#############################################################################  
import visa  
import time  
  
# Establish communication  
rm = visa.ResourceManager()  
devlist = rm.list\_resources()  
  
for  s in devlist:  # look through all devices in the list  
    dev =  rm.open\_resource(s)  # make a "device" object to represent this instrument  
    instr\_id = dev.query('\*IDN?')  # ask the instrument for its ID  
    print 'Found ', instr\_id  # tell user what ID this one is  
    if instr\_id.find('Agilent')>-1 :  # is this the one we are looking for?  
        print 'Happy with this instrument; going ahead.' # indicate success  
        break  # found what we wanted; leave loop, skipping over 'else' section  
    else : # (this else belongs to the 'if') some other instrument; close and try the next one  
        print 'You are not my mother.  You are a SNORT.'  
        dev.close() # end communication with this device and move to the next in the loop  
else:  # (this else belongs to the 'for') will only arrive here if we never found the device we wanted  
    print 'Failed to find device'  # print error mesage  
    exit()   # and quit  
  
dev.timeout = 3000 # specify 3-second timeout limit  
  
# Construct a code that maps voltage into character  
alphabet = " abcdefghijklmnopqrstuvwxyz~"  
vstart = 500.0  
vstep = 100.0  
  
# Configure Device  
#dev.write('CONF:FREQ')  # Configure for frequency readings  
  
# Measure the incoming voltages and decode the message  
message = []  
result = float(dev.query('MEAS:FREQ?'))  # Measure freq  
print "Initial frequency:", result  
  
while True:  
result = float(dev.query('MEAS:FREQ?'))  # Measure freq  
position = int(round((result - vstart) / vstep))  
if position < 0 or position >= len(alphabet):  
        print("Frequency out of range")  
        break  
    char = alphabet[position]  
  
    print char  
    if char == "~":  
        break  
    message.append(char)  
print("".join(message))  # Print the decoded message  
###############################################################################