Terminus T2
Python Quick Start Guide

JANUS REMOTE COMMUNICATIONS

Bulletin	**JA15-PUM**
Revision	**A00**
Date	**13 Aug 2013**

TABLE OF CONTENTS

DISCLAIMER

Guide Overview

This guide is designed to help streamline the user's embedded python experience with the Terminus 2 platform. The topics discussed will be an overview of the system and what to expect, what steps to take to upload and run a script, and a demo script run-through. The script is developed for the HE910 Telit modem, and will not function properly on the GE865 Telit modem.

This guide assumes that Python 2.7.2 has been installed, if you don't have it installed please visit:

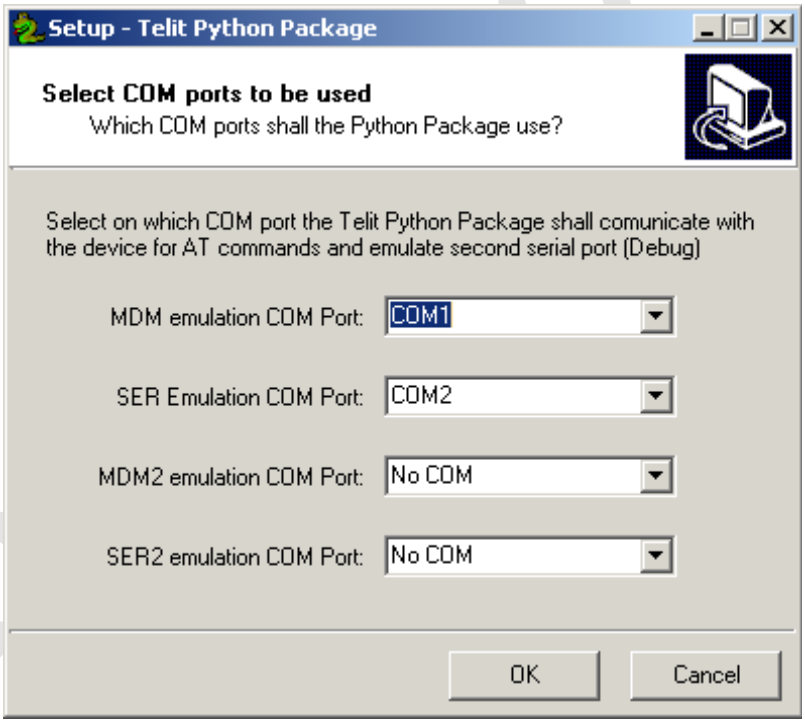
http://www.python.org/download/releases/2.7.2/

Optional install #1: Python 1.5.2+ v4.1

There are a few ways to upload the scripts to the modem, but one very useful way is the Right Click -> Download method. This is an older method available from a previous version of Python.

This version of Python should be utilized if developing scripts for the GE865 Telit modem.

The only required setup is to configure the COM ports with the COM Port Selection Tool as shown.



Note that Python 2.7.2 will still be used to develop scripts for the Telit HE910, the older version of Python is to allow access to the Right Click -> Download method for loading scripts and also for developing scripts to use with the Telit GE865.

Optional install #2: RSTerm

RSTerm is a multi-function tool from www.thebyteworks.com that can be used to communicate via AT commands to the modem, along with several pre-built functions available via button clicks.

However, the tool was designed around older GSM terminals, and uses legacy commands for many of the pre-built functions. Janus recommends not using these functions. Instead, we recommend installation because of its useful ability to mass transfer python scripts and files.

References

Janus T2 User Manual

Telit AT Command Reference

Telit HE910 Easy Script in Python

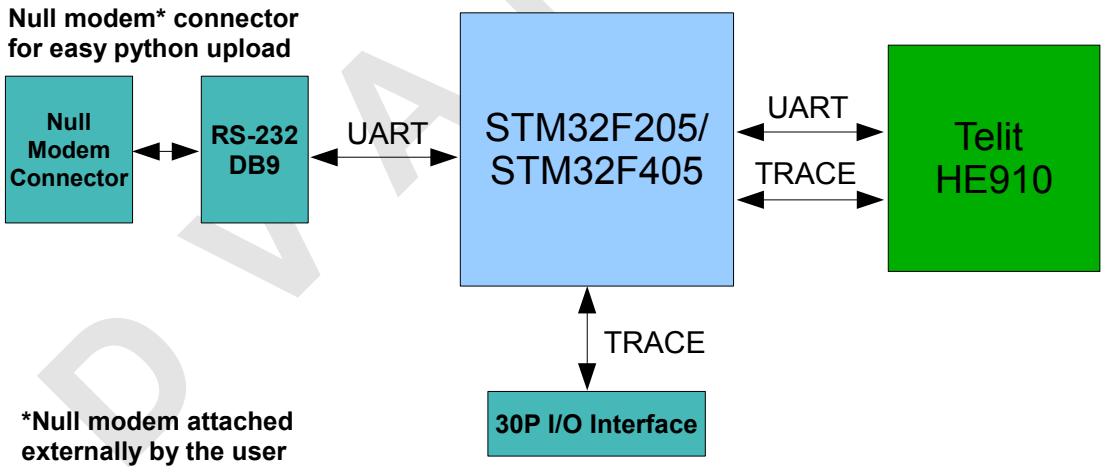
Telit HE910 Family Ports Arrangements

Applicable Products

HSPA910T2

System Overview

The T2 is a powerful platform that utilizes an STM32F205 or STM32F405 to control and use different communication interfaces and I/O. Among the communication interfaces is an embedded Telit modem that has its own python script interpreter. The user has the ability to use C compilers for the ST Micro directly, or run python scripts on the modem. The user has the ability to use both, but this guide will focus on just using the embedded Python.



