```
# This module demonstrates the use of HartDll from HartTools 7.5.
# It is kept very simple and shows the basic procedure for loading
# and using a windows dll in python.
# The Python Version which I used for these tests was 3.7.9, which
# is the latest stable version in Python.
# If you are having any questions, please send email to:
# info@borst-automation.de
# ----------------------
# Borst Automation
# Embedded Solutions
# Walter Borst
# Kapitaen-Alexander-Str.  39
# 27472 Cuxhaven, Germany
# Fon: +49 (0)4721 6985 100
# Fax: +49 (0)4721 6985 102
# info@borst-automation.de
# www.borst-automation.de

import ctypes
from ctypes import *

# Data Structures
class STConnectionData(Structure):
    _fields_ = [("ManIdByte", c_ubyte),
                ("DevId", c_ubyte),
                ("NumPreambs", c_ubyte),
                ("CmdRevNum", c_ubyte),
                # ---------------------
                ("SpecRevCode", c_ubyte),
                ("SwRev", c_ubyte),
                ("HwRev", c_ubyte),
                ("HartFlags", c_ubyte),
                # ---------------------
                ("ServiceCode", c_ubyte),
                ("RespCode1", c_ubyte),
                ("RespCode2", c_ubyte),
                ("UsedRetries", c_ubyte),
                # ---------------------
                ("DeviceInBurstMode", c_ubyte),
                ("ExtDevStatus", c_ubyte),
                ("CfgChCount", c_ushort),
                # ---------------------
                ("MinNumPreambs", c_ubyte),
                ("MaxNumDVs", c_ubyte),
                ("ManuIdUshort", c_ushort),
                # ---------------------
                ("LabDistID", c_ushort),
                ("DevProfile", c_ubyte),
                ("Reserved", c_ubyte),
                # ---------------------
                ("UniqueId_1", c_ubyte),
                ("UniqueId_2", c_ubyte),
                ("UniqueId_3", c_ubyte),
                ("UniqueId_4", c_ubyte),
                ("UniqueId_5", c_ubyte)]
# Data
comport = 1
address = 0
# Use the above defined class
connectionData = STConnectionData()
# Let user know that operation is running
print()
print(" Communication with Hart")
# Load the dll
Hartdll = windll.LoadLibrary("BaHartDrv76.dll")
# Register license
Hartdll.BHDrv_ValidateLicense("30-Days-Trial-User-License".encode(),
                              "Ea58v60F-x3jk-wi9n-RrI3-7c072aA6ae0B".encode())
# Open a channel on com port
myhandle = Hartdll.BHDrv_OpenChannel(comport)
# Connect to a device if it is a valid com port
# Address = 0, WaitForService = 1, NumRetries = 2
if myhandle != -0x1:
    print(" Connecting to device at address ", address)
    print(" Waiting for service completion ..")
    myservice = Hartdll.BHDrv_ConnectByAddr(myhandle, address, 1, 2)
```

```
    if myservice != -0x1:
        Hartdll.BHDrv_FetchConnection(myservice, byref(connectionData))
        if connectionData.ServiceCode == 5:
            print(" ------ Device Data ------")
            print(" Manufacturer Id: ", connectionData.ManIdByte)
            print(" Device Id: ", connectionData.DevId)
            print(" Command Response: ", connectionData.RespCode1)
            print(" Device Status: ", connectionData.RespCode2)
            print(" ------ Hart DLL ---------")
            print(" Service Completion Code: ", connectionData.ServiceCode)
        else:
            print(" ------ Hart DLL ---------")
            print(" Service Completion Code: ", connectionData.ServiceCode)
    else:
        print(" HartDLL out of service handles!")
else:
    print(" Could not open com port: ", comport)
# Close channel if valid
if myhandle != -0x1:
    Hartdll.BHDrv_CloseChannel(myhandle)
input(" Press <Enter> Key to finish ...")
```