Reference 'TDK-Lambda Zup Power Supplies User Manual, IA549-04-01R'.

* Available at <https://product.tdk.com/system/files/dam/doc/product/power/switching-power/prg-power/instruction_manual/zup-user-manual.pdf>.
* Copied to [P:\Test\Engineers\JTAG\TDK-Lambda Zup Power Supplies\TDK-Lambda Zup Power Supplies User Manual, IA549-04-01R.pdf](file:///P:\Test\Engineers\JTAG\TDK-Lambda%20ZUp%20Power%20Supplies\TDK-Lambda%20ZUp%20Power%20Supplies%20User%20Manual,%20IA549-04-01R.pdf).

Purpose of this Python Zup library is to eliminate occasional Zup programming glitches, wherein Zup supplies don't turn on and/or off as programmed:

* Occasionally some Zup supplies fail to power on or off as programmed by Rafael's Windows batch files.
* When failing to power on, the Boundary Scan test will fail.
* When failing to power off, technicians may remove a CCA from the fixture jig while still partly powered.
* And it's always sub-optimal for a CCA to have some, but not all, of its power supplies applied to it.

Another purpose is to allow reading Zup status registers & model IDs:

* This allows checking correct application of voltage & current; 5.0V @ 1.0A was programmed, but what's actually being output by the Zup?
  + Actual Voltages/Amperages can be read from each Zup & compared to Set Voltages/Amperages.
  + Querying status allows checking the Zup's Alarm & Error Codes registers, which indicate if programming or communication errors occurred.

FWIW, suspect most of the on/off glitches are due to 'TDK-Lambda Zup Power Supplies User Manual, IA549-04-01R', paragraph 5.6.1:

* The average command processing time of the Zup Series is 15mSec.
* It is not recommended to send strings of commands to the Zup power supply without considering the processing time.
  + For query commands ( ? , ! ) wait until the Zup reply message has been completed before sending a new command.
  + 10mSec minimum delay is required before sending: ADRxx; command.
  + Additional 30mSec after: ADRxx; command.
* There's no simple method to implement 10, 15 or 30mSec processing delays with a Windows batch file, so Rafael programs the Zups at 2400 Baud, which appears mostly slow enough to provide processing delays, and mostly works.
  + Power on/off glitches appear more common at any Baud rate > 2400.
  + The obvious solution is to slow the Zup's Baud rate to its minimum of 300, but at < 2400 Baud, Zup supplies don't turn on/off highly simultaneously; there's a sub-optimal & noticeable time lag between each one activating/deactivating.
  + It's best for a CCA to apply all its supplies as simultaneously as possible, which can't be done at < 2400 Baud, and can't be reliably done at >= 2400 Baud with a batch file.