

Synergy488 Kit for Synergy Nano and Synergy Quattro GPIB/IEEE 488 Applications



Introduction

Tidal Engineering's Synergy Controllers, including the Synergy Micro 2, Synergy Quattro, and the 1/4 DIN Synergy Nano provide state-of-the-art usability and connectivity for environmental test control and data acquisition. They combine the functions of a chamber controller and a data logger and are designed to improve test efficiency by supporting both factory automation and test and measurement protocols and standards. With the flexibility afforded with their multiple communication ports; Ethernet (10/100 Base-T), GPIB/IEEE 488, and RS-232, these controllers are perfect for today's dynamic testing environments.

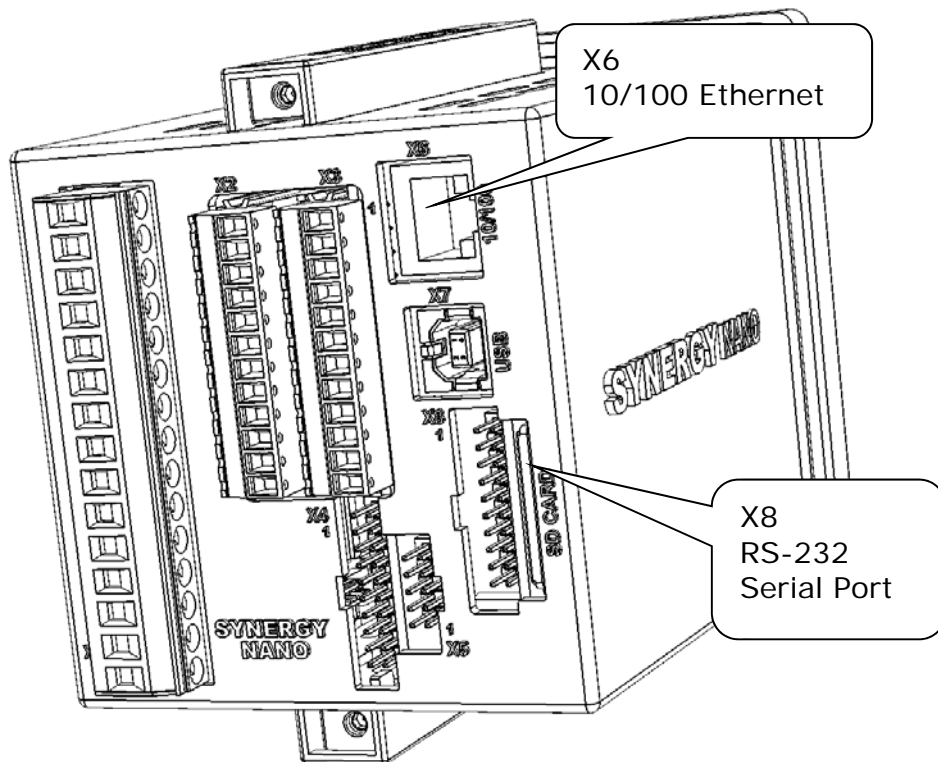
The Synergy Nano and Synergy Quattro controllers are equipped with both 10/100 Base-T Ethernet and RS-232 ports. The optional Synergy488 communications adapter provides plug and play GPIB/IEEE 488 communications for these two controllers. This application note covers the setup and configuration of the Synergy488, P/N TE1588 adapter for use with Synergy Nano and Synergy Quattro controllers.

Software Setup and checkout is included for the following applications:

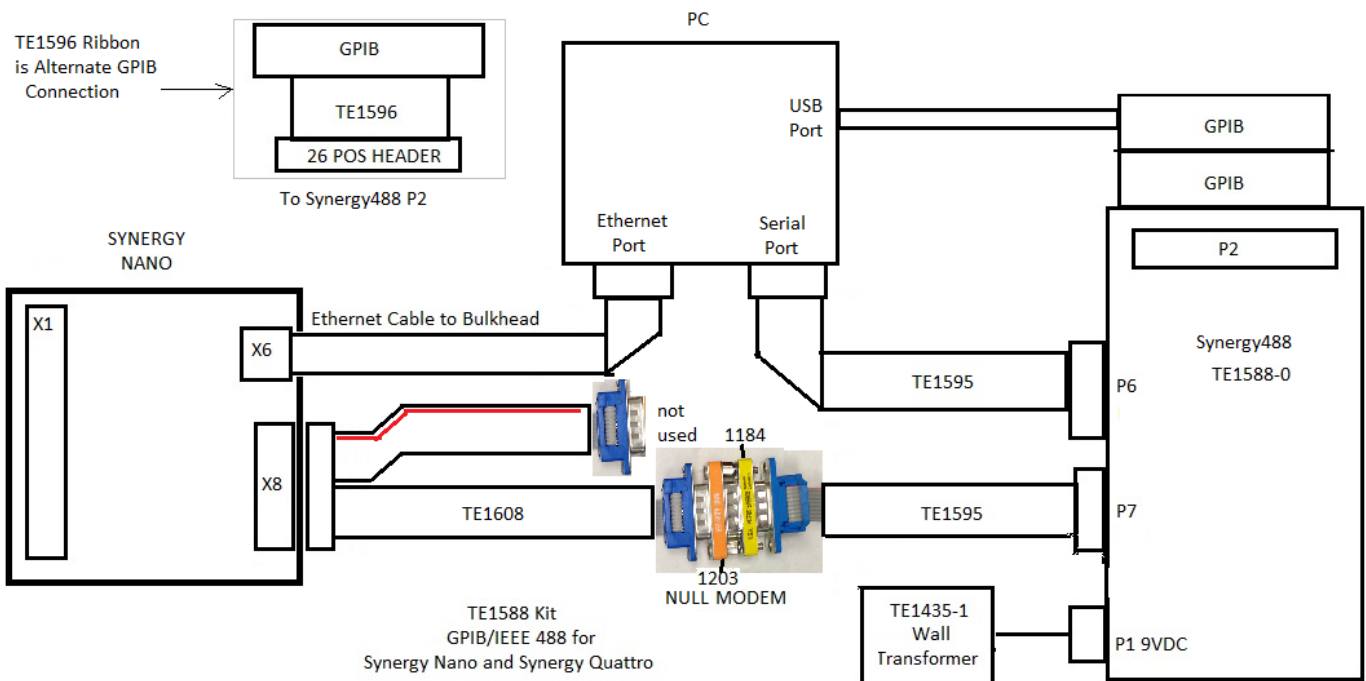
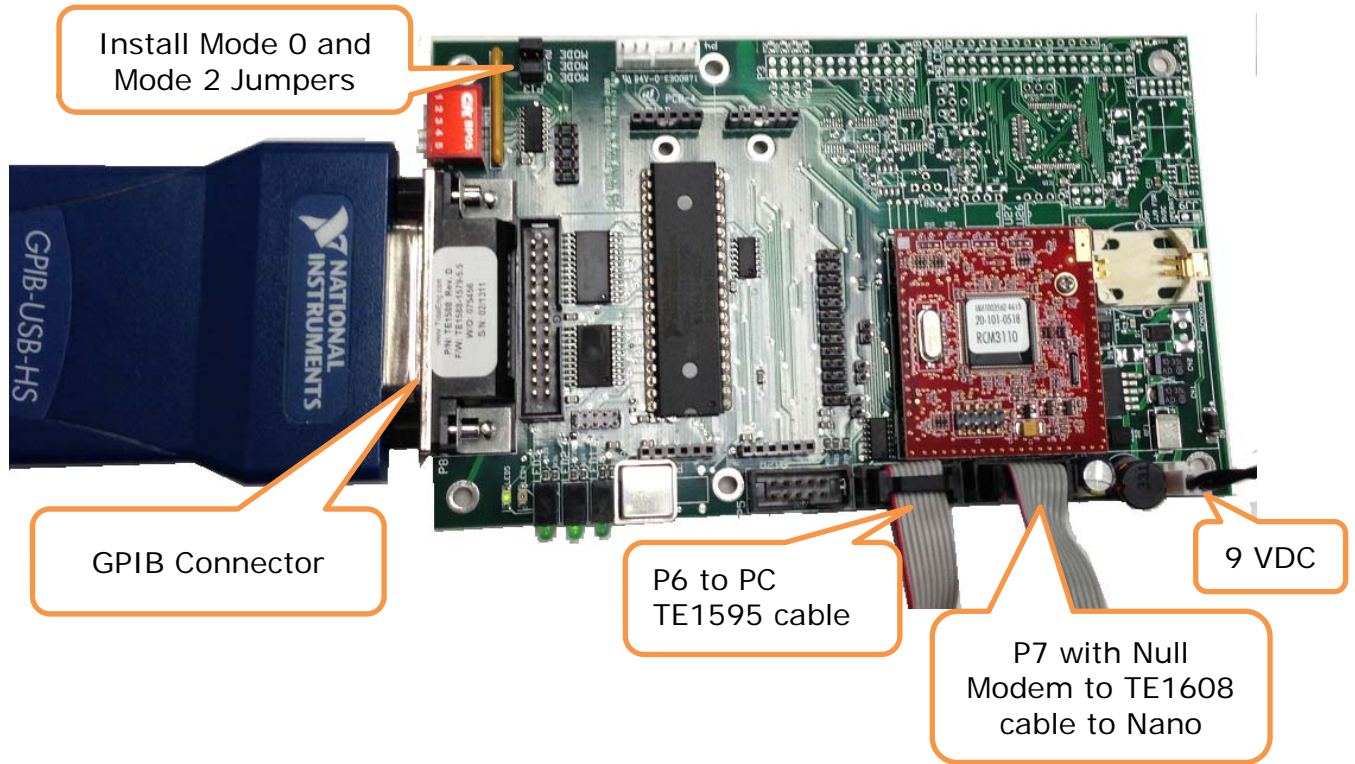
- Tidal Engineering Synergy Manager software
- Tidal Engineering SimpleComm software
- Python