Before Loading Your Python Scripts

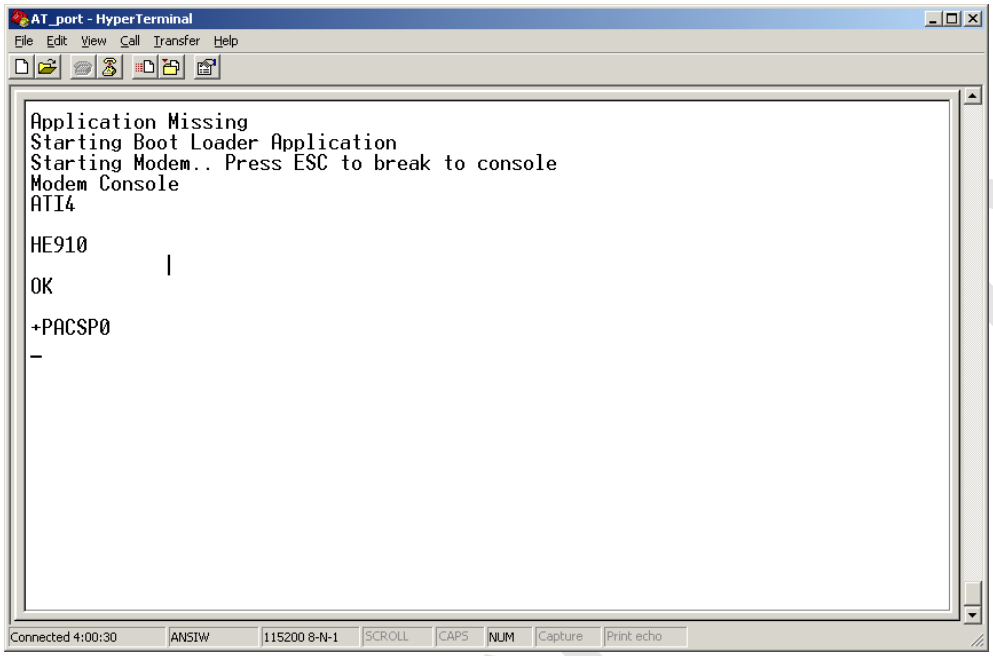
Open HyperTerminal or a similar terminal emulator program and create a connection to the COM port attached to the T2 DB9. Use the following settings for the connection:

- Baud Rate: 115200bps
- Bits: 8
- Stop Bits: 1
- Parity: None
- Hardware Handshaking: Yes

By default both the Telit modem and the T2 utilize 115200bps.

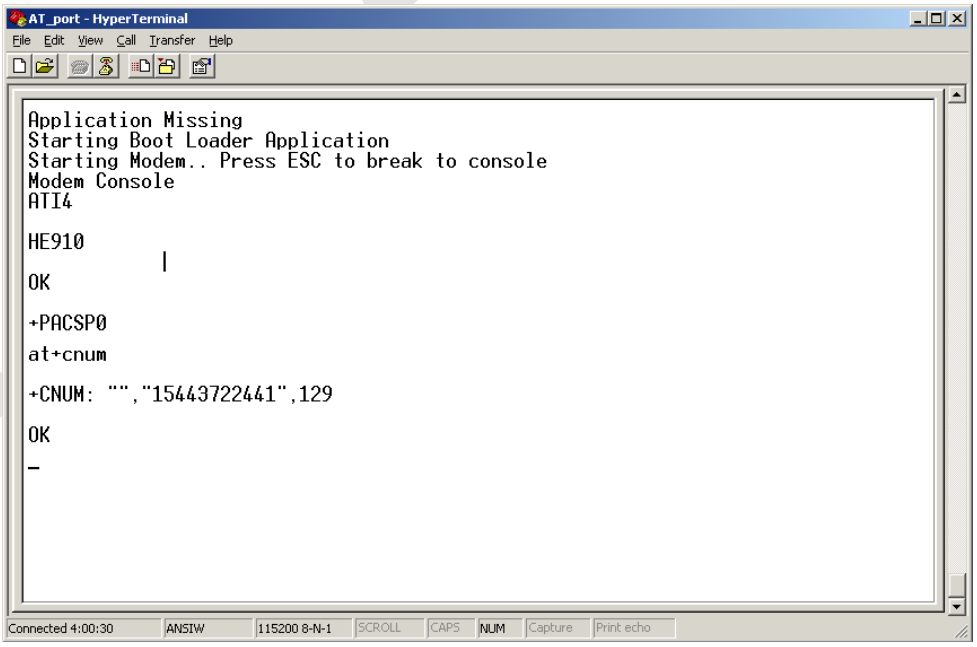
Before Loading Your Python Scripts continued

The ST Micro's bootloader will, on power up, search for a loaded application. If one is not found it will continue and allow the user to access the Telit modem directly via AT commands. When powering up you will see something similar to the following example.



