

telitpython: Run_Python_scripts

Please, refer to the last version of Easy Script in Python user guide.

However:

The steps required to have a script running by the python engine of the module are:

- write the python script;
- compile the python script;
- download the python script into the module NVM;
- enable the python script;
- execute the python script.

To enable the Python Script:

AT#ESCRIPT="< script_name >"

the script_name extension could be **pyo** or **py** : better if you download on the module pyo files only

For example:

AT#ESCRIPT="a.pyo"

Wait for enable result: OK.

NOTE: There is no error return value for non existing script name in the module memory typed in command **AT#ESCRIP**T. A trick to avoid mistakes is to use **AT#LSCRIPT** command to list the files in the module,

copy the string name of the file to enable and past it as argument of the **AT#ESCRIP**T= command .

To check to have enabled the righth file, issue:

AT#ESCRIPT?

- < script_name >: file name

Enabling the script means to select the Python script which will be executed (the enabled script) from the next start-up and in every future start-up. In case the Python script consists of more than one file only the main file should be executed.

To execute the Python Script

Generally: The real execution of the Python script is delayed from the power on due to the time needed by Python to parse the script. The longer is the script, the longer is this delay.

The real execution of compiled Python scripts (.pyo) is faster just opening and reading the file from NVM.

There are more ways to execute/run the enabled Python script.

1.

The Python script you have downloaded to module and enabled is executed at every module power on **IF** the DTR line is sensed LOW (2.8V at the module DTR pin - RS232 signals are inverted -) at startup, (in this case no AT command interface is connected to the modem port).

Note: Python script will be executed when the module is powered ON and if the serial cable has previously been disconnected.

2.

using **AT#STARTMODESCR** command:

AT#STARTMODESCR=0

script will be executed at startup only if the DTR line is found Low (e.g.: COM is not open on a PC) - factory default

AT#STARTMODESCR=1,<script_start_to>

Useful option if it's not possible to have the DTR low (with argument set to 1) ; the script will start to execute after <script_start_to> seconds at startup only if the user does not send any AT command on the serial port for the time interval specified in <script_start_to> parameter

AT#STARTMODESCR=2

Useful option to use the standard AT command interface on ASC0 and run a Python script

3.

Most useful option in some situations during the debug :

AT#EXECSCR

this command causes the current script (see #ESCRIP) execution not at startup. This command is useful when the execution at startup has been blocked deliberately and the user wants to control execution start. Very useful during the debug: it's not necessary to switch off and then on the module to run the script.

A good test script to know if the script is running or not is the following: [SERtest.py \(/f/SERtest.py\)](#)

If it's running, this script sends the "TEST" string on the ASC0 and the "1" string on the debug port after the initial parsing period