0714

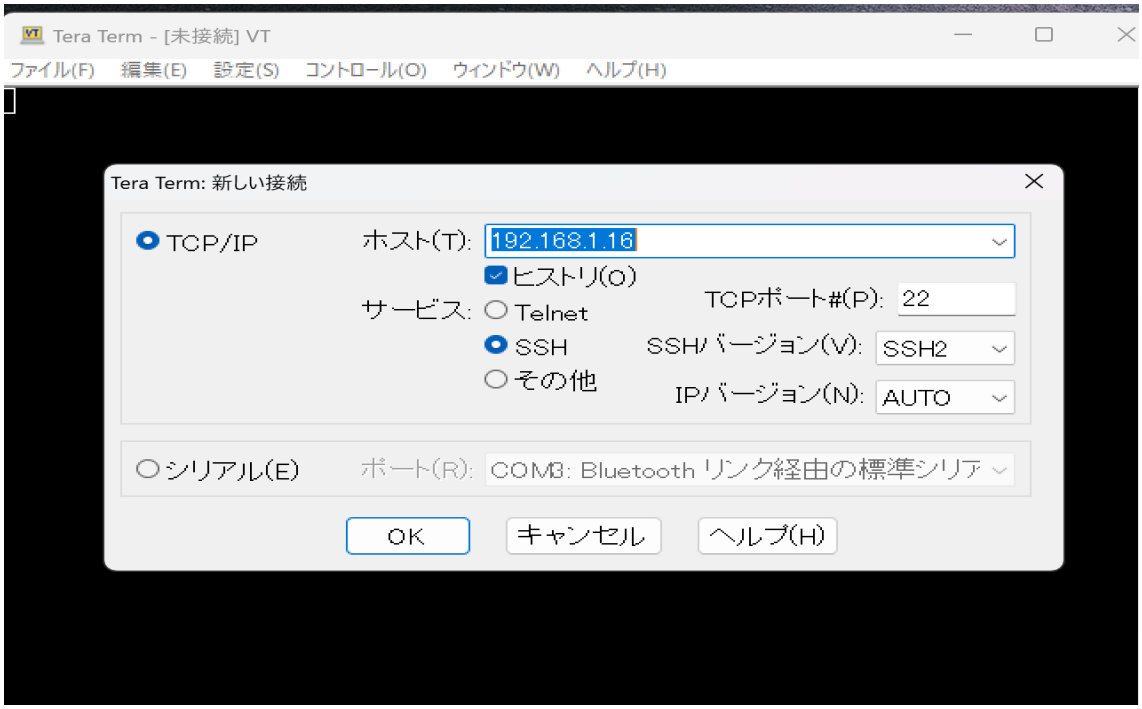
【TeraTerm】

TeraTerm をインストールした．細かい設定は使用する時に行う．

ネットワークを経由して他のコンピュータ（サーバ）に接続し，遠隔操作する SSH で接続する際に使用するソフトである．

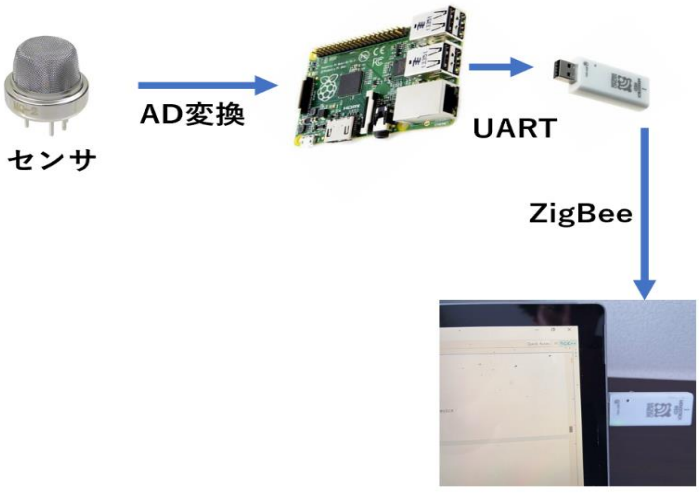
1 つのディスプレイに接続されている 2 つのラズパイの選択も可能．

今回は自分のパソコン(Windows)とラズパイの接続のために使用．



ラズパイの IP アドレスを指定しても「拒否された」と表示される．

【最終的な構造】



【UART 通信を用いた MONOSTICK 1 台での送受信(ループバック)】
始めとして，自身で送信（UART の Tx を使用）したデータを受信（UART の Rx を使用）するように構築する． 以下がコードである．

```
import serial
import struct

use_port = '/dev/ttyUSB0'
```

