

การเขียนโปรแกรม Python เชื่อมต่ออุปกรณ์ Rigol DG1022 Digital Function Generator ผ่านพอร์ต USB

เขียนโดย เรวัต ศิริโกศาภิรมย์

ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ ม.เทคโนโลยีพระจอมเกล้าพระนครเหนือ (KMUTNB)

เผยแพร่ครั้งแรก: วันที่ 25 พฤศจิกายน พ.ศ. 2560 (November 25, 2017)

ปรับปรุงแก้ไข: วันที่ 26 กันยายน พ.ศ. 2563 (September 26, 2020)

คำสำคัญ / Keywords: Digital Function Generator, Python for Automated Instrumentation, Rigol DG1022

เครื่องมือสร้างคลื่นสัญญาณทางไฟฟ้าและการเขียนโปรแกรมควบคุม

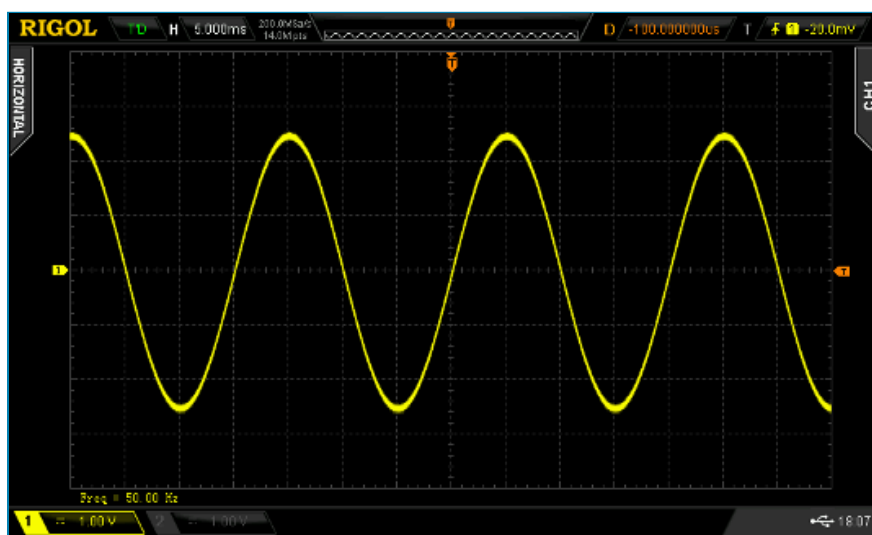
เครื่องมือวัดอย่างเช่น ออสซิลโลสโคป (Oscilloscope) และเครื่องกำเนิดสัญญาณ (Waveform Generator / Function Generator) เป็นสิ่งสำคัญในการทำงานด้านระบบสมองกลฝังตัว อิเล็กทรอนิกส์ และงานไฟฟ้าอื่น ๆ

เมื่อพิจารณาความสามารถและคุณลักษณะเฉพาะของเครื่องมือวัดออสซิลโลสโคป และเครื่องกำเนิดสัญญาณที่ทำงานแบบดิจิทัล จะเห็นได้ว่าหลาย ๆ รุ่น มีพอร์ตสำหรับเชื่อมต่อกับคอมพิวเตอร์ เช่น RS232, USB และ Ethernet (RJ45) และสามารถใช้ซอฟต์แวร์จากคอมพิวเตอร์ควบคุมการทำงานของอุปกรณ์ดังกล่าวได้ นอกเหนือจากการควบคุมผ่านแผงควบคุม หรือปุ่มกดต่าง ๆ ของตัวเครื่องมือวัด

เครื่องกำเนิดสัญญาณ สามารถสร้างสัญญาณไฟฟ้าแบบมีคาบ (Periodic Signals) ตามรูปแบบที่ต้องการได้ เช่น คลื่นไซน์ (Sinusoidal) คลื่นสี่เหลี่ยม (Rectangular) คลื่นสามเหลี่ยม (Triangular) สัญญาณแบบ PWM (Pulse Width Modulation) หรือ สร้างคลื่นสัญญาณแบบมีคาบ จากชุดข้อมูลที่อัปโหลดไปยังหน่วยความจำภายในของเครื่องได้



