

プログラム言語 Python について



Python は商用利用可能なオープンソースの開発言語であり、OS・環境への依存が低い言語となっています。このため C#などのように開発環境の商用利用に費用が発生しません。プログラミングのための参考資料やサンプルが豊富にあります。人口知能や Web アプリケーション、統計処理についてが多く、通信を使って計測機器などを制御する例はそれほど多くありません。本資料では LAN(Socket)と RS-232C(USB-CDC を含む)で通信を行う場合の最低限の情報を提供いたします。GP-IB や VISA ライブラリでの利用については、pyVISA のライブラリを使用しています。

開発環境について

環境については Windows 上で Python3.6 を標準としています。またシリアル通信を使うために pySerial、VISA 通信では pyVISA が必要になりますので先にインストールしておいてください。環境によっては 32bit と 64bit のバイナリの混在する場合がありますので注意が必要な場合があります。本資料では環境構築については扱いません。必要に応じて Python の実行環境(Visual Studio code、pyscript など)をご用意ください。

プログラミングの注意について

- ・文字列の扱いについて
アプリケーション内では文字列はユニコードで扱われています。通常計測機器の制御は ASCII 文字列を使用しています。サンプルでは送信時は.encode('utf-8')、受信時は.decode('ascii')を使用して変換を行っています。VISA の場合は文字列変換は使用していません。
- ・シリアルポートのポート名の記述について
Windows の環境ではシリアルポートは COM1 から始まるポート番号が割り当てられます。USB 機器にシリアル番号が付与されている場合はそのポート番号は確保され次に接続した場合にはそのポート番号が使われるようになっていきます。Win10 ではデバイスマネージャの非表示のデバイスを表示させて未接続のデバイスを削除することでポート番号の再利用が可能になります。
Linux 系の環境ではシリアルポートは ttyACM または ttyUSB にポート番号 0 から割り当てられます。ポート番号の確認は dmesg コマンドを利用します。ポートの割り当て直後はパーミッションが不足していますので、RW 可能に変更してください。また、ポート番号はシステムの再起動などで変化する場合もあり、通信確立後に '*IDN?' コマンドで機器を確認して使うようにしてください。

dmesg コマンドの使用例

```

root@kona:~# dmesg
[ 4362.576991] usb 1-2.1: new high-speed USB device number 5 using xhci_hcd
[ 4362.988457] usb 1-2.1: New USB device found, idVendor=098f, idProduct=2205
[ 4362.988487] usb 1-2.1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 4362.988530] usb 1-2.1: Product: DCS-1104B
[ 4362.988556] usb 1-2.1: Manufacturer: TEXIO
[ 4362.988582] usb 1-2.1: SerialNumber: GJP180497
[ 4363.376707] cdc_acm 1-2.1:2.0: This device cannot do calls on its own. It is not a modem.
[ 4363.378076] cdc_acm 1-2.1:2.0: ttyACM0: USB ACM device
[ 4363.381822] usbcore: registered new interface driver cdc_acm
[ 4363.381859] cdc_acm: USB Abstract Control Model driver for USB modems and ISDN adapters
    
```

機器を認識した順番に表示されるので最後に接続すると見やすいです

利用するポート名

- ・通信のウエイトについて
応答を持つコマンドを送信する場合は、送信後に機器に届くまでの時間が必要な場合があります。その時間はばらつきがあり、長期安定を求める場合は応答を受信するまでにウエイトを入れることをお勧めしています。
- ・サンプルプログラムについて
本資料に掲載されているプログラムはサンプルです。環境によっては動作しないことがありますのでご注意ください。また確認用の Python のバージョンは 3.6 を採用しています。

