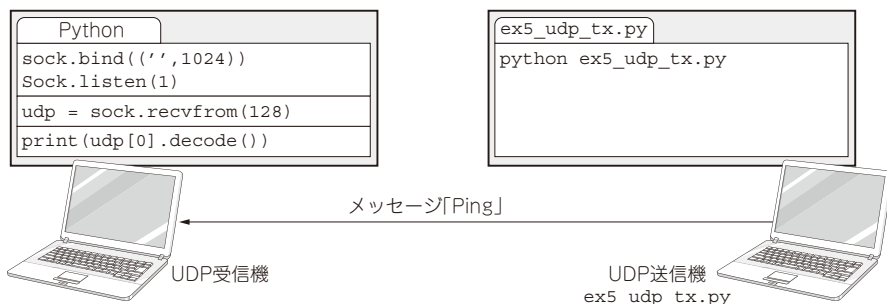


図1 UDPを受信する実験の構成
ポート番号1024でUDPデータ待ち受ける

本章では、Pythonの対話モードでUDPのメッセージを受信する方法について解説します(図1)。ローカル・ループバックで実験する、もしくはラズベリー・パイとPCの間で実際に通信を行うことが可能です。

ステップ①…対話モードで通信実験

第6章のレッスン⑤では、受信側にUDP受信ログャー `udp_logger.py` を使いました。ここでは、その動作をPythonの対話モードで確認します。前章で実行したUDP受信ログャーが起動している場合は、終了します。

● 手順1：Pythonの対話モードを起動する

ターミナル上でPythonの対話モードを起動します。次のコマンドを実行します。

```
> python
```

● 手順2：UDPのソケット通信用オブジェクトを生成する

UDP送信のときと同様に、ソケット通信用のライブラリ・モジュールである `socket` を組み込んで、`sock` オブジェクトを生成します。

```
>>> import socket
>>> sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

```
AF_INET, socket.SOCK_DGRAM)
```

● 手順3：待ち受け用ポートを設定する

`sock` オブジェクトをUDP待ち受け用ソケット通信インターフェースに接続します。引数はIPアドレスとポート番号です。全インターフェースのポート1024を設定する場合は、次のコマンドを実行します。

```
>>> sock.bind(('', 1024))
```

全インターフェースを指定することで、同じマシン上の他のプログラムだけでなく、ネットワーク上の他のマシンからの接続も受け入れます。

● 手順4：パケットを受信する

UDPの受信には、`recvfrom` メソッドを使います。引数は最大受信データ長(バイト)です。パケットの受信待機中は何も表示されません。

```
>>> udp = sock.recvfrom(128)
```

ここでUDP通信を実行します。新たにターミナルを起動し、次のコマンドを実行します。

```
> python ex5_udp_tx.py
```

パケットを受信すると、受信側のターミナルにプロンプト `>>>` が表示され、変数 `udp[0]` に受信データが、`udp[1]` に送信元のIPアドレスとポート番号が格納されます。次のコマンドで確認します。

```
>>> print(udp[0].decode(), udp[1])
```