Alicat Python and Command Prompt Communication

Setup

Alicat Python and Python-driven command prompt (Windows Powershell) drivers have been developed principally by Pat Fuller at NuMat Technologies and are available for download from github.com:
https://github.com/numat/alicat
In order to run the command prompt/Powershell driver, Python will first have to be installed on the computer. The latest stable Python 3 release is the recommended version to install:
https://www.python.org/downloads/
Once Python has been installed, the simplest way to install the python/command prompt drivers is to command the following in the command prompt:

```
git clone https://github.com/numat/alicat.git
cd alicat
python setup.py install
```

If the Python PIP package manager is installed and enabled, this is a simpler method:

```
pip install alicat
```

Command Prompt (Powershell) Usage:

A menu of possible arguments and usage information within the command prompt can be accessed by entering “alicat --help” or “alicat -h”:

```
alicat --help

positional arguments:
  port                  The target serial port or TCP address. Default
                        '/dev/ttyUSB0'. Use e.g. 'tcp://192.168.1.100:4000 to
                        read devices routed through a converter.

optional arguments:
  -h, --help            show this help message and exit
  --address ADDRESS, -a ADDRESS
                        The device address, A-D. Should only be used if
                        multiple flow controllers are connected to one port.
  --set-gas SET_GAS, -g SET_GAS
                        Sets the gas type. Supported gas types are: 'Air',
                        'Ar', 'CH4', 'CO', 'CO2', 'C2H6', 'H2', 'He', 'N2',
                        'N2O', 'Ne', 'O2', 'C3H8', 'n-C4H10', 'C2H2', 'C2H4',
                        'i-C2H10', 'Kr', 'Xe', 'SF6', 'C-25', 'C-10', 'C-8',
                        'C-2', 'C-75', 'A-75', 'A-25', 'A1025', 'Star29',
                        'P-5'
  --set-flow-rate SET_FLOW_RATE, -f SET_FLOW_RATE
                        Sets the target flow rate of the controller.
```

```
  --set-pressure SET_PRESSURE, -p SET_PRESSURE
                        Sets the target pressure of the controller.
  --stream, -s          Sends a constant stream of flow controller data,
                        formatted as a tab-separated table.
```

To poll an Alicat that is the only one connected to your target COM port, commanding only the port name will return data from the Alicat:

```
alicat COM7
```

To poll a device with unit ID A, your command should look like this:

```
alicat COM7 –a A
```

and the response from either of these commands should look similar to this:

```
{
  'setpoint': 0.0,         # Setpoint, either mass flow rate or pressure
  'control_point': 'flow', # Either 'flow' or 'pressure'
  'gas': 'Air',            # Can be any option in `flow_controller.gases`
  'mass_flow': 0.0,        # Mass flow (in units specified at time of purchase)
  'pressure': 25.46,       # Pressure (normally in psia)
  'temperature': 23.62,    # Temperature (normally in C)
  'total_flow': 0.0,       # Optional. If totalizer function purchased, will be
included
  'volumetric_flow': 0.0   # Volumetric flow (in units specified at time of purchase)
}
```

Python Usage:
This command will open a serial connection to an Alicat device on the specified port:

```
from alicat import FlowController
flow_controller = FlowController(port='/dev/ttyUSB0')
print(flow_controller.get())
```

If the Alicat is communicating on the specified port, it will return the same dictionary as when polled via Powershell.

Using this with Windows COM port COM1, for example, would look like this:

```
from alicat import FlowController
flow_controller = FlowController(port='COM1')
print(flow_controller.get())
```