รูปเครื่องกำเนิดสัญญาณ Rigol DG1022



ตัวอย่างรูปคลื่นสัญญาณ Sine (50Hz) จากเครื่องกำเนิดสัญญาณที่วัดได้ด้วยออสซิลโลสโคป

บทความนี้กล่าวถึง สาธิตการทดลองใช้ภาษาคอมพิวเตอร์ Python ในการเชื่อมต่อกับ Rigol Digital Function Generator DG1000 Series และได้เลือกทดลองกับเครื่องโมเดล DG1022 โดยเชื่อมต่อกับคอมพิวเตอร์ผ่านพอร์ต USB 2.0

การเขียนโปรแกรมสื่อสารกับอุปกรณ์นี้ จะอาศัยรูปแบบการสื่อสารที่เรียกว่า VISA (Virtual Instrument Software Architecture) หรือในเชิงซอฟต์แวร์ ก็มองว่าเป็น Programming API สำหรับการเชื่อมต่อกับอุปกรณ์เครื่องมือวัด ซึ่งเป็นที่ยอมรับและใช้งานแพร่หลาย และสำหรับการใช้งานร่วมกับภาษา Python ก็มีไลบรารีที่ชื่อว่า PyVISA <https://pyvisa.readthedocs.io/> รองรับการใช้ VISA

แต่ในกรณีที่ใช้ Windows จะต้องติดตั้งซอฟต์แวร์อย่างเช่น NI VISA <https://www.ni.com/visa/> ของบริษัท NI (National Instruments) ก่อนใช้งาน แนะนำให้ติดตั้ง NI-VISA Runtime 17.0 หรือเวอร์ชันใหม่กว่า แต่ถ้าใช้ Linux Ubuntu สามารถใช้ pyvisa-py (Pure Python Implementation of VISA) <https://github.com/pyvisa/pyvisa-py> แทนได้

การใช้ภาษา Python นั้นสามารถใช้ได้ทั้งระบบปฏิบัติการ Windows, Linux และ MAC OS X แต่ในบทความนี้จะกล่าวถึงเฉพาะการใช้งานสำหรับระบบปฏิบัติการ Windows 10 และได้ติดตั้ง Python 3.x ไว้แล้ว การทดลองใช้คำสั่งเพื่อรัน Python Script จะกระทำผ่าน Git Bash for Windows <https://git-scm.com/download/win>

วัตถุประสงค์การเรียนรู้

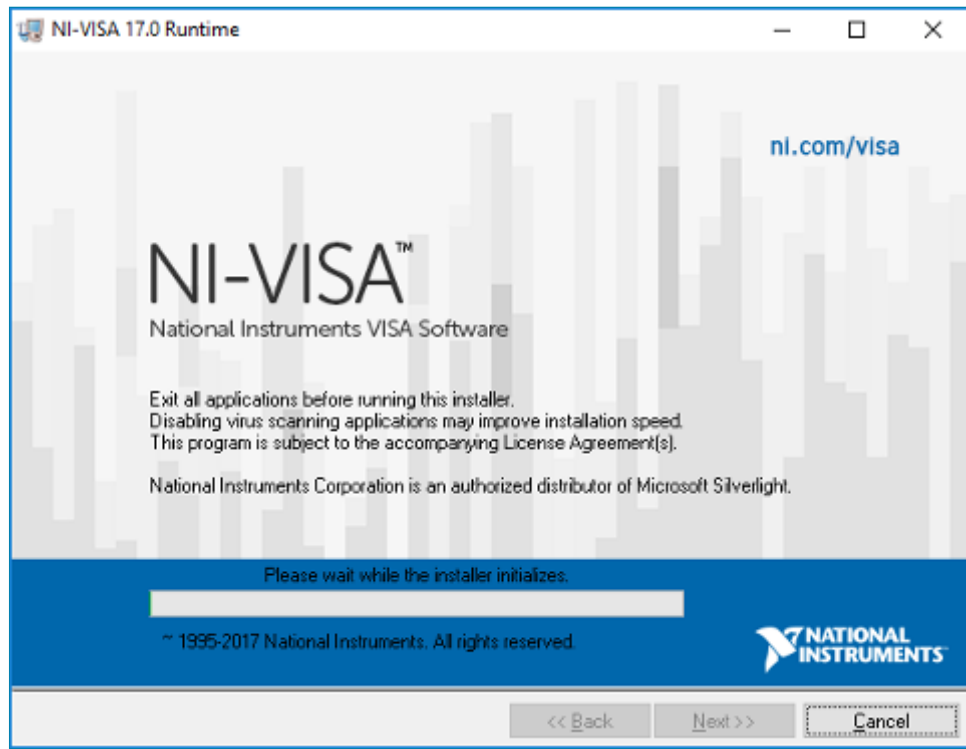
- ศึกษาขั้นตอนการติดตั้งและใช้งานซอฟต์แวร์สำหรับเชื่อมต่อกับเครื่องมือกำเนิดสัญญาณโดยใช้ภาษา Python
- ทดลองใช้ Python Script ที่ให้ไว้เป็นตัวอย่างในการเชื่อมต่อและควบคุมสั่งการอุปกรณ์ผ่านพอร์ต USB ด้วยภาษา Python
- เรียนรู้คำสั่งต่าง ๆ เบื้องต้น ที่ใช้โปรแกรมอุปกรณ์ Rigol Digital Function Generator Series DG1000 เพื่อสร้างรูปคลื่นสัญญาณทางไฟฟ้า