LAN / RS-232C / USB-CDC サンプルプログラム例

```

# coding:utf-8
import serial          # pyserial モジュール
import socket         # socket モジュール
from time import sleep

# LAN 用定義
def main_LAN():
    ipAddr = "192.168.1.100"    # IPv4 アドレス指定
    ports = 1026                # Socket ポート指定

    try:
        s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.connect((ipAddr, ports))

        s.send("*idn?\n".encode("utf-8"))
        sleep(0.1)
        txtRcv = s.recv(256)
        print(' IDN: ' + txtRcv.decode('ascii'))

        s.close()

    except:
        print(' Socket Error: ' + ipAddr + '::' + str(ports))

# RS-232C/USB-CDC 用定義
def main_COM():
    ports='COM10'              # Windows での指定例 : COM10
    #ports='/dev/ttys0'         # Linux 系での RS-232C 指定 : ttys0
    #ports='/dev/ttyUSB0'      # Linux 系での USB-CDC 指定 : ttyUSB0

    try:
        ser = serial.Serial()
        ser.port = ports        # ポート指定
        ser.baudrate = '115200' # ボーレート指定
        ser.timeout = 0.5      # タイムアウト指定

        ser.open()
        ser.write("*idn?\n".encode("utf-8"))
        sleep(0.1)
        txtRcv = ser.read(256)
        print(' IDN: ' + txtRcv.decode('ascii'))

        ser.close()

    except:
        print(' COM Port Error: ' + ports )

# メイン関数
if __name__ == '__main__':
    # どちらかをコメントアウトします。
    main_LAN()
    #main_COM()

```

VISA サンプルプログラム例

```

# coding:utf-8
import visa          # pyVISA モジュール
from time import sleep

def main_VISA():
    # LAN Socket
    #   ipAddr = '172.22.44.208'      # IPv4 アドレス指定
    #   ports  = '5025'              # Socket ポート指定
    #   visaAddr = 'TCPIP0::' + ipAddr + '::' + ports + '::SOCKET'
    # LXI
    #   visaAddr = 'TCPIP0::' + ipAddr + '::inst0::INSTR'
    # USBTMC
    #   USBVid = '0x098F'            # Vendor ID
    #   USBPid = '0x0127'            # ProductID
    #   USBSN  = '15050232'          # Serial Number
    #   visaAddr = 'USB0::' + USBVid + '::' + USBPid + '::' + USBSN + '::INSTR'
    # GP-IB
    #   GPAddr = '8'                 # GP-IB Address
    #   visaAddr = 'GPIB::' + GPAddr

    # デバイスを検索し一覧表示します
    try:
        rm = visa.ResourceManager()
        instruments = rm.list_resources('*')
        if len(instruments) > 0:
            print('Device List')
            for num, device in enumerate(instruments):
                print(f' {num}: {device}')

    # 選択されたデバイスと通信を行います
    num = int(input('Select Device ? '))
    ass = len(instruments)
    if (num < len(instruments)):
        visaAddr = instruments[num]
        instr = rm.open_resource(visaAddr)
        if instr.resource_name.startswith('ASRL') or instr.resource_name.endswith('SOCKET'):
            instr.read_termination = '\n'
            instr.write_termination = '\n'
            instr.write('*idn?')
            print(instr.read())
            sleep(0.1)
            instr.close()
        else:
            print('No Device!')
    else:
        print('No Device!')

    # Error 処理
    except:
        print('Device Error')

    # デバイスとの通信を終了します。
    rm.close()

if __name__ == '__main__':
    main_VISA()

```

TEXIO ブランドの専用 API の使用について

Python から TEXIO ブランドの PSF/PS-A/PDS-A/LSA シリーズを USB で利用する場合は、直接 API の DLL を利用する必要があります。

- ・Windows が 64bit の場合は python が 64bit か 32bit かによって利用する DLL が異なります。
- ・python の標準の文字コードは UNICODE なので DLL が使える形に変換する。
- ・DLL の登録は ctypes で引数、restype で戻り値を登録する。
- ・文字列は encode で byte 型に変換してから c_char_p 型に変換して使用する
- ・構造体はクラスでフィールドを指定して作成し、Void ポインタで渡します