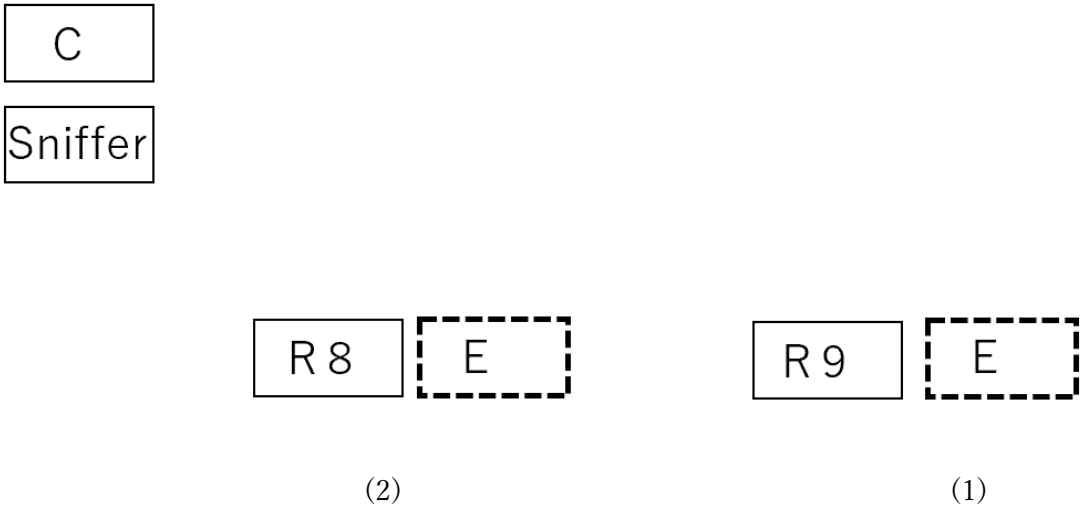
```
_serial = serial.Serial(use_port)
_serial.baudrate = 9600
_serial.parity = serial.PARITY_NONE
_serial.bytesize = serial.EIGHTBITS
_serial.stopbits = serial.STOPBITS_ONE
_serial.timeout = 5 #sec

commands = [ 0xB6, 0x01, 0x02, 0x00 ]

for cmd in commands:
    data = struct.pack("B", cmd)
    print("tx: ", data)
    _serial.write(data)

_serial.flush()

rx = _serial.readline()
print("rx: ", rx)

_serial.close()
```

serial：初期設定（シリアル通信の設定）
write：バイナリーデータ（bytes 型）を送信する
readline：データを受信する
open：ポート開く
close：ポート閉じる
flush：データを送信するまで待機する

実行結果として, 3 行目の use_port = '/dev/ttyUSB0' にエラーが表示される.

以下は Python でのシリアル通信 API の詳細 URL である.
pySerial API — pySerial 3.0 documentation (pythonhosted.org)

【狭い範囲でマルチホップ型になっているかの確認】
院生の研究室内で以下のように配置して行った.
E からデータを送信し, 宛先は C にしている.



(1) E を Router9 の近くに配置し, データを送信した.
以下が実行結果である. 送信元が R9 の MAC アドレスになっている.

```
Frame Counter: 2
Extended Source: IEEERegi_01:22:01:54:b2 (00:1b:c5:01:22:01:54:b2)
```

(2) E を Router8 の近くに配置し, データを送信した.
以下が実行結果である. 送信元が R8 の MAC アドレスになっている.

```
Destination: 0x0000
Source: 0x599b
Radius: 30
Sequence Number: 187
[Extended Source: IEEERegi_01:22:01:6c:13 (00:1b:c5:01:22:01:6c:13)]
```

考察として，
(2)の結果は配置の観点から正しいが，
(1)は R 9 →R 8 →C ならば，送信元のアドレスとしては R8 を示すはずだが，
Sniffer が同じ狭い空間にいるため，送信元として R9 の MAC アドレスを示す．

【スケジュール】

	スケジュール	実施したこと	できなかったこと	来週への課題
5/26 ~ 6/2	・JN5169にbeaconがないため JN5189を使用	・JN5189を検討結果使用しない ・pollコードを制御	・wiresharkの全般の理解	・JN5169を継続 ・E→Cの送信で検証 ・wiresharkでの確認
6/2 ~ 9	・E→Cでの送信を wiresharkで確認	・E→Cでの送信	・wiresharkでの正確な表示	・ArduinoかRaspberry Piを用いてAD変換を実施する
6/9 ~ 16	・ラズパイの初期設定 ・フィルタありのwireshark	・ラズパイの初期設定 ・wiresharkのフィルタで表示内容を制限	・適切なフィルター表示	・wiresharkで指定のパケットのみを表示 ・ラズパイでデータ収集 ・AD変換のプログラミング（Pythonの予定）
6/16 ~23	・wiresharkでのデータ確認 ・PythonでのAD変換とUART通信プログラミング	・wiresharkでの送信データの確認 ・実際のセンサを用いた過程でのプログラム構築 ・UART通信を実現するプログラム構築	・プログラムの動作確認 (AD変換に必要なラズパイのチップが手元にないため)	・センサとチップを使用しプログラムの動作確認
6/23~30	・AD変換に必要なチップ（ADS1015)を実装 ・E→R→Cの経路をsnifferで確認	特にR→Cの経路をsnifferで確認	AD変換チップが手元にない ため、未確認	AD変換チップをラズパイに実装する．
6/30~7/7	・AD変換チップをラズパイに実装 ・UARTの初期設定	・UARTに関する情報収集 ・論文調査	AD変換チップの実装	AD変換チップをラズパイに実装する．
7/7〜14	・ラズパイとMONOSTICK間でのUART通信の構築 ・狭い範囲でのSnifferを動作	・UART通信の大まかなプログラミング ・SnifferをWiresharkで確認	・UART通信のloopback	・loopbackを可能にする ・AD変換チップの実装
7/14~ 21	・AD変換チップの実装			