ขั้นตอนการติดตั้งแพ็คเกจสำหรับ Python สำหรับ Windows 10

```
pip3 install pyusb -U
pip3 install pyvisa -U
pip3 install pyvisa-py -U
```

ขั้นตอนการติดตั้งแพ็คเกจสำหรับ Python สำหรับ Linux Ubuntu

```
sudo -H pip3 install pyusb -U
sudo -H pip3 install pyvisa -U
sudo -H pip3 install pyvisa-py -U
```



การติดตั้ง NI-VISA Runtime สำหรับ Windows

```
Select Git CMD

C:\Users\RSP>pip3 install pyusb -U
Collecting pyusb
Installing collected packages: pyusb
Successfully installed pyusb-1.0.2

C:\Users\RSP>pip3 install pyvisa -U
Collecting pyvisa
Installing collected packages: pyvisa
Successfully installed pyvisa-1.8
```

การติดตั้ง Python Packages เช่น pyusb, pvisa โดยใช้คำสั่ง PIP สำหรับ Python 3

รายการอุปกรณ์สำหรับการทดลอง

- เครื่องกำเนิดคลื่นสัญญาณ Rigol Digital Function Generator DG1022 พร้อมสายสัญญาณ 1 เส้น สำหรับสร้างสัญญาณทดสอบ
- เครื่องมือวัด Rigol Digital Oscilloscope รุ่น DS1052E พร้อมสาย Probe อย่างน้อย 1 เส้น สำหรับวัดคลื่นสัญญาณ
- สาย USB type-B สำหรับเชื่อมต่อระหว่างคอมพิวเตอร์และเครื่อง Rigol DG1022
- คอมพิวเตอร์สำหรับรันสคริปต์ Python และติดตั้งซอฟต์แวร์ที่จำเป็นในการทำงานไว้แล้ว

ตัวอย่าง Python Script สำหรับทดสอบการเชื่อมต่อกับอุปกรณ์เครื่องมือวัด

```
#!/usr/bin/env python
#####
# Author: Rawat S.
#   (Dept. of Electrical & Computer Engineering, KMUTNB, Bangkok/Thailand)
#####

import visa
import time, sys

visa_driver = '' # use either 'visa64' or 'visa32' or '@py' or left empty.
resources   = visa.ResourceManager( visa_driver )
devices     = resources.list_resources()
usb_device  = None

if len(devices) > 0:
    print ('Found #devices: %d' % len(devices) )
    for device in devices:
        print ('>>', device)
        if str(device).startswith('USB0'):
            usb_device = device
            print ('select:', str(device)) # select the first device found
            break

print (30*'-')

if usb_device != None:
    instr = resources.open_resource( usb_device ) # use the selected USB device
    instr.write( '*IDN?' )
    time.sleep(0.5)
    ret_str = instr.read()
    fields = ret_str.split(',')
    dev_model = fields[2]
    print ( ', '.join(fields[0:3]) )
    #print (ret_str)
    print ( dev_model )
else:
    print ('No device found')
    sys.exit(-1)

print ('\nDone....')
```