PSF / PS-A / PDS-A / LSA シリーズ用 API: IF-60.dll のアクセス例

```
# coding:utf-8
import ctypes
import sys
from time import sleep

def main():

    is64bit = sys.maxsize > 2 ** 32
    if is64bit:
        tmi_api = ctypes.windll.LoadLibrary('c:\windows\system32\IF_60.dll')
    else:
        tmi_api = ctypes.windll.LoadLibrary('c:\windows\SysWOW64\IF_60.dll')

    c_open = tmi_api.USB488_DeviceOpen
    c_open.argtypes = [ctypes.c_int]
    c_open.restype = ctypes.c_int

    c_close = tmi_api.USB488_DeviceClose
    c_close.argtypes = [ctypes.c_int]
    c_close.restype = ctypes.c_int

    c_send = tmi_api.USB488_Send
    c_send.argtypes = [ctypes.c_int, ctypes.c_char_p]
    c_send.restype = ctypes.c_int

    c_Receive = tmi_api.USB488_Receive
    c_Receive.argtypes = [ctypes.c_int, ctypes.c_int, ctypes.c_char_p]
    c_Receive.restype = ctypes.c_int

    PC_ADDR = ctypes.c_int(1)
    c_handle = c_open(PC_ADDR)

    if c_handle > 0:
        c_send(PC_ADDR, (ctypes.c_char_p("idn?\n".encode("utf-8"))))
        txtRcv = ctypes.create_string_buffer(255)
        c_Receive(PC_ADDR, ctypes.c_int(256), txtRcv)
        c_close(PC_ADDR)
        print(txtRcv.value.decode('ascii'))

if __name__ == '__main__':
    main()
```

PW-A / PAR-A シリーズ用 API:TMI_API.dll のアクセス例

```

# coding:utf-8
import ctypes
import sys
from time import sleep

def main():

    is64bit = sys.maxsize > 2 ** 32
    if is64bit:
        tmi_api = ctypes.windll.LoadLibrary('c:\windows\system32\tmi_api.dll')
    else:
        tmi_api = ctypes.windll.LoadLibrary('c:\windows\SysWOW64\tmi_api.dll')

    c_open = tmi_api.TMI_HandleOpen
    c_open.argtypes = [ctypes.c_char_p, ctypes.c_char_p]
    c_open.restype = ctypes.c_int

    c_close = tmi_api.TMI_HandleClose
    c_close.argtypes = [ctypes.c_int]
    c_close.restype = ctypes.c_int

    c_mainout = tmi_api.TMI_MainOutput
    c_mainout.argtypes = [ctypes.c_int, ctypes.c_char]
    c_mainout.restype = ctypes.c_int

    c_out = tmi_api.TMI_Out
    c_out.argtypes = [ctypes.c_int, ctypes.c_char_p]
    c_out.restype = ctypes.c_int

    c_in = tmi_api.TMI_In
    c_in.argtypes = [ctypes.c_int, ctypes.c_char_p]
    c_in.restype = ctypes.c_int

    c_voltage = tmi_api.TMI_Voltage
    c_voltage.argtypes = [ctypes.c_int, ctypes.c_char, ctypes.c_char, ctypes.c_double]
    c_voltage.restype = ctypes.c_int
    c_current = tmi_api.TMI_Current
    c_current.argtypes = [ctypes.c_int, ctypes.c_char, ctypes.c_char, ctypes.c_double]
    c_current.restype = ctypes.c_int

    c_mondata = tmi_api.TMI_MoniDataQ
    c_mondata.argtypes = [ctypes.c_int, ctypes.c_char, ctypes.c_void_p, ctypes.c_void_p, ctypes.c_void_p]
    c_mondata.restype = ctypes.c_int

    c_str = ctypes.create_string_buffer(256)

    c_handle = c_open((ctypes.c_char_p("PW-A".encode("utf-8"))), (ctypes.c_char_p("USB:1:1".encode("utf-8"))))
    if c_handle > 0:
        c_out(c_handle, (ctypes.c_char_p("SW1\n".encode("utf-8"))))
        sleep(0.5)
        c_out(c_handle, (ctypes.c_char_p("SW0\n".encode("utf-8"))))
        c_out(c_handle, (ctypes.c_char_p("*idn?\n".encode("utf-8"))))
        c_in(c_handle, c_str)
        print(c_str.value.decode('ascii'))
        c_close(c_handle)

if __name__ == '__main__':
    main()