No application was found, so the modem was then forwarded and we are ready to continue.

For this demo you will need the phone number associated with the SIM. If you are unsure of what this is you can query the modem with "AT+CNUM".



Close the connection and continue.

HE910 SMS Response Demo

Overview

The SMS Response demo is designed to give the user a fast way to bring the unit up for evaluation while also covering example code for various functions. The demo features the following for usage/evaluation:

- Automatic checks and settings for network registration and general operation
- File system reading
- Remote control and status checks via SMS

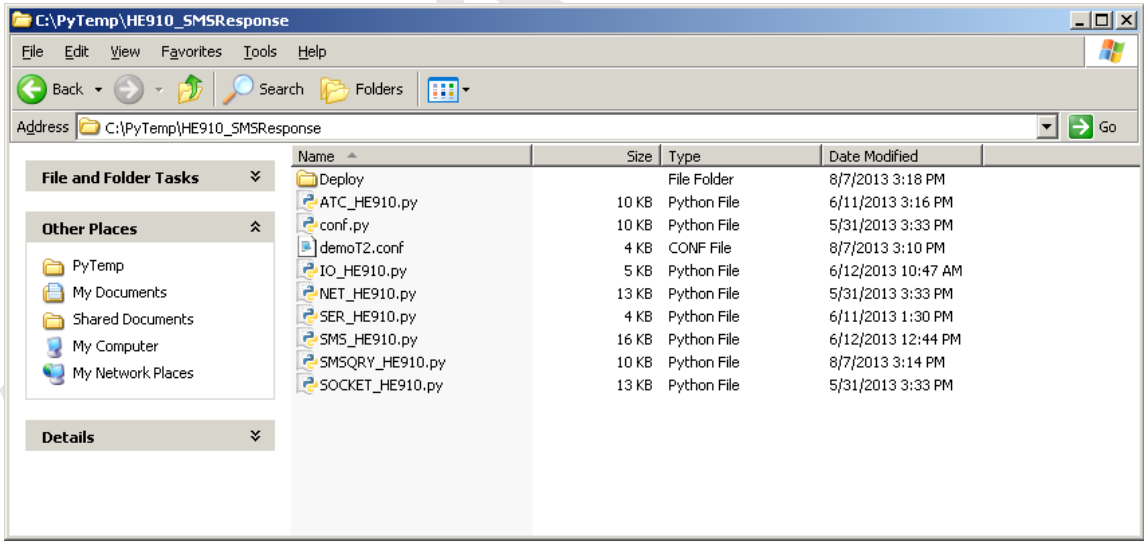
Demo Package Contents

HE910_SMSResponse.zip

- Python Scripts
  - ATC_HE910.py
  - conf.py
  - IO_HE910.py
  - NET_HE910.py
  - SER_HE910.py
  - SMS_HE910.py
  - SMSQRY_HE910.py
  - SOCKET_HE910.py
- Other
  - demoT2.conf