โค้ดตัวอย่างแรกนี้ ใช้ทดสอบการทำงานของ Python Script เพื่อเชื่อมต่อกับ PyVISA ผ่านพอร์ต USB ไปยังเครื่องกำเนิดสัญญาณ DG1022 ในกรณีที่มียุอุปกรณ์เชื่อมต่อกับพอร์ตของ USB อยู่ในขณะนั้น จะมีการส่งคำสั่ง *IDN? เพื่ออ่านข้อมูลจากเครื่องที่ใช้ระบุว่าเป็นเครื่องรุ่นใด มีรหัส Serial Number เป็นอย่างไร

ตัวอย่างที่ 2: Python Script สำหรับการเลือกอุปกรณ์สัญญาณแบบ Sine Waveform

```
#!/usr/bin/env python3
#####
# Author: Rawat S.
#   (Dept. of Electrical & Computer Engineering, KMUTNB, Bangkok/Thailand)
#####
import visa
import time, sys, re

vendor_id = None
device_id = None
instr_model = None
resources = None
instr = None

#####
visa_driver = '' # use either 'visa64' or 'visa32' or '@py' or left empty.
resources = visa.ResourceManager( visa_driver )
devices = resources.list_resources()

if len(devices) > 0:
    print ('Found #devices: %d' % len(devices) )
    for device in devices:
        print ('>>', device)
        device = device.replace(':', ',')
        fields = device.split(',')
        if len(fields) == 5 and fields[3].startswith('DG'):
            vendor_id = fields[1]
            device_id = fields[2]
            instr_model = fields[3]
            print (vendor_id, device_id, instr_model)
print (30*'-')

#####

def listInstruments():
    global resources
    devices = resources.list_resources()
    print (devices)

def selectInstrument( vendor_id, device_id, instr_model ):
    cmd_str = "USB0::%s::%s::%s::INSTR" % (vendor_id,device_id,instr_model)
    instr = resources.open_resource( cmd_str )
    return instr
```

```

def cmdWrite(cmd, dly=0.1):
    global instr
    instr.write( cmd )
    time.sleep( dly )

def cmdRead(cmd, dly=0.1):
    global instr
    instr.write( cmd )
    time.sleep( dly )
    try:
        str = instr.read()
    except Exception:
        str = None
    return str

def showInstrumentInfo():
    print (cmdRead("*IDN?"))

#####

# select Rigol DG1022
if vendor_id == '0x1AB1' and device_id == '0x0588':
    instr = selectInstrument( vendor_id, device_id, instr_model)
else:
    print ('No DG1022 instrument found !!!')
    sys.exit(-1)

showInstrumentInfo()
# sample output: RIGOL TECHNOLOGIES,DG1022 ,DG1D131101556,,00.02.00.06.00.02.07

# connect to the remote instrument
cmdWrite( "SYSTem:REMOte" )

# reset the instrument
cmdWrite( "*RST", 1.0 )

#####

freq  = 100
volt  = 1
offset = 0
phase = 0

#####

print ('turn off output CH1')
cmdWrite( "OUTP1 OFF" )

print ("select Sine waveform")
cmdWrite( "FUNC SIN" )