TE1588 Synergy488 Kit contents (for Synergy Nano and Synergy Quattro)

Item	Qty.	Part	Description
1	1	TE1588-0	Single Board Computer with Synergy488 firmware
2	1	TE1596	Cable, IEEE 488 to 26-pos box header, 6 Feet
3	2	TE1595	Cable, 10 Conductor, Box Header, 6 feet
4	1	TE1435-1	Power Supply, 9 VDC with Molex connector
5	1	1184	Gender Changer, M-M

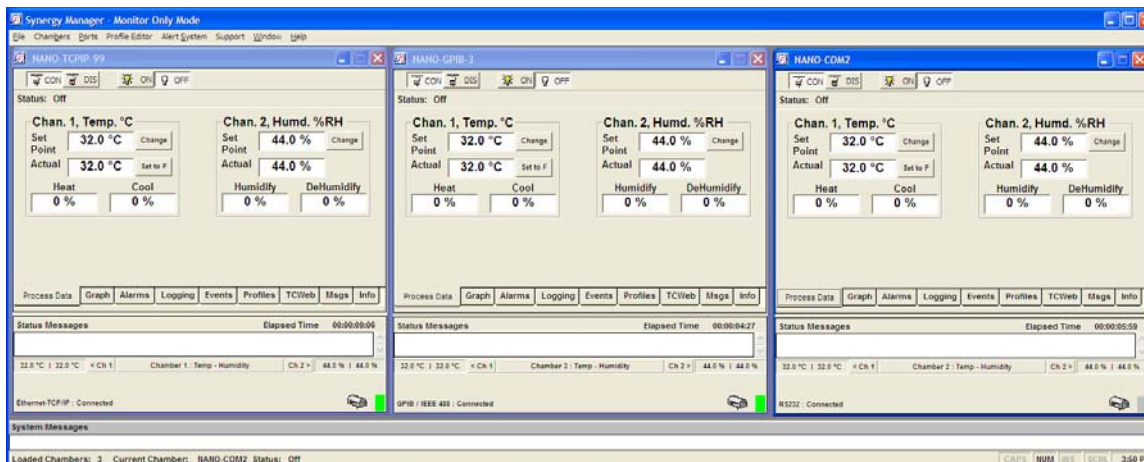
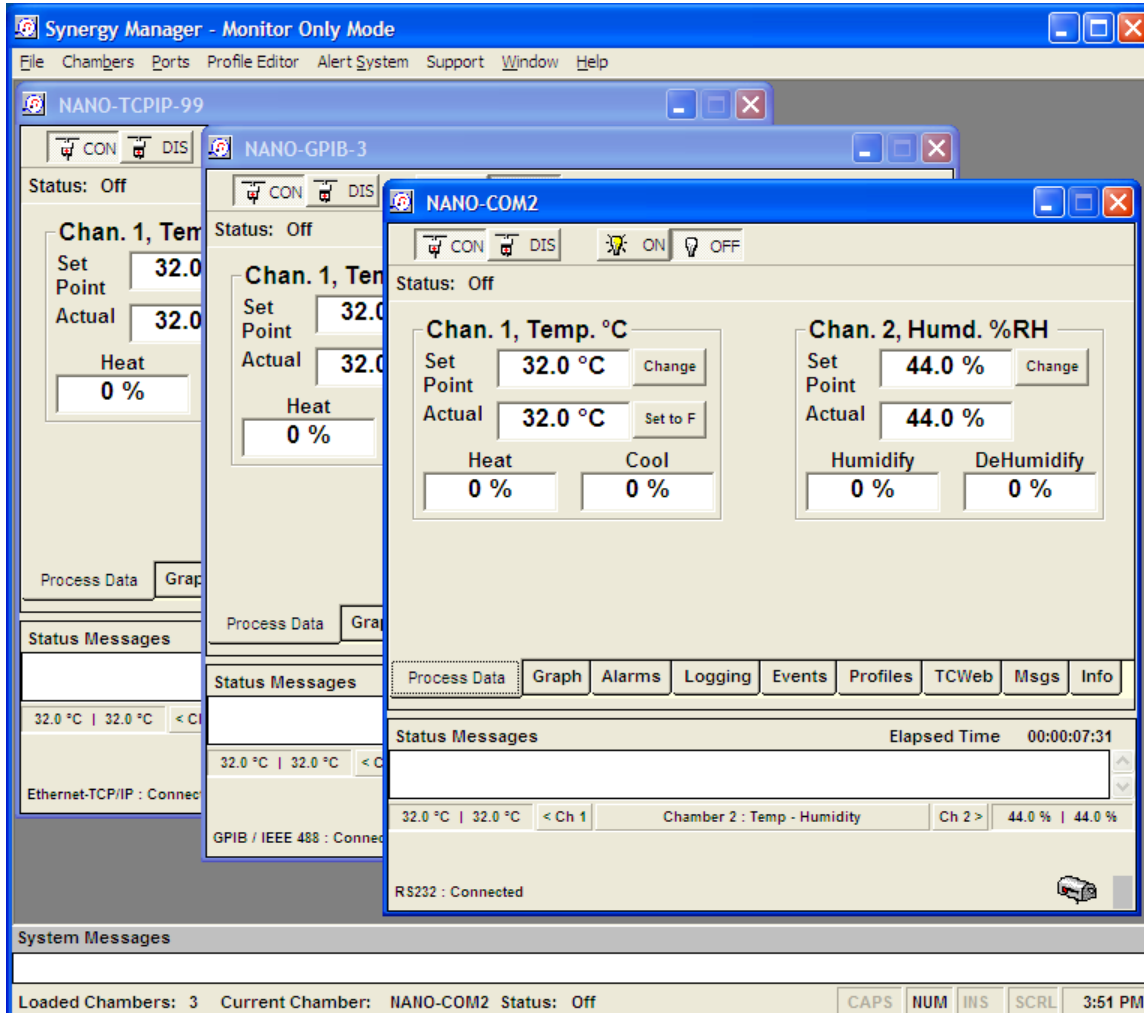