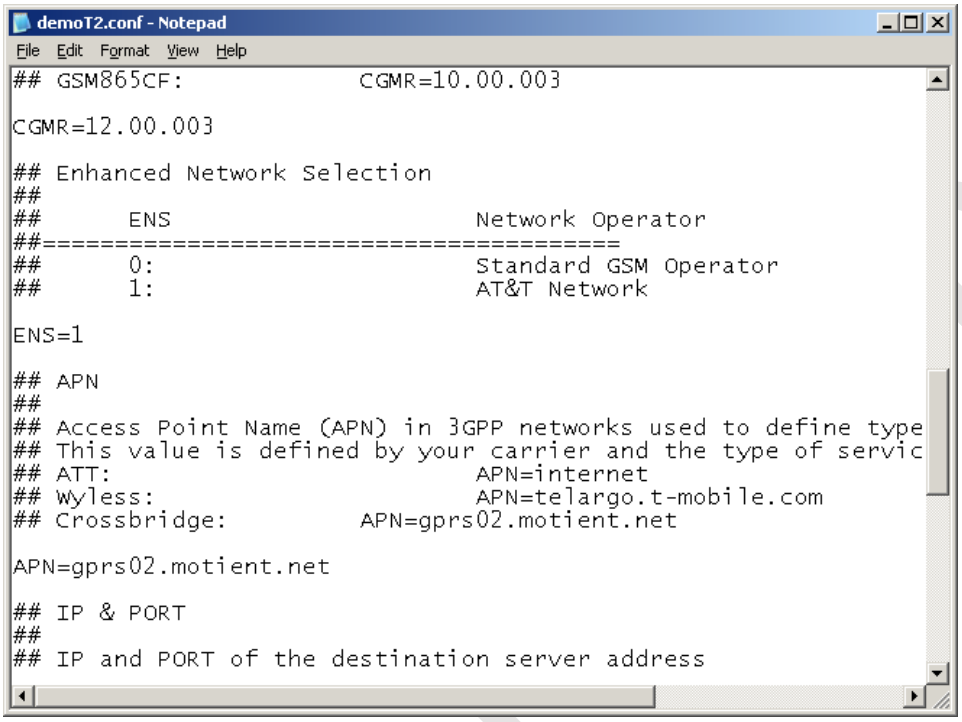
Compiling the Python Scripts

First extract the python scripts from the HE910_SMSResponse archive



Compiling the Python Scripts continued

Now open the demoT2.conf file to edit the options. For this demo, the only option you need to pay attention to is ENS. Please refer to the AT command guide if you are unsure what AT#ENS setting needs to be utilized. If you are on AT&T, you can leave it set to 1.

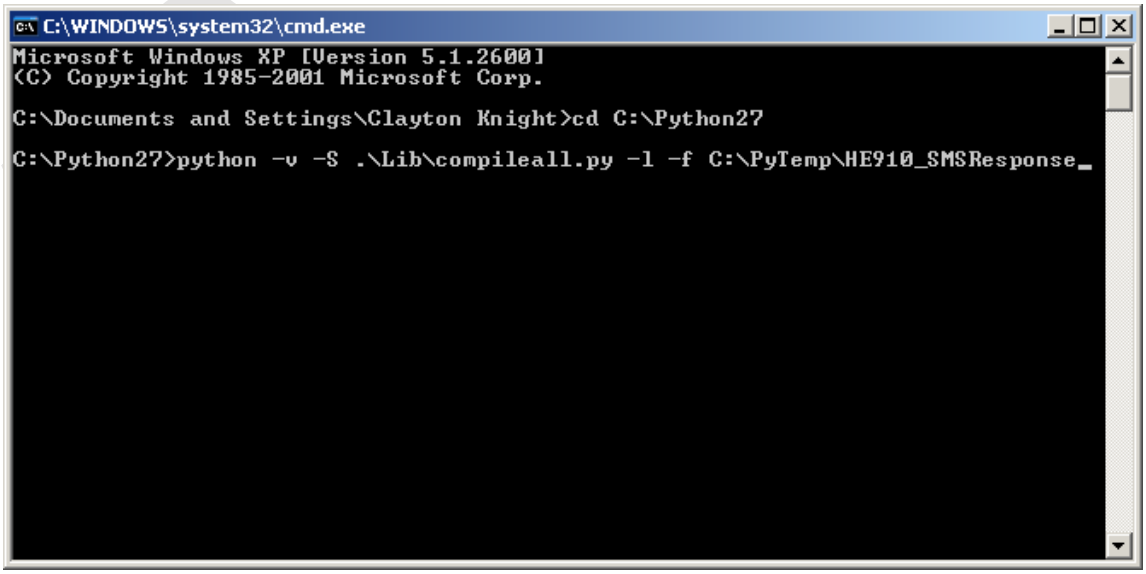


Save and exit the file.

Use the compileall.py library Python script on your PC to compile all .py files in your working directory (as an example in directory C:\pytemp)