```

```

print ("set initial frequency: {:d} Hz".format(freq))
cmdWrite( "FREQ %d" % freq )

print ("select voltage output unit: voltage peak-to-peak")
cmdWrite( "VOLT:UNIT VPP" )

print ("set initial voltage value: {:.3f}".format( volt ))
cmdWrite( "VOLT %.1f" % volt )

print ("set initial voltage offset: {:d}".format( offset ))
cmdWrite("VOLT:OFFS %.3f" % offset )

print ("set initial phase: {:d} [Deg.}".format( phase ))
cmdWrite( "PHAS %d" % phase )

print ("turn on the output CH1")
cmdWrite("OUTP1 ON")

try:
    volt = 2.0
    for freq in [100,200,500,1000,2000,5000,10000]:
        cmdWrite("APPL:SIN {:d},{:.3f},{:.3f}".format(freq, volt, offset) )
        print ('Freq:      {:6.1f} Hz'.format( float(cmdRead( "FREQ?" )) ))
        print ('VOLT:LOW:  {:+2.1f} V'.format( float( cmdRead( "VOLT:LOW?" )) ))
        print ('VOLT:HIG:  {:+2.1f} V'.format( float( cmdRead( "VOLT:HIG?" )) ))
        print (40*'-')
        time.sleep(2.0)

except KeyboardInterrupt:
    print ('Terminated....')

print ('Done....')
time.sleep(1.0)

```

ตัวอย่างที่สองนี้ แสดงการเลือกรูปคลื่นสัญญาณแบบ Sine Wave และมีการกำหนดความถี่ (Frequency) เช่น เริ่มต้นที่ 100 Hz และกำหนดแอมพลิจูด (Amplitude) หรือ ช่วงแรงดัน Vpp (Voltage Peak-to-Peak) เช่น เท่ากับ +/- 1V เป็นต้น ตามที่ต้องการ และยังทดลองส่งคำสั่งเพื่อเปลี่ยนค่าความถี่ของสัญญาณดังกล่าว ในช่วง 100,200,...,5000,10000 Hz ตามลำดับ โดยเว้นช่วงเวลาประมาณ 2 วินาที ในการเปลี่ยนความถี่ ถ้าใช้เครื่องออสซิลโลสโคป ก็สามารถตรวจสอบได้ว่าสัญญาณเอาต์พุตจากเครื่อง DG1022 นั้นเป็นไปตามที่โปรแกรมด้วย Python Script หรือไม่ เช่น ตรวจสอบดูค่าความถี่ ค่าแอมพลิจูดหรือระดับแรงดันของสัญญาณ

