



## Sigma Gateway

Using Python for load bank control

Document Number: FP1012-007 – rev 1.22

Revision Date: Nov 2023

## 1. Introduction

The Sigma Gateway or Sigma Wireless Gateway Modbus TCP interface provides a means of integrating the Sigma 2 and LT load banks with generator test cell automation or building supervisory and monitoring systems.

This document describes the steps involved installing and using a Python script to control a load bank connected to a Sigma Gateway.

## 2. Requirements

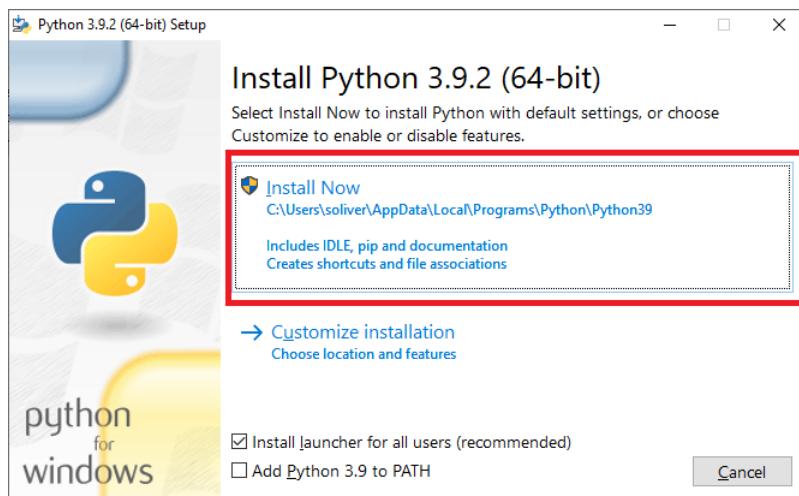
The following are required:

- Sigma Gateway (PR02899) or Sigma Wireless Gateway (PR03074).
- Python 3 [installer](#).
- Sigma LT or Sigma 2 load bank.
- Sigma Gateway Modbus Register Manual (FP1012-002).
- Sigma Gateway Python example [script](#) 'gateway\_modbus.py'.

## 3. Python Installation

Download Python and then launch the installation. Select "Install Now" as highlighted below.

Python version 3.9.2 was used in this document.



Once Python has been installed, any required modules will also need installing.

Launch a ‘command prompt’ using the start menu or using shortcut “Win+R” then type cmd. Enter the following cmd to install the python module ‘pyModbusTCP’.

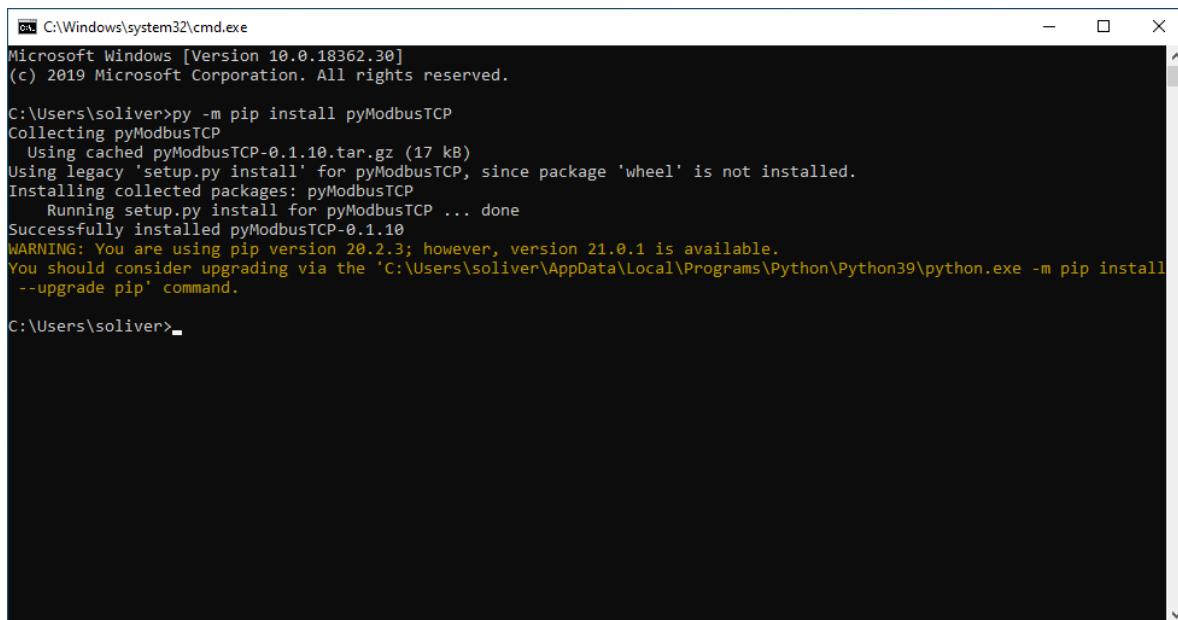
‘pip’ is the preferred method for installing Python modules, the cmd used depends on the OS being used.