```

LW シリーズ用 API:IF_50.dll のアクセス例

```

# coding:utf-8
import ctypes
import sys
from time import sleep

def main():

    is64bit = sys.maxsize > 2 ** 32
    if is64bit:
        tmi_api = ctypes.windll.LoadLibrary('c:\windows\system32\IF_50.dll')
    else:
        tmi_api = ctypes.windll.LoadLibrary('c:\windows\SysWOW64\IF_50.dll')

    c_open = tmi_api.LW_DeviceOpen
    c_open.argtypes = [ctypes.c_int, ctypes.c_int, ctypes.c_int, ctypes.c_int]
    c_open.restype = ctypes.c_int

    c_close = tmi_api.LW_DeviceClose
    c_close.argtypes = [ctypes.c_int]
    c_close.restype = ctypes.c_int

    c_mainout = tmi_api.LW_SetMainInput
    c_mainout.argtypes = [ctypes.c_int, ctypes.c_char]
    c_mainout.restype = ctypes.c_int

    c_monitor = tmi_api.LW_GetMonitor
    c_monitor.argtypes = [ctypes.c_int, ctypes.c_int, ctypes.c_void_p]
    c_monitor.restype = ctypes.c_int

    class _mon_data(ctypes.Structure):
        _fields_ = [("curr", ctypes.c_double), ("volt", ctypes.c_double), ("pow", ctypes.c_double)]

    monitor_data = _mon_data(0.0, 0.0, 0.0)

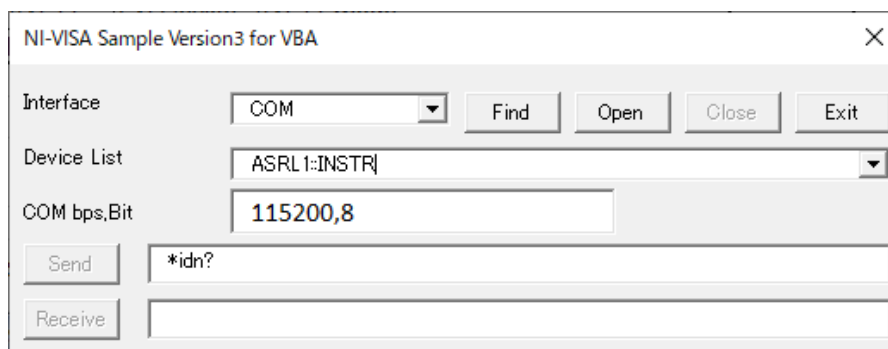
    c_handle = c_open(2, 0, 1, 1)
    if c_handle > 0:
        c_mainout(c_handle, 1)
        sleep(0.5)
        c_monitor(c_handle, ctypes.c_int(1), ctypes.byref(monitor_data))
        c_mainout(c_handle, 0)
        c_close(c_handle)
        print("Monitor:%f[V], %f[A], %f[W]" % (monitor_data.volt, monitor_data.curr, monitor_data.pow))

if __name__ == '__main__':
    main()

```

Python で GUI を使う場合について(VISA を使用した例)

GUI環境は標準のライブラリとしてTkinterが存在します。使用の宣言のみで利用できるため追加のインストール不要です。本サンプルでは当社で配布している ExcelVBA の VISA 用サンプルプログラムと似た構成の物を作成します。