Note: คำสั่งที่เกี่ยวข้องกับ Rigol DG1022 สามารถศึกษาเพิ่มเติมได้จากเอกสารที่เป็นไฟล์ .PDF ของผู้ผลิต [DG1000 Programming Guide \(2014\)](#)


```
Select Git CMD

C:\Users\RSP>python test_pyvisa_dg1022.py
C:\Python27\lib\site-packages\pyvisa\ctwrapper\functions.py:1225: VisaIOWarning: VI_WARN_0
e specified configuration either does not exist or could not be loaded. VISA-specified def
ret = library.viOpenDefaultRM(byref(session))
(u'USB0::0x1AB1::0x0588::DG1D131101556::INSTR',)
RIGOL TECHNOLOGIES, DG1022 , DG1D131101556, , 00.02.00.06.00.02.07

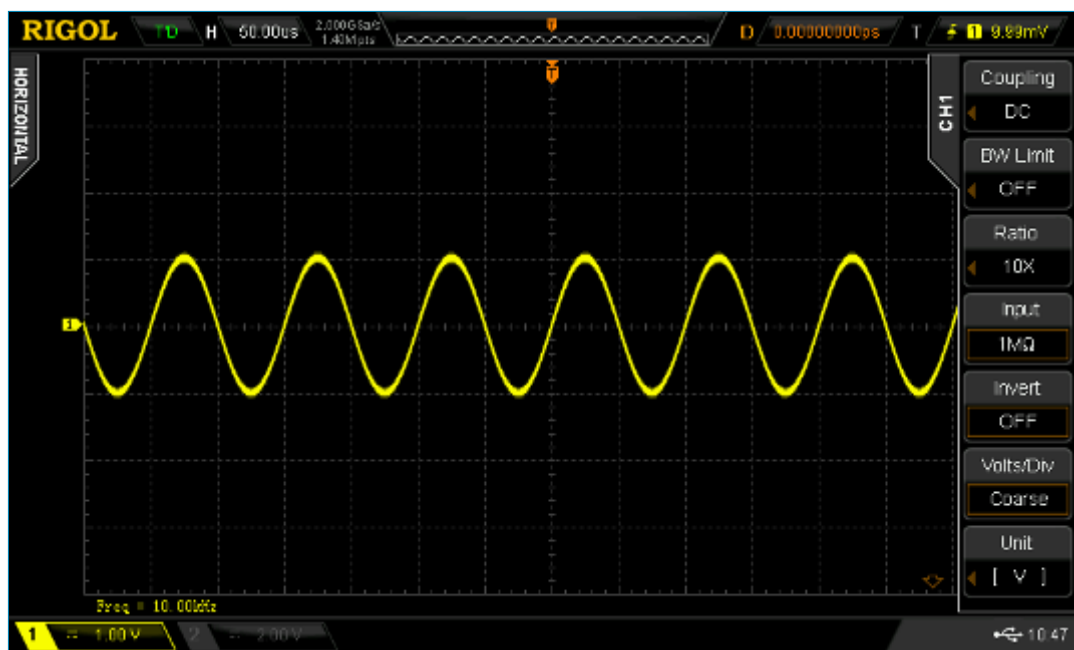
turn off output CH1 1
select Sine waveform
set initial frequency: 100 Hz
select voltage output unit: voltage peak-to-peak
set initial voltage value: 1.000
set initial voltage offset: 0
set initial phase: 0 [Deg.]
turn on the output CH1
Freq:      100.0 Hz
VOLT:LOW:  -1.0 V
VOLT:HIG:  +1.0 V

Freq:      200.0 Hz
VOLT:LOW:  -1.0 V
VOLT:HIG:  +1.0 V

Freq:      500.0 Hz
VOLT:LOW:  -1.0 V
VOLT:HIG:  +1.0 V

Freq:      1000.0 Hz
VOLT:LOW:  -1.0 V
VOLT:HIG:  +1.0 V
```

ตัวอย่างเอาต์พุตที่เป็นข้อความเมื่อรันสคริปต์ Python ตามตัวอย่างที่สอง



รูปคลื่นสัญญาณแบบ Sine Wave (10kHz, 2Vpp) จากเครื่องกำเนิดสัญญาณที่วัดได้ด้วยออสซิลโลสโคป

สรุปผลการเรียนรู้ที่คาดหวัง

- ได้เรียนรู้การติดตั้งซอฟต์แวร์ที่จำเป็นสำหรับการเชื่อมต่อกับอุปกรณ์เครื่องมือวัดที่รองรับ VISA Programming API

- สามารถใช้ภาษา Python ในการติดต่อสื่อสารและเชื่อมต่อกับเครื่องกำเนิดสัญญาณ Rigol DG1022 ผ่านพอร์ต USB
- ได้ลองสร้างคลื่นสัญญาณทางไฟฟ้าจำนวน 1 ช่องสัญญาณ แบบ Sinusoidal Waveform ตามคุณสมบัติที่ต้องการ จากเครื่องกำเนิดสัญญาณ โดยใช้ Python Script

แนวทางการเรียนรู้เพิ่มเติม

- ทดลองใช้ฟังก์ชันหรือรูปสัญญาณแบบอื่น เช่น คลื่นรูปสามเหลี่ยม คลื่นรูปสี่เหลี่ยม เป็นต้น
- ทดลองสร้างสัญญาณเอาต์พุต จำนวน 2 ช่องสัญญาณ เช่น คลื่นรูปไซน์ที่มีความถี่และแอมพลิจูดเหมือนกัน แต่มีเฟสต่างกัน เป็นต้น
- ทดลองเขียนโปรแกรม Python เพื่อควบคุมการทำงานทั้งออสซิลโลสโคปและเครื่องกำเนิดสัญญาณพร้อมกัน

เผยแพร่ภายใต้ลิขสิทธิ์ / This work is licensed under: Attribution-NonCommercial 4.0 International (CC BY-NC-SA 4.0)