

計測器の遠隔制御について

1. はじめに

かつてはオシロスコープやデジタルマルチメーターをはじめとした計測器を外部から制御するには、GPIB(General Purpose Interface Bus)を介して行うのが主流でした。近年では多くの計測器にLANやUSB等のハードウェアインターフェース(以下、ハードウェア I/F)が搭載され、外部からの制御を実現する手段の選択肢が増えています。

2. 計測器との通信

PC等のコンピュータから計測器への通信にはハードウェア I/F 制御用のライブラリを用います。かつてはボードメーカーごとにライブラリが異なり、メーカーが変わるとプログラムで使用する関数等もライブラリに合わせて変更しなければならないという難点がありました。最近では、計測器業界の標準仕様である VISA (Virtual Instrument Software Architecture) に基づいたライブラリが計測器メーカーから提供されています。VISA は計測器のメーカーや種類の違いだけでなく、ハードウェア I/F の違いも気にすることなく、制御プログラムの容易な開発や変更を可能にします。

3. 制御用プログラミング

制御を行うプログラムは Visual Basic や Excel VBA、MATLAB 等により作成されることが多いですが、人工知能やデータ解析で利用される Python でも作成することが可能です。VISA を扱えるライブラリである pyVISA を Python へ取り込み、write メソッドや query メソッドを用いて IEEE 488.2 共通コマンドや SCPI コマンドを実行させることで計測器の制御やデータの取得ができます。Python は他のプログラミング言語と比べて少ないコード量で記述できる特徴があり、手軽に外部からの遠隔制御・計測プログラムを作成できます。

デジタルマルチメーターの電流値を表示する Python によるプログラムの例を図 1 に示します。データを一回読みだすだけの動作内容ですが、10 行程度で実現できます。

また、Python は Numpy をはじめとした科学計算や統計処理を効率的に行えるライブラリや Matplotlib のようにデータを視覚化するためのライブラリが豊富に用意されているため、制御及びデータの取得にとどまらず、取得したデータの分析や表示も一連の処理で実行できます。

```
import pyvisa
# 制御対象計測器の指定 (LANを介した制御の例)
rm = pyvisa.ResourceManager()
inst=rm.open_resource('TCPIP0::192.168.2.2::INSTR')
# 計測器の初期化
inst.write("*RST")
inst.write("*CLS")
# 直流電流の測定の設定 (メーカー・機種により引数は変わる)
inst.write("CONF:CURR:DC 1")
# 測定及びデータの問い合わせ (メーカー・機種により引数は変わる)
curData=inst.query("READ?")
# 測定データの表示
print(float(curData))
```

図 1 Python による制御プログラムの例

4. ハードウェア I/F の選択

複数のハードウェア I/F を利用できる場合、実現したいことや各ハードウェア I/F の長所・短所を総合的に勘案して選択することになります。中でも LAN は、かなり離れた場所でも制御できる、専用の USB ドライバが不要である、大量のデータ転送が短時間でできる点がメリットになります。また、LAN 上にプログラム開発ツールである Node-RED が動作するコンピュータがあれば、収集した計測値データやグラフを表示させることができ、計測器から離れた場所でも LAN を介してデータをリアルタイムで確認することもできます(図 2)。

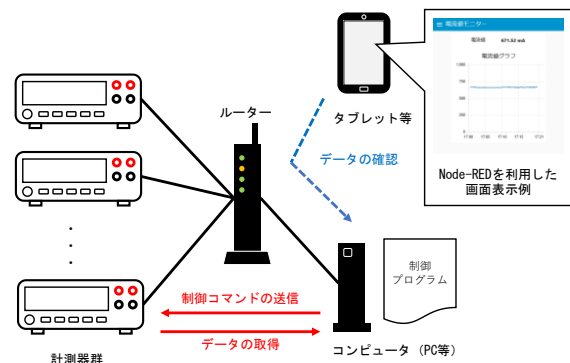


図 2 LAN を介した遠隔制御のイメージ

5. おわりに

計測器の遠隔制御についてご紹介しました。Python による LAN を介した制御は容易に制御を実現できる手段の一つであると思われます。



共同研究支援部 試作評価室 浅井 徹(0561-76-8316)

研究テーマ: EMC

担当分野: EMC、情報技術