

# セキュリティキャンプ LT 実験機器を自動化しよう

松岡 航太郎

Aug. 2024



- 松岡 航太郎
- S10 「暗号のまま計算しよう」ゼミ講師・開発コースアドバイザ
- 京都大学大学院情報学研究科通信情報システムコース 佐藤研究室 博士課程 2 年
  - 集積回路の研究室
  - 今日喋るのは電気系の実験機器の自動化
- NHK 学生ロボコン 2019 優勝



# 今回自動化した実験機器

## ① WaveStation 2052 (手前/黒)

- クロックを生成する Function Generator

## ② Keysight 34465A/34410A(右/白2段)

- 電圧測定用 Digital MultiMeter (DMM)

## ③ DMM7510

- 電流測定用 DMM

## ④ Keysight E36233(奥下段)

- 電源

## ⑤ Tektronix DPO4104B (画面外左)

- オシロスコープ

## ⑥ Raspberry Pi 5(中央)

- 全体の制御と SPI



図: 実験系全体図



- General Purpose Interface Bus の略
  - HP 社の HP-IB が IEEE488 として規格化
  - ピギーバック方式の 24 ピンコネクタ
- 一昔前の実験機器はみんなこれを搭載
  - きっと使うのが簡単なはず
  - そう思っていた時期が私にも.....



図: GPIB ケーブル



# カーネルモジュールビルドしてね!(なお RasPi5ではできないとする)

- Raspberry Pi 5 を制御に利用したいので必然的に Linux で動かせる必要がある
- GPIB-USB 変換モジュールのドライバはメインラインに入ってない
  - <https://linux-gpib.sourceforge.io/>
  - GPL-v3 でライセンスされているのでメインラインに入ってない
  - 多分メーカ独自の何かが入ってる
- ドライバを入れるにはカーネルのヘッダが必要
  - Raspberry Pi OS の場合通常 *rpi-source* を用いてソースをダウンロードする
  - Raspberry Pi 4B 以降まだサポートされていないらしい.....
  - <https://forums.raspberrypi.com/viewtopic.php?t=371039>
- 頑張れば多分動くがあきらめた



図: GPIB-USB 変換



- GPIB が使われていたのはパラレルバスの利用が当たり前だった頃
- 今日の周辺機器接続規格としては USB や Ethernet が一般的
- USBTMC: USB Test and Measurement Class の略
  - USBHID なんかと同じように USB の標準的なクラスの一つ
- LXI: LAN eXtention for Instruments の略
  - Ethernet で実験機器を接続するためのプロトコル
  - Class A, B, C の 3 種類に分類される
    - ほとんどの機器は Class C
    - Class C: ほぼ GPIB の置き換え
    - Class B: Class C + IEEE-1588 (Precision Time Protocol)
    - Class A: Class B + トリガーバス (同期用の信号線)
- これらはあくまで物理層とその上の最低限のプロトコルの規格
  - この上でどういうデータを流すかはまた別の規格



- Virtual Instrument Software Architecture の略
- 物理層を抽象化するための API
  - 右の図は VISA の立ち位置を示す NI 社の図
  - 実装が複数あるだけで規格は IVI Foundation が管理
  - Python 実装の PyVISA が有名 (私もこれ)
- あくまで物理層の差異を隠すだけなので流すデータは規定しない

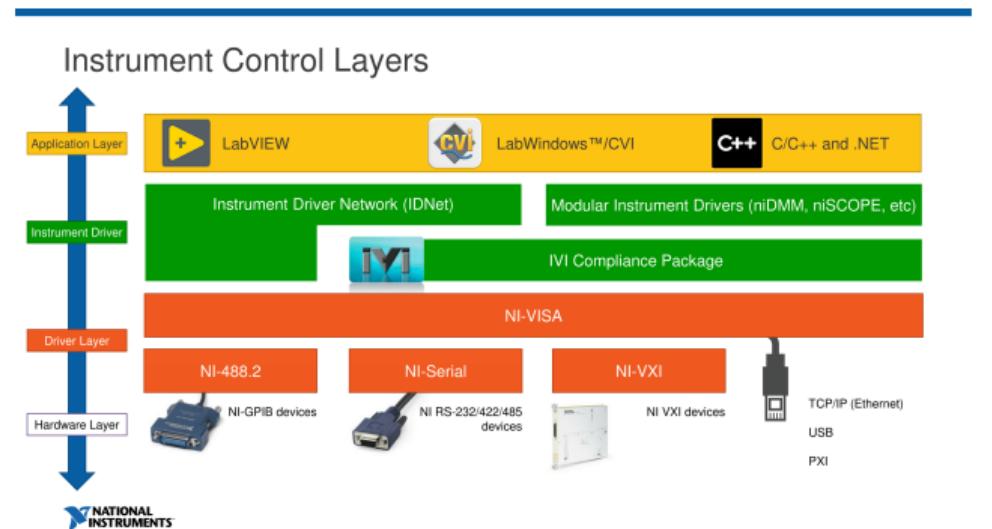


図: [https://knowledge.ni.com/  
KnowledgeArticleDetails?id=kA00Z0000019XkSAM](https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000019XkSAM)



- Standard Commands for Programmable Instruments の略
- いくつかあるデータの規格のうちの最も一般的
  - GPIB の拡張である IEEE 488.2 が源流
- ASCII の文字列でコマンドを送ること、基本的なコマンドの文法、最低限のコマンドくらいしか規格化していない
  - 機種の間でコマンドに互換性がないことはよくある
    - リファレンス上は動きそうなコマンドが動かないこともよくある
    - 何なら各社コマンド例の誤植が.....
  - 最低限のコマンドには\*RST(リセット), \*IDN?(機器の名称クエリ)などがある



# 実際こんな感じのコード

```
1 import pyvisa
2 rm = pyvisa.ResourceManager()
3 # 電源と接続
4 powersupply=rm.open_resource('TCPIP::10.0.20.72::INSTR')
5 # 電圧を SCPI で設定
6 powersupply.write('APPL CH1, 0.9,0.2')
7 # Function Generator と接続
8 functiongenerator =
    → rm.open_resource('USB0::1535::2593::LCRY2362C01679::0::INSTR')
9 # 矩形波の 1MHz に設定
10 functiongenerator.write('C1:BSWV
    → WVTP,SQUARE,AMP,0.9,OFST,0.9,FRQ,1000000')
11 # 出力を ON
12 powersupply.write('OUTP ON,(@1)')
13 functiongenerator.write('C1:OUTP ON')
14 # 電流量を表示
15 print(powersupply.query('MEAS:CURR? (@1)'))
```



- 実験機器の現在主流な物理層は3つ
  - GPIB, USBTMC,LXI
- 物理層がどれであっても VISA が提供されていれば同じ API でたたける
  - PyVISA を使えば簡単に接続できる
  - GPIB はそれ用のドライバが必要だが他は標準的
- SCPI に対応していれば説明書とにらめっこするだけで操作可能
  - 移植性のない機種依存なので定義上動きそうなコマンド列が動かなかったりはある
  - 誤植は勘で補おう
- 物理的な実験をすることになっても自動化して家からできるようにしよう！
  - 実験の再現性の意味でも有意義