On Windows the cmd would be:

```
py -m pip install pyModbusTCP
```

For other operating system use the following cmd:

```
python -m pip install pyModbusTCP
```



The screenshot shows a Windows Command Prompt window titled 'cmd C:\Windows\system32\cmd.exe'. The window displays the following text output from the pip command:

```
Microsoft Windows [Version 10.0.18362.30]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\soliver>py -m pip install pyModbusTCP
Collecting pyModbusTCP
  Using cached pyModbusTCP-0.1.10.tar.gz (17 kB)
Using legacy 'setup.py install' for pyModbusTCP, since package 'wheel' is not installed.
Installing collected packages: pyModbusTCP
  Running setup.py install for pyModbusTCP ... done
Successfully installed pyModbusTCP-0.1.10
WARNING: You are using pip version 20.2.3; however, version 21.0.1 is available.
You should consider upgrading via the 'C:\Users\soliver\AppData\Local\Programs\Python\Python39\python.exe -m pip install --upgrade pip' command.

C:\Users\soliver>
```

‘pip’ requires a valid internet connection to download and install the required Python modules. If you are behind a corporate firewall you may need to configure the proxy used if not properly detected.

This is an example when using ZScaler Cloud Security:

```
python -m pip install pyModbusTCP --proxy https://127.0.0.1:9000
```

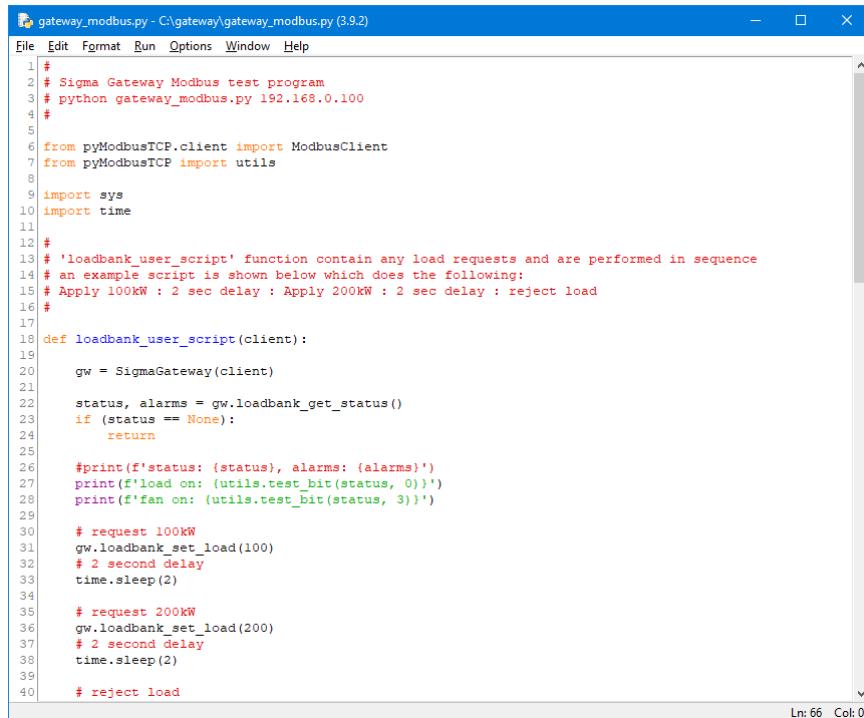
[Python](#) and [pyModbusTCP](#) are now installed and ready to use with the Sigma Gateway.