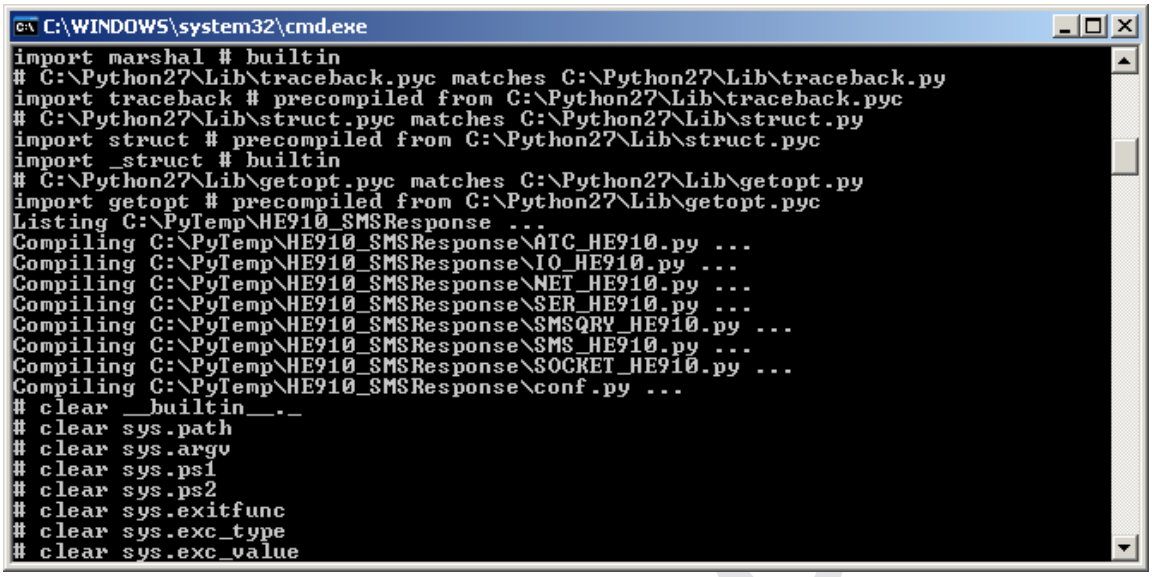
cd C:\Python27

python -v -S .\Lib\compileall.py -l -f C:\pytemp

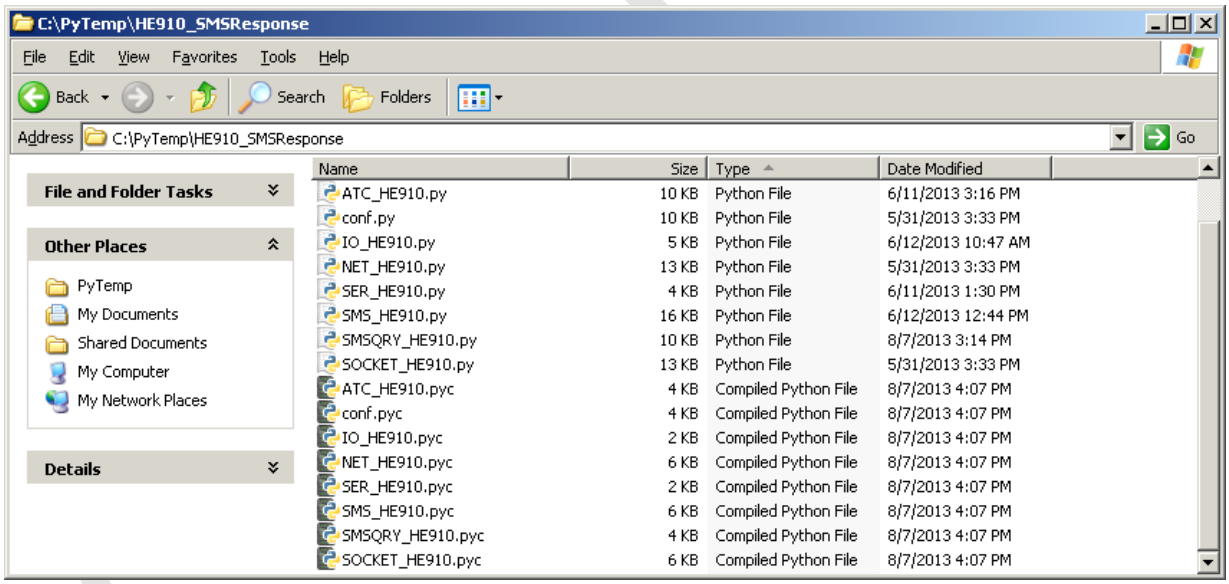


Compiling the Python Scripts continued

Once you press return a lot of information gets put into the command window, but the important information can be found in the middle:



As long as there are no errors reported, the files compiled OK. There are other ways to compile scripts, such as doing so in the IDE, but the important thing is that they are compiled into .pyc format!



Move the compiled scripts into the “Deploy” folder, and continue.

Uploading the Python Scripts

Method #1: AT#WSCRIPT

Use the following AT command:

AT#WSCRIPT="<script_name>",<size>

Where:

<script_name>: file name

<size>: file size (number of bytes)

The script, the compiled script, any text or binary file, can be downloaded to the module using the AT#WSCRIPT command. In order to download the, optionally compiled, Python script you have to choose a name for your script on the module, taking care of the following:

- The extension for scripts is .py.
- The extension for compiled scripts is .pyc.
- Any or no extension is permitted for generic text or binary file.
- The maximum file name length allowed is 16 characters.
- File names are case sensitive.

Then you have to find out the exact size in bytes of the script or compiled script, or generic text or binary file. For example, right clicking on the file and selecting "size" in "properties" (attention: this is different from selecting "size on the disc").

It is important for large files, compared to module serial port buffer size of 4096 bytes, to activate hardware flow control on your terminal emulator.

It is possible to overwrite an existing file, there is no need to delete old one first.

For the configuration file we do the following:

AT#WSCRIPT=" demoT2.conf",3422

wait for the prompt

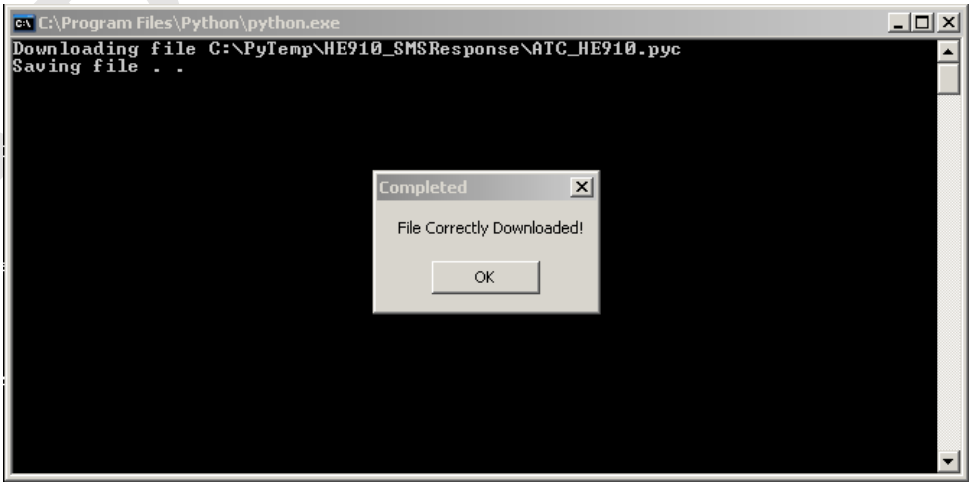
>>>

and use "Send Text file" selecting the proper file.

Wait for the result: OK or ERROR.

Method #2: Right Click -> Download

For .py and .pyc files we can use the Right Click -> Download method as they are recognized python scripts. Simply right click on the .py or .pyc file and select "Download". You should see the following for a successful transfer.



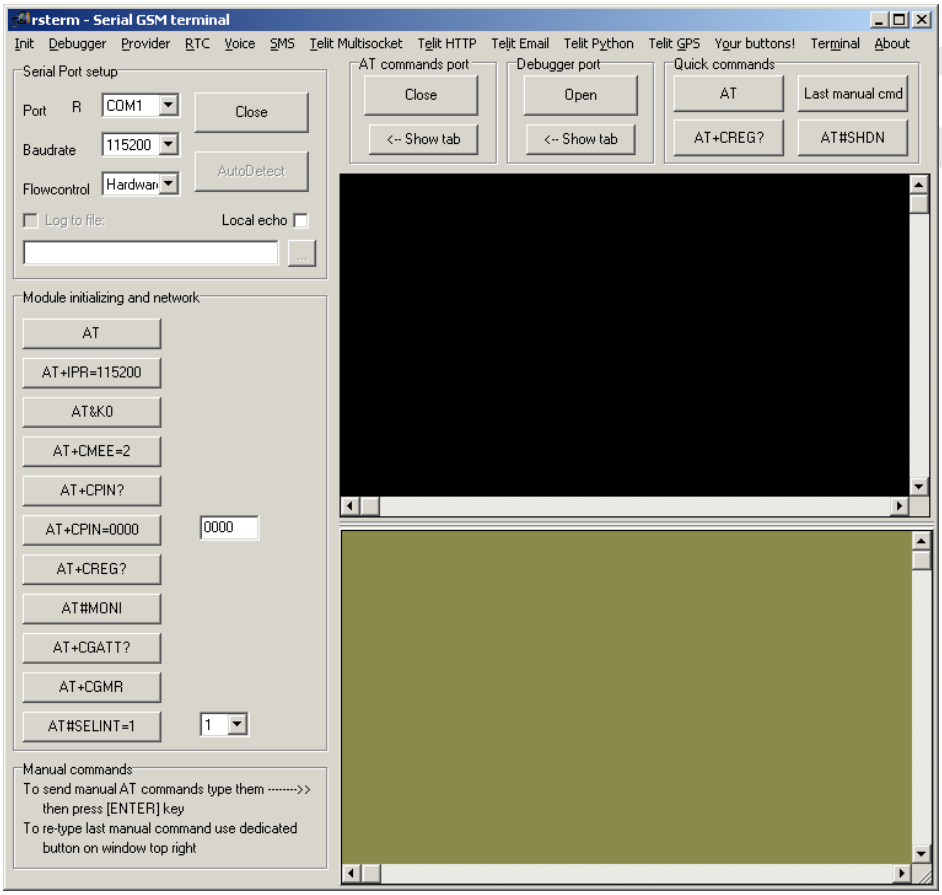
Once you have loaded all of the files to the modem, open the Hyperterminal session again, do an AT#LSCRIPT to verify all have been added. Once verified, select the main script with AT#ESCRIPT.

