```
#!/usr/bin/python


# Date : 2014-3-6
# Writen By : Manson Engineering Industical Limited.
#
# This is sample program of remote control Manson HCS series power supply under Raspberry Pi.
# The pySerial modulde is required to run this program. you can install by using
#
#   sudo apt-get install python-serial
#
# Raspbian version : Kernel 3.10.30+
#
# USB driver : Kernel build-in cp210x driver.
#
# Testing model: HCS-3102, 1-36VDC Max 5A


import serial, time     # load serial communication and time control module



# Configure Port as baudrate 9600, Data 8bits, Parity None, 1 Stop bit. /dev/ttyUSB0 is virtual serial
port
ser = serial.Serial('/dev/ttyUSB0', 9600, timeout=0,
            bytesize=serial.EIGHTBITS,
            parity=serial.PARITY_NONE,
            stopbits=serial.STOPBITS_ONE)

x=1 # set dummy variable

if ser.isOpen():  #exit if USB port is not open
    try:
        ser.flushInput() #flush input buffer, discarding all its contents
        ser.flushOutput() #flush output buffer, aborting current output


        #Set PSU OUTPUT off
        print("Set PSU OUTPUT off")
        while x:            # Loop to confirm PSU is received the command correclty.
          ser.write("SOUT1\r")      # Send SOUT1 to PSU. Each command must end with '\r'.
          time.sleep(0.5)       # Give serial port sometime to receive the command.
          response=ser.readline()   # Get reply from PSU
          if response == '':continue
          print(response)
          break
        time.sleep(3)

        #Set PSU output to 10V
        print("Set PSU OUTPUT to 10V")
        while x:
          ser.write("VOLT100\r")    # Send set voltage command. 100 means 10.0V
          time.sleep(0.5)
          response=ser.readline()
          if response == '':continue
          print(response)
          break
        time.sleep(3)

        #Get PSU setting
        print("Get PSU OUTPUT setting")
        while x:
          ser.write("GETS\r")       # Get setting information
          time.sleep(0.5)
          response=ser.readline()
          if response == '':continue

          voltage=response[:3]  # PSU return string in format "100200", that means 10.0V, 2.00A,
          current=response[3:6] # then use first 3 data for voltage and next 3 data for current.
          print(voltage[:2]+'.'+voltage[2:]+'V')
          print(current[:1]+'.'+current[1:]+'A')
                  break
```

```
        time.sleep(3)


        #Set PSU OUTPUT on
        print("Set PSU OUTPUT on")
        while x:
          ser.write("SOUT0\r")  # Set PSU output On
          time.sleep(0.5)
          response=ser.readline()
          if response == '':continue
          print(response)
          break
        time.sleep(3)

        #Set PSU output to 15V
        print("Set PSU OUTPUT to 15V")
        while x:
          ser.write("VOLT150\r")  # change Votlage setting
          time.sleep(0.5)
          response=ser.readline()
          if response == '':continue
          print(response)
          break
        time.sleep(3)


        #Set PSU output to 2A
        print("Set PSU OUTPUT to 2A")
        while x:
          ser.write("CURR200\r")  # change Current setting
          time.sleep(0.5)
          response=ser.readline()
          if response == '':continue
          print(response)
          break
        time.sleep(3)

        ser.close()

    except Exception, e1:
        print "Error communicating...: " + str(e1)


else:
    print "Cannot open serial port "
```