It is possible to use other Python Modbus modules such as [PyModbus](#).

## 4. Python Example Script

The example script 'gateway\_modbus.py' can be used as a starting point to understand how Python can be used to control a Sigma based load bank.

Open the file by right clicking on the 'gateway\_modbus.py' script and select "Edit with IDLE". IDLE is the default editor installed with Python, however others such as [Notepad++](#) can also be used.



A screenshot of the Python IDLE editor showing the 'gateway\_modbus.py' script. The window title is 'gateway\_modbus.py - C:\gateway\gateway\_modbus.py (3.9.2)'. The code is as follows:

```
1 #
2 # Sigma Gateway Modbus test program
3 # python gateway_modbus.py 192.168.0.100
4 #
5
6 from pyModbusTCP.client import ModbusClient
7 from pyModbusTCP import utils
8
9 import sys
10 import time
11
12 #
13 # 'loadbank_user_script' function contain any load requests and are performed in sequence
14 # an example script is shown below which does the following:
15 # Apply 100kW : 2 sec delay : Apply 200kW : 2 sec delay : reject load
16 #
17
18 def loadbank_user_script(client):
19
20     gw = SigmaGateway(client)
21
22     status, alarms = gw.loadbank_get_status()
23     if (status == None):
24         return
25
26     #print(f'status: {status}, alarms: {alarms}')
27     print(f'load on: {(utils.test_bit(status, 0))}')
28     print(f'fan on: {(utils.test_bit(status, 3))}')
29
30     # request 100W
31     gw.loadbank_set_load(100)
32     # 2 second delay
33     time.sleep(2)
34
35     # request 200kW
36     gw.loadbank_set_load(200)
37     # 2 second delay
38     time.sleep(2)
39
40     # reject load
```

Ln: 66 Col: 0

The script has been written so that the user only has to edit the function 'loadbank\_user\_script' to perform a simple load bank test.

The contents of this function can be seen below, please edit this to obtain the required results.

```
def loadbank_user_script(client):  
  
    gw = SigmaGateway(client)  
  
    status, alarms = gw.loadbank_get_status()  
    if (status == None):  
        return  
  
    #print(f'status: {status}, alarms: {alarms}')  
    print(f'load on: {utils.test_bit(status, 0)}')  
    print(f'fan on: {utils.test_bit(status, 3)}')  
  
    # request 100kW  
    gw.loadbank_set_load(100)  
    # 2 second delay  
    time.sleep(2)  
  
    # request 200kW  
    gw.loadbank_set_load(200)  
    # 2 second delay  
    time.sleep(2)  
  
    # reject load  
    gw.loadbank_set_load(0)  
  
# end of user script
```

This script can be launched directly in IDLE by selecting “Run\Run Module” menu item or pressing F5. It is also possible to launch the script directly in Windows Explorer.

By default the script will use a Gateway at the default IP address of 169.254.1.1. If this is not correct, then pass the correct IP address using an argument to the script, eg.