Connect and configure the Synergy488 as follow:



Synergy Manager Setup

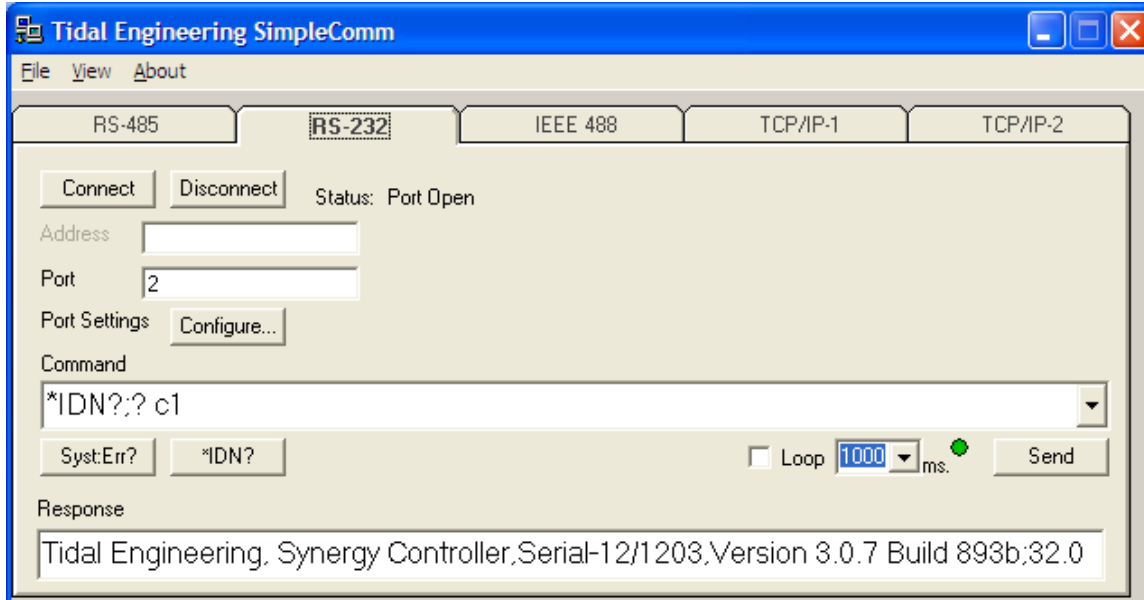
The following figures show three simultaneous connections between Synergy Manager running on the PC and the Synergy Nano. The GPIB connection and the Serial connection are passing thru the Synergy488 (TE1588) board. The TCP/IP connection is going thru the network.



