Why this?

The PDXC_COMMAND_LIB provides all APIs for controlling. But most of them are not frequently used. This document aims to conclude some useful functions from the lib.

examples

constructor

```
def __init__(self):
    #the constructor will load the dll and create an pdxcobj
        lib_path = "./PDXC_COMMAND_LIB_win64.dll"
        if not pdxc.isLoad:
            pdxc.load_library(lib_path)
        self.hdl = -1
        @staticmethod
def load_library(path):
    pdxc.pdxcLib = cdll.LoadLibrary(path)
    pdxc.isLoad = True
```

acquire the connected devices

```
def ListDevices():
        """ List all connected pdxc devices
        Returns:
           The pdxc device list, each device item is serialNumber/COM
        """
```

Open a device

```
def Open(self, serialNo, nBaud, timeout):
        """ Open device
        Args:
            serialNo: serial number of pdxc device
            nBaud: bit per second of port
            timeout: set timeout value in (s)
        Returns:
            non-negative number: hdl number returned Successful; negative number:
failed.
        """
```

check whether device is opened or not

```
def IsOpen(self, serialNo):
        """ Check opened status of device
        Args:
            serialNo: serial number of device
        Returns:
            0: device is not opened; 1: device is opened.
        """
```

close a device

```
def Close(self):
        """ Close opened device
        Returns:
            0: Success; negative number: failed.
        """
```

set the daisy chain when multiple devices run simultaneously

```
def SetDaisyChain(self, index):
        """ Set device position in daisy-chain.
        Args:
            index: index in daisy chain (0:Single Mode, 1:Main, 2 -12 : Secondary1
- Secondary11)
        Returns:
            0: Success; negative number: failed.
        """
```

get status

```
def GetCurrentStatus(self, secondary, status):
    """ In D - sub Mode, it will return strings consist of ERR, KP, KI, KD,
Current Loop, Velocity Level(open loop),
    Step size(open loop), Speed(Closed loop), Jog Step(closed loop), Home Flag and
Abnormal Detection Status,
    parameters in one command, results will be separated by ',' for each segment.
In SMC Mode, it will return
    strings consist of ERR, Velocity and Step Size for Channel 1, Velocity and
Step Size for channel 2.
    Args:
        secondary: index in daisy chain (0:Single Mode or Main, 1 -11 : Secondary1
- Secondary11)
        status: device status
    Returns:
        0: Success; negative number: failed.
    """
```

get position

```
def GetCurrentPosition(self, secondary, position):
    """ Get current position counter.It only returns position value in D - SUB
mode, other will return warnings
    Args:
        secondary: index in daisy chain (0:Single Mode or Main, 1 -11 : Secondary1
- Secondary11)
        position: current position: PDX1[-10,10]mm; PDX2[-2.5,2.5]mm;
PDXZ1[-2.25,2.25]mm; PDXR:[-999.9,999.9] °
    Returns:
        0: Success; negative number: failed.
    """
```

switch loop model

```
def SetLoop(self, secondary, loop):
    """ Switch closeloop and open loop
    Args:
        secondary: index in daisy chain (0:Single Mode or Main, 1 -11 : Secondary1
- Secondary11)
        loop: loop type (1 : open loop (default), 0 : close loop.)
    Returns:
        0: Success; negative number: failed.
    """
```

move forward and backward

```
def SetOpenLoopMoveForward(self, secondary, pulses, channel):
        """ Set Open Loop Move Forward
        Args:
            secondary: index in daisy chain (0:Single Mode or Main, 1 -11 :
Secondary1 - Secondary11)
            pulses: pulses of move channel:SMC[1,65535]; PD2/PD3[1,400000]
            channel: Move fordward channel (0 : channel 1, 1 : channel 2, others
:both channels)
        Returns:
            0: Success; negative number: failed.
        """
def SetOpenLoopMoveBack(self, secondary, pulses, channel):
        """ Set Open Loop Move Forward
        Args:
            secondary: index in daisy chain (0:Single Mode or Main, 1 -11 :
Secondary1 - Secondary11)
            pulses: pulses of move channel:SMC[1,65535]; PD2/PD3[1,400000]
```

```
            channel: Move fordward channel (0 : channel 1, 1 : channel 2, others
:both channels)
        Returns:
            0: Success; negative number: failed.
        """
```

For more detailed information about APIs for each type of device,please refer to the official document *PDXC_COMMAND_LIB.py*