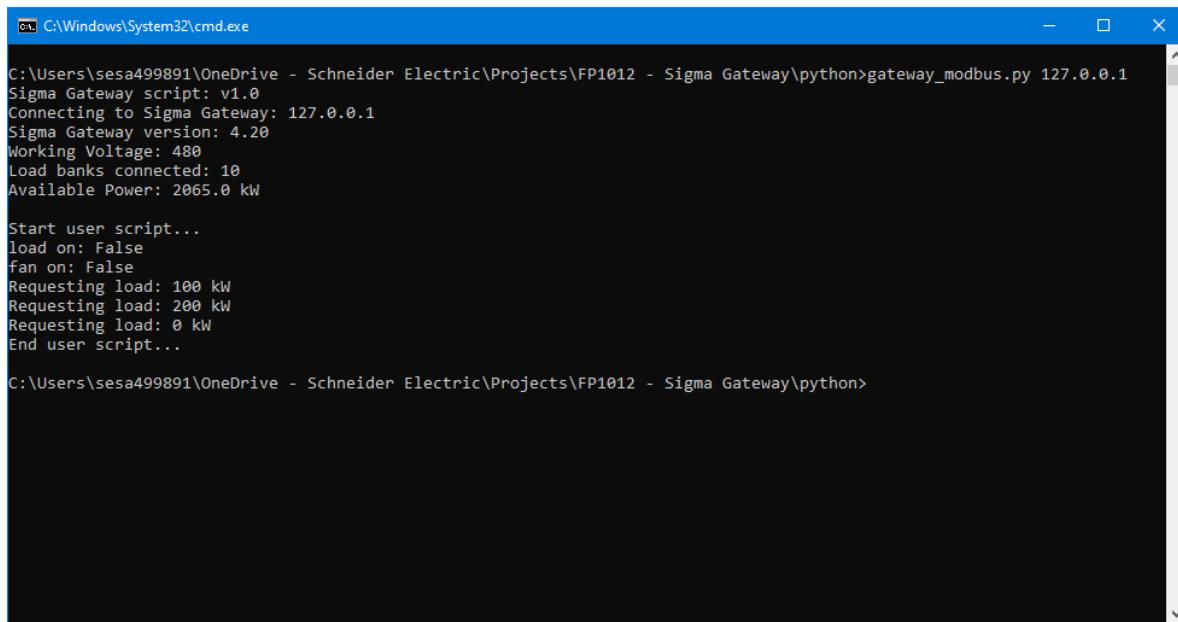
```
gateway_modbus.py 192.168.0.20
```

This IP address can also be changed in IDLE by using “Run\Run Customised”, or alternately using Shift-F5.

The output from this script can be seen below, however it simply does the following:

- Query Gateway for Version Info, load bank count and available capacity (kW)
- Apply 100kW and wait 2 seconds.
- Apply 200kW and wait 2 seconds.
- Reject load.
- Exit script

By default load will be shared proportionally (relative to load bank capacity) across all connected load banks. Individual load bank control is also possible but beyond the scope of this example.



A screenshot of a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The window displays the output of a Python script named 'gateway\_modbus.py' running on a local host (127.0.0.1). The script connects to a Sigma Gateway at version 4.20, which is operating at 480V and has 10 load banks connected. It shows the start of a user script, setting fan and load parameters, and then ending the script. The command prompt then returns to the directory 'C:\Users\sesa499891\OneDrive - Schneider Electric\Projects\FP1012 - Sigma Gateway\python>'.

```
C:\Users\sesa499891\OneDrive - Schneider Electric\Projects\FP1012 - Sigma Gateway\python>gateway_modbus.py 127.0.0.1
Sigma Gateway script: v1.0
Connecting to Sigma Gateway: 127.0.0.1
Sigma Gateway version: 4.20
Working Voltage: 480
Load banks connected: 10
Available Power: 2065.0 kW

Start user script...
load on: False
fan on: False
Requesting load: 100 kW
Requesting load: 200 kW
Requesting load: 0 kW
End user script...

C:\Users\sesa499891\OneDrive - Schneider Electric\Projects\FP1012 - Sigma Gateway\python>
```

The documentation for Python is installed locally but can also be accessed [online](#). Documentation for pyModbusTCP module documentation can be found [here](#).

[www.avtronpower.com](http://www.avtronpower.com)

## Avtron Power Solutions

6255 Halle Drive  
Cleveland, OH 44125, USA  
Tel: 216 573 7600  
LBsales@ascopower.com

Cliffe Road, Easton-on-the-Hill  
Stamford PE9 3NP, United Kingdom  
Tel: +44 1780 480033  
LBsalesint@ascopower.com