Report on the [pyDevSup](https://github.com/mdavidsaver/pyDevSup) module

# Architecture

The module consists of these code areas:

1. There are 5 example programs, altogether, that operate the IOC.

These launch the IOC from a command script (as is normal EPICS practice).

1. The IOC code softIocPy and pDevSup.

These are written in C and depend on EPICS base.

The C code makes calls into Python code, invoking the devsup.\_init Python function.

1. The Python module code devsup.

It loads the \_dbapi module and calls in to its C code.

The \_init function creates a temporary file containing the definitions of ‘Python Device’ which are needed by the IOC (above), and pushes the contents in to the \_dbapi module.

1. A shared library \_dbapi(.so or .pyd).

This is written in C and depends on EPICS base.

This dependency is wholly *separate* from the dependency of the IOC code on EPICS base.

The C code here in turn makes calls in to the Python code (3), in response to the ‘py’ command(s) that are present in the script (1).

The inter-operation of 2, 3 and 4 means that the architecture is extraordinarily complex and difficult to debug.

# Major design issue

The above architecture means that the EPICS database is apparently loaded twice, once into the IOC code (2) and once into the \_dbapi code (4). If these copies are discrepant from each other, the application won’t work.

This appears to work on Linux, although I am at a loss to understand *how*.

It does not work on Windows, in that EPICS commands like ‘Python Device’ and ‘py’ that have been programmed in to (4) are unavailable for scripted use by (2).

# Python 2 -> 3 issues

It is clear that the module has not been thoroughly tested with Python 3.

Several 2->3 compatibility issues have been identified which needed to be corrected, before the code would run.

In particular, the examples were launched with the script command:

#!../../bin/linux-x86/softIocPy2.7

Which is easily changed to:

#!../../bin/linux-x86\_64/softIocPy3.10

But indicates the examples have not been operated since 32-bit Linux and Python 2.7.

N.B. Python 2.7 was released in July 2010, and support ended in Jan 2020.

# Support issues

There’s no distribution of the module through pip or wheel.

The 1.2 tagged library could build on Linux, but the head failed.

It is also noted that there haven't been updates or pull requests to the GitHub module in the last 2 or 3 years.

These points tend to give the impression that this is not a mainstream or well-supported EPICS IOC implementation library.

# Windows build

The module has not previously been used on Windows, and some build changes were required.

I’m aware that ITER’s requirement is for use on Linux, and not Windows.

However, it is a project objective to make the RGA software available as open source and available to both Windows and Linux users.