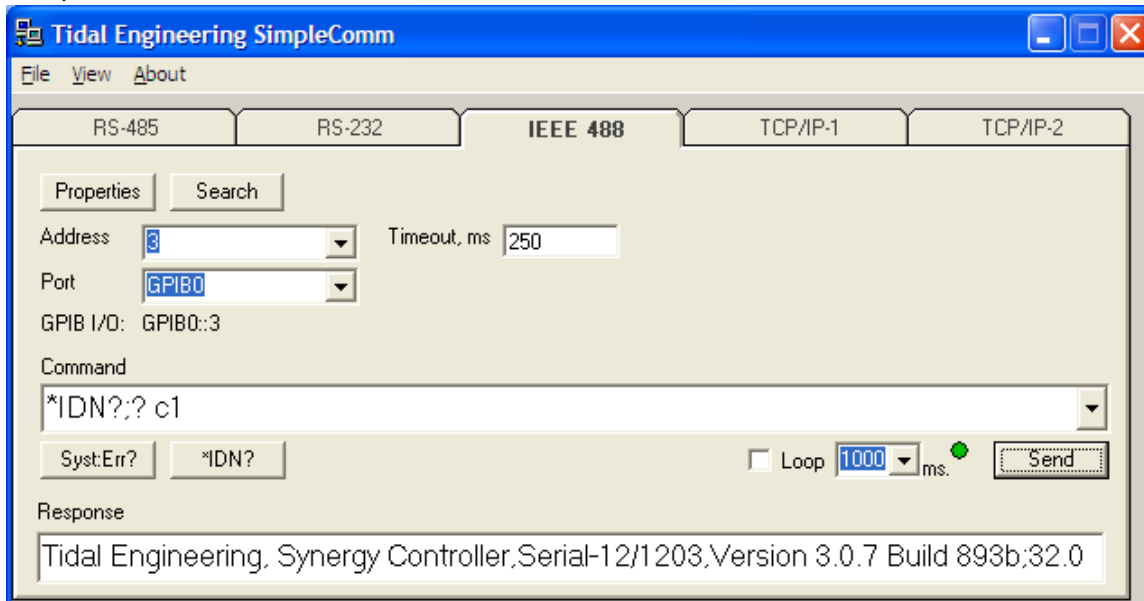
SimpleComm Setup

Download SimpleComm here: <http://www.tidaleng.com/downloads/SimpleCommV4.5.3Setup.zip>.