Uploading the Python Scripts continued

Method #3: RSTerm

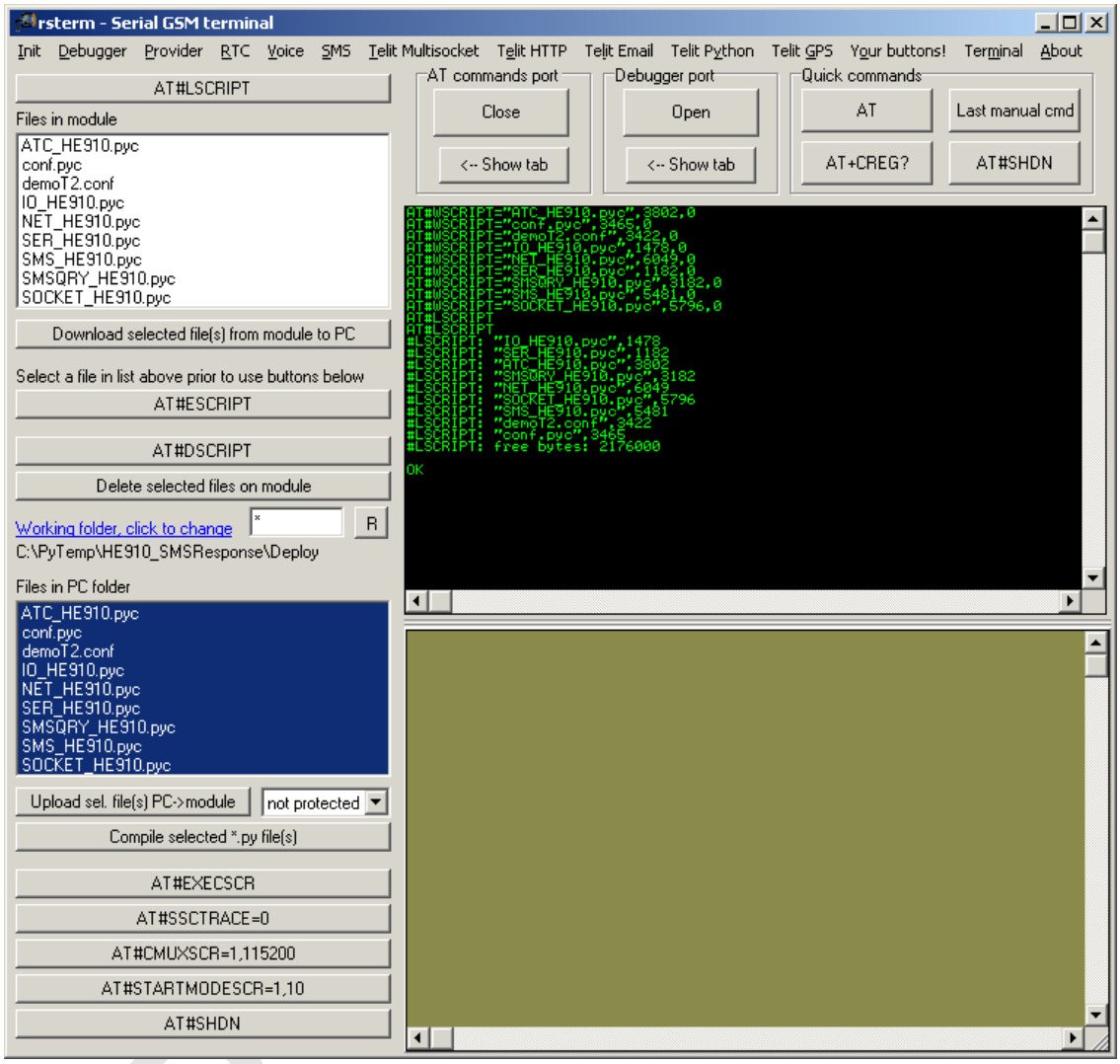
We can use RSTerm to remove a lot of time and hassle when managing scripts and files for the modem. This is the fastest way to move your scripts to the modem when there are multiple, including non python script files.

First, open RSTerm using the intended COM port, select Hardware flow control, and 115200 baud rate, click “Open” to open the connection.



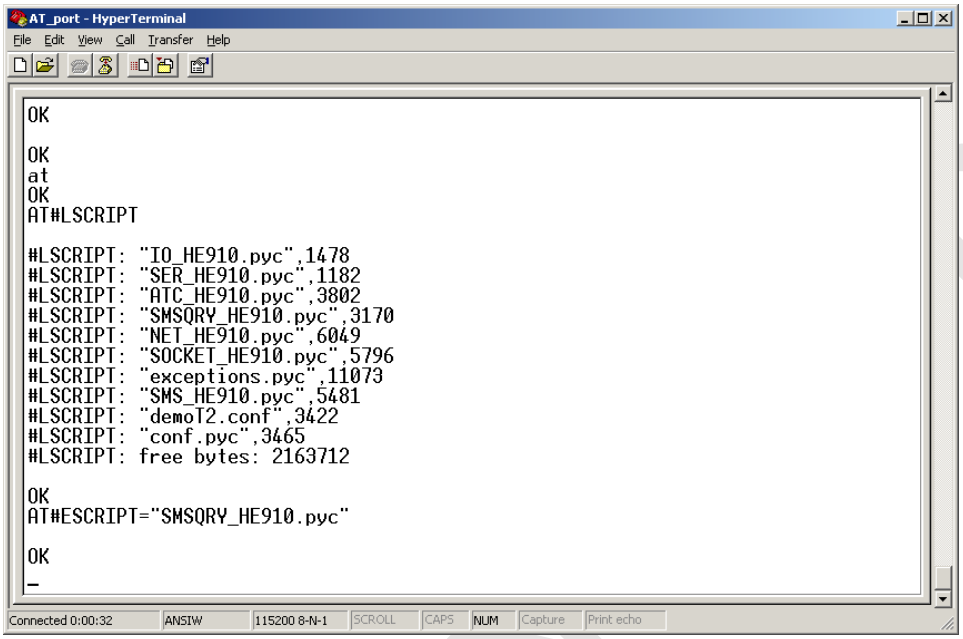
Uploading the Python Scripts continued

Go to the “Telit Python” tab, change your working directory to where the Deploy folder is. Highlight all of the files, and then select “Upload sel. file(s) PC->Module”.