ExcelVBA の作成例



Python/Tkinter での GUI 作成例

Python での画面の構成としては、ボタン 6 個、ドロップダウンリスト、テキストボックス 3 個、ラベル 2 個となっています。また単純なインターバルでの動作も盛り込んであります。

```
# coding:utf-8
import visa          # VISA モジュール
import tkinter as tk # tk モジュール
import time
from time import sleep
from tkinter import ttk

rm = visa.ResourceManager() #rm と instr をグローバルとして定義
instr = None

def Find_clicked():
    try:
        # デバイスを検索し一覧表示します
        cbDevList['values']=['']

        instruments = rm.list_resources('*')
        if len(instruments) > 0:
            cbDevList['values']=instruments
            cbDevList.current(0) # Error 処理
            btn3['state'] = tk.NORMAL
    except:
        print('Device Error')

def Exit_clicked():
    rm.close() # デバイスとの通信を終了します。
    root.destroy()
```

```

def Open_clicked():
    global instr
    visaAddr = cbDevList.get()
    if( visaAddr == '' ):
        print(' No Device!')
    else:
        try:
            instr = rm.open_resource(visaAddr)
            if instr.resource_name.startswith(' ASRL'):
                instr.read_termination = '\n'
                instr.write_termination = '\n'
                instr.baud_rate = int(txtOption.get())

            if instr.resource_name.endswith(' SOCKET'):
                instr.read_termination = '\n'
                instr.write_termination = '\n'

            btn2['state'] = tk.DISABLED      # OPEN/EXIT 禁止、CLOSE/Send/Recv 有効
            btn3['state'] = tk.DISABLED
            btn4['state'] = tk.NORMAL
            btn5['state'] = tk.NORMAL
            btn6['state'] = tk.NORMAL

            txtRecv.delete('1.0', tk.END)
            txtRecv.insert (tk.END, ' Device Opened!')
# Error 処理
            except:
                txtRecv.delete('1.0', tk.END)
                txtRecv.insert (tk.END, ' Device Open Error!')

def cmb_Sel(event):
    txtOption.delete(0, tk.END)
    cbuff = cbDevList.get()
    if(' ASRL' in cbuff):
        txtOption['state'] = tk.NORMAL
        txtOption.insert(tk.END, ' 115200')
    else:
        txtOption['state'] = tk.DISABLED

def Close_clicked():
    global instr

    btn2['state'] = tk.NORMAL      # OPEN/EXIT 有効、CLOSE/Send/Recv 禁止
    btn3['state'] = tk.NORMAL
    btn4['state'] = tk.DISABLED
    btn5['state'] = tk.DISABLED
    btn6['state'] = tk.DISABLED

    instr.close()

def Send_clicked():
    global instr

    cbuff = txtSend.get()
    instr.write(cbuff)
    txtRecv.delete('1.0', tk.END)
    if ('?' in cbuff) :
        txtRecv.insert (tk.END, instr.read())

```



```

def Recive_clicked():
    global instr

    txtRecive.delete('1.0', tk.END)
    txtRecive.insert(tk.END, instr.read())

def interval_work():
    # インターバルで行う処理を記述
    root.title('pyVisa Sample ' + time.ctime()) # ここでは時刻をタイトルに表示
    root.after(1000, interval_work) # インターバルの時間(ms)を指定
    # 指定しなければインターバルは終了

root = tk.Tk()
root.geometry('530x160')
root.title('pyVisa Sample')

label1 = tk.Label(root, text = 'Device List : ')
label1.grid(row=0, column=0)
label2 = tk.Label(root, text = 'Serial Baud : ')
label2.grid(row=1, column=0)

cbDevList= ttk.Combobox(root , width=50 )
cbDevList.grid(row=0 , column=1)
cbDevList.bind('<<ComboboxSelected>>', cmb_Sel)

txtOption = ttk.Entry(root , width=50 , state =tk.DISABLED )
txtOption.grid(row=1 , column=1)
txtSend = ttk.Entry(root , width=50 )
txtSend.grid(row=2 , column=1)
txtSend.insert(tk.END, '*idn?')

txtRecive = tk.Text(root , width=42, height=5 )
txtRecive.grid(row=3 , column=1)

btn1 = tk.Button(master=root, text=' Find ' , command=Find_clicked)
btn1.grid(row = 0 , column=2)
btn2 = tk.Button(master=root, text=' Exit ' , command=Exit_clicked)
btn2.grid(row = 0 , column=3)
btn3 = tk.Button(master=root, text=' Open ' , command=Open_clicked, state =tk.DISABLED )
btn3.grid(row = 1 , column=2)
btn4 = tk.Button(master=root, text=' Close ' , command=Close_clicked, state =tk.DISABLED )
btn4.grid(row = 1 , column=3)
btn5 = tk.Button(master=root, text=' Send ' , command=Send_clicked, state =tk.DISABLED )
btn5.grid(row = 2 , column=0)
btn6 = tk.Button(master=root, text=' Recive ' , command=Recive_clicked, state =tk.DISABLED )
btn6.grid(row = 3 , column=0)

interval_work() #インターバル動作を開始するときに実行

root.mainloop()

```

本プログラムはあくまでもサンプルとなっております。環境の違いなどで正しく動作しないこともありますのでご注意ください。