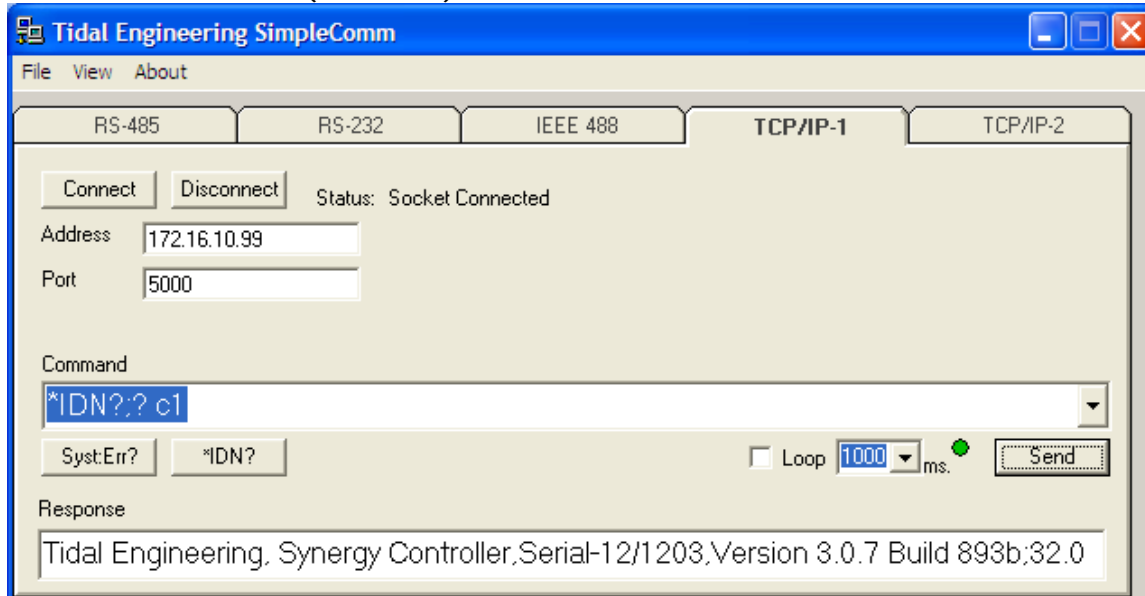
RS-232 Connection



GPIO/IEEE 488 Connection



TCPI/IP Connection (Ethernet)



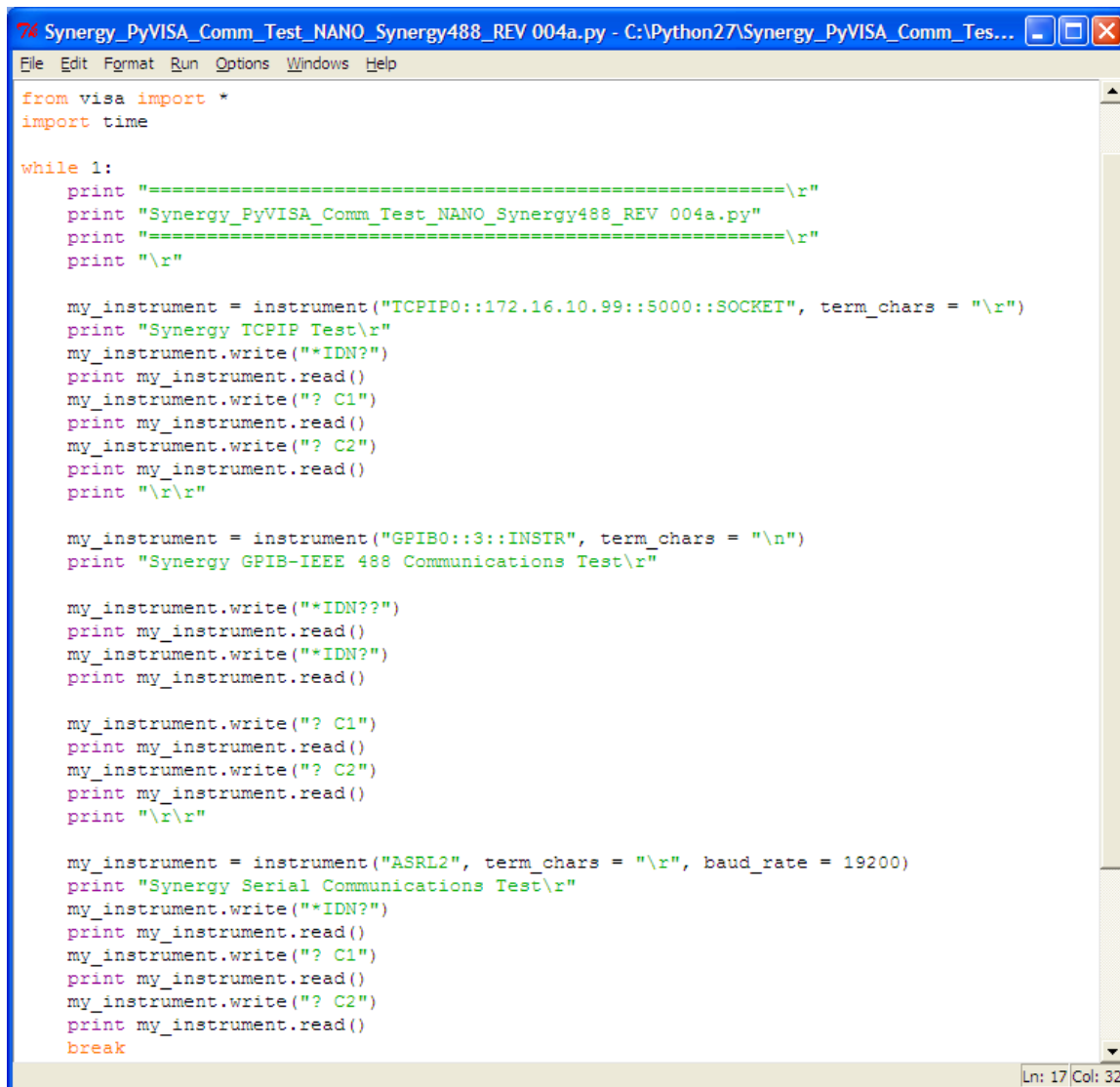
Python application

For Python and Python VISA setup help, see the following Synergy Application note:

Synergy Controller AppNote 107 - Synergy Controller Programming with Python

<http://www.tidaleng.com/appnotes/SCAP107.pdf>

Python Application



```
74 Synergy_PyVISA_Comm_Test_NANO_Synergy488_REV 004a.py - C:\Python27\Synergy_PyVISA_Comm_Tes...
File Edit Format Run Options Windows Help

from visa import *
import time

while 1:
    print "=====\r"
    print "Synergy_PyVISA_Comm_Test_NANO_Synergy488_REV 004a.py"
    print "=====\r"
    print "\r"

    my_instrument = instrument("TCPIP0::172.16.10.99::5000::SOCKET", term_chars = "\r")
    print "Synergy TCPIP Test\r"
    my_instrument.write("*IDN?")
    print my_instrument.read()
    my_instrument.write("? C1")
    print my_instrument.read()
    my_instrument.write("? C2")
    print my_instrument.read()
    print "\r\r"

    my_instrument = instrument("GPIB0::3::INSTR", term_chars = "\n")
    print "Synergy GPIB-IEEE 488 Communications Test\r"

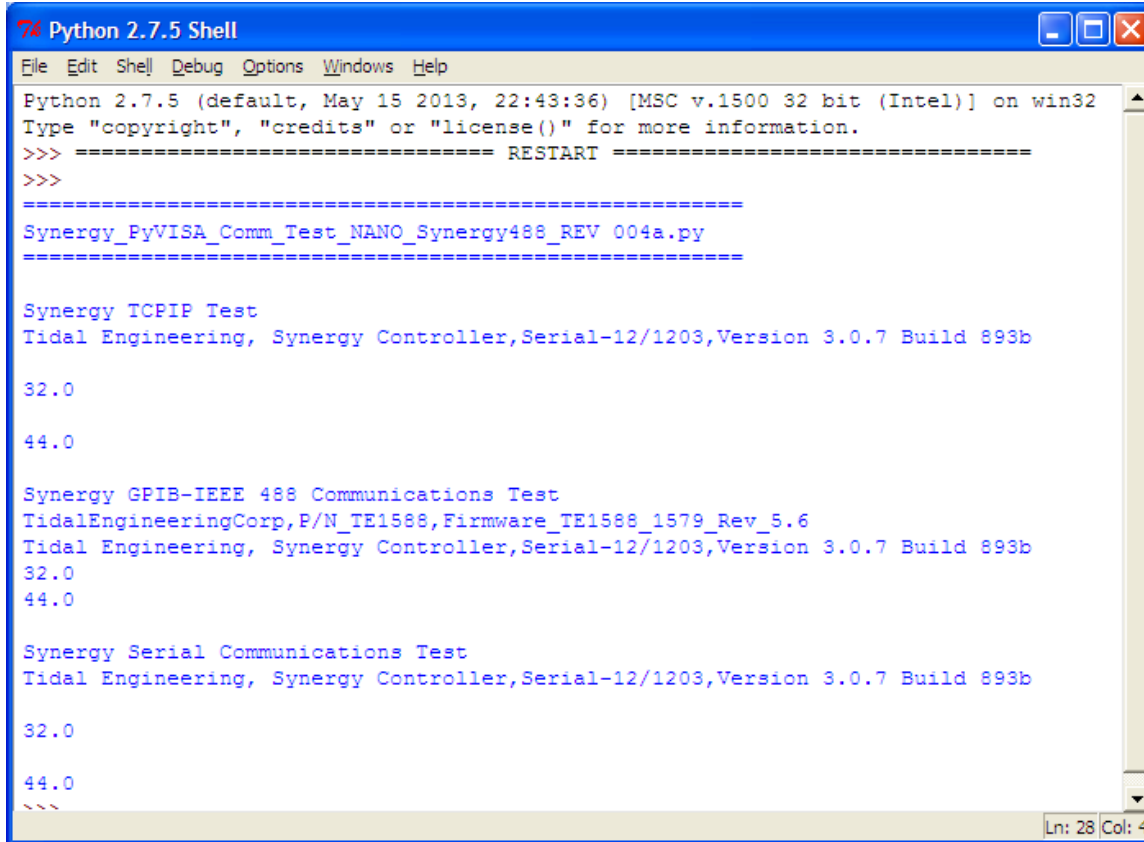
    my_instrument.write("*IDN??")
    print my_instrument.read()
    my_instrument.write("*IDN?")
    print my_instrument.read()

    my_instrument.write("? C1")
    print my_instrument.read()
    my_instrument.write("? C2")
    print my_instrument.read()
    print "\r\r"

    my_instrument = instrument("ASRL2", term_chars = "\r", baud_rate = 19200)
    print "Synergy Serial Communications Test\r"
    my_instrument.write("*IDN?")
    print my_instrument.read()
    my_instrument.write("? C1")
    print my_instrument.read()
    my_instrument.write("? C2")
    print my_instrument.read()
    break

Ln: 17 Col: 32
```