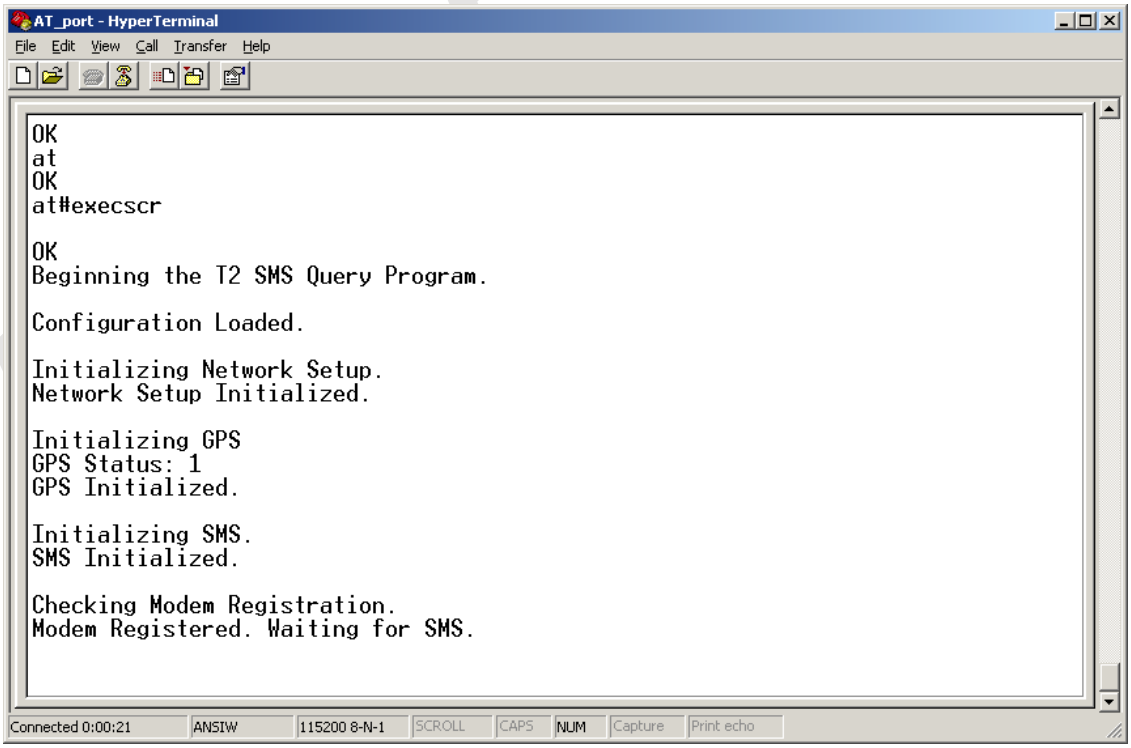
Uploading the Python Scripts continued

RSTerm will then use the AT#WSCRIPT command to mass upload the files to the modem, and then use AT#LSCRIPT to list them back for you to verify they are there. The current files on the modem will appear in the top left window as well for selection if you need to download, delete, or even select which file will be labeled with AT#ESCRIPT.



Select the main script by entering AT#ESCRIPT=”SMSQRY_HE910.pyc”.

Now you can run the script with AT#EXECSCR and the script will begin. You should see the following if everything has run successfully.



Uploading the Python Scripts continued

The Demo supports receiving SMS commands to do 3 possible things:

1. Status query for the current reporting information and location
2. On the fly adjustment of how the unit will behave
3. Remote AT commands

The status query can be acquired by sending: STATUS

The demo will send an SMS back to the originating number with the following information:

- Unit: 111111111111111
- Status LED: ON
- User LED: OFF
- Current Location: $GPRMC,000000.000,V,,,,,,,000000,,,N*4D

To change a function, send “CMD: x = y” (no quotation marks) where x is the command to change and y is the value. Below are the settings which are capable of being changed by this and what values the demo can accept:

- SLED
  - ON
  - OFF
- ULED
  - ON
  - OFF

For example, to change the status LED to OFF you would send:

CMD: SLED = OFF

The script will the save the setting and respond to the originating phone number with an echo:

Auto ON : OFF

The demo can also accept and carry out standard AT commands. Once complete the unit will send back the response of the command.

For example, sending the basic “AT<cr>” command you would send:

AT (No carriage return)

The script will the save the setting and respond to the originating phone number with an echo:

OK

Terminus T2
Python Quick Start Guide

Revision History

Revision	Revision Date	Note
A00	08/13/13	Advanced Release - Python Quick Start Guide

JANUS REMOTE COMMUNICATIONS

Division of The Connor-Winfield Corporation
2111 Comprehensive Drive • Aurora, Illinois 60505
630.499.2121 • Fax: 630.851.5040
www.janus-rc.com

Janus Remote Communications Europe
Bay 143
Shannon Industrial Estate
Shannon, Co. Clare, Ireland
Phone: +353 61 475 666