Python Output



```
Python 2.7.5 Shell
File Edit Shell Debug Options Windows Help
Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
=====
Synergy_PyVISA_Comm_Test_NANO_Synergy488_REV_004a.py
=====

Synergy TCP/IP Test
Tidal Engineering, Synergy Controller, Serial-12/1203, Version 3.0.7 Build 893b

32.0

44.0

Synergy GPIB-IEEE 488 Communications Test
TidalEngineeringCorp, P/N_TE1588, Firmware_TE1588_1579_Rev_5.6
Tidal Engineering, Synergy Controller, Serial-12/1203, Version 3.0.7 Build 893b
32.0
44.0

Synergy Serial Communications Test
Tidal Engineering, Synergy Controller, Serial-12/1203, Version 3.0.7 Build 893b

32.0

44.0
>>>
Ln: 28 Col: 4
```


Python Source Code

```
# Synergy_PyVISA_Comm_Test_NANO_Synergy488_REV 004.py
# Python VISA application tests Synergy NANO TCP/IP, GPIB, and Serial communications

from visa import *
import time

while 1:
    print "=====\r"
    print "Synergy_PyVISA_Comm_Test_NANO_Synergy488_REV 004a.py"
    print "=====\r"
    print "\r"

    my_instrument = instrument("TCPIP0::172.16.10.99::5000::SOCKET", term_chars = "\r")
    print "Synergy TCPIP Test\r"
    my_instrument.write("*IDN?")
    print my_instrument.read()
    my_instrument.write("? C1")
    print my_instrument.read()
    my_instrument.write("? C2")
    print my_instrument.read()
    print "\r\r"

    my_instrument = instrument("GPIB0::3::INSTR", term_chars = "\n")
    print "Synergy GPIB-IEEE 488 Communications Test\r"

    my_instrument.write("*IDN??")
    print my_instrument.read()
    my_instrument.write("*IDN?")
    print my_instrument.read()

    my_instrument.write("? C1")
    print my_instrument.read()
    my_instrument.write("? C2")
    print my_instrument.read()
    print "\r\r"

    my_instrument = instrument("ASRL2", term_chars = "\r", baud_rate = 19200)
    print "Synergy Serial Communications Test\r"
    my_instrument.write("*IDN?")
    print my_instrument.read()
    my_instrument.write("? C1")
    print my_instrument.read()
    my_instrument.write("? C2")
    print my_instrument.read()
    break
```

About the Synergy Controller Family

Tidal Engineering's Synergy Controllers, both the Synergy Micro 2 and the ¼ DIN Synergy Nano provide state-of-the-art usability and connectivity for environmental test control and data acquisition. They combine the functions of a chamber controller and a data logger and are designed to improve test efficiency by supporting both factory automation and test and measurement protocols and standards.

Synergy Controller feature highlights includes:

- ➔ Color touch screen
- ➔ Ethernet, RS-232 and GPIB communications
- ➔ Built in 100 MB Data logger with USB drive support
- ➔ Data Acquisition, up to 64 T-type thermocouples (Optional)
- ➔ Built-in Web Server for remote control; WebTouch Remote™
- ➔ Compatible with Synergy Manager for PC based control, monitoring and programming.
- ➔ Built-in FTP Server for factory automation and test and measurement applications

For more information regarding these controllers please see the full Synergy Controller Technical Manual on our website at <http://www.tidaleng.com/synergy.htm>

About Tidal Engineering

Headquartered in Randolph, NJ, Tidal Engineering Corporation has been designing and building award-winning embedded hardware and software for test and measurement and data acquisition applications since 1992. The company is recognized for technical expertise in such areas as Embedded IEEE 488, and turnkey SCADA (Supervisory Control and Data Acquisition) systems.

Tidal Engineering Corporation
2 Emery Avenue
Randolph, NJ 07869
Tel: 973/328-1173
Fax: 973/328-2302
www.TidalEng.com
info@tidaleng.com

