Reading sending data and automatic simulations from-to Matlab/Python

RTDS training course

2018

Intelligent Electrical Power Grids

Electric Sustainable Energy

TU Delft

TU Delft

Outlines

- Introduction
- Connecting with a MATLAB M file
- Connecting with Python 3

Introduction

From time to time, some RTDS users want is to be able to **send out small number of commands to the simulation from an external application such as MATLAB™ as well as retrieve the corresponding simulation results**. The incoming results from RTDS can be utilized in the external application. Usually the necessary interaction speed in such circumstance is not too high, i.e., in the range of hundreds of milliseconds.

- A script function in RSCAD/Runtime.
  - 'ListenOnPort()' command
- a TCP/IP socket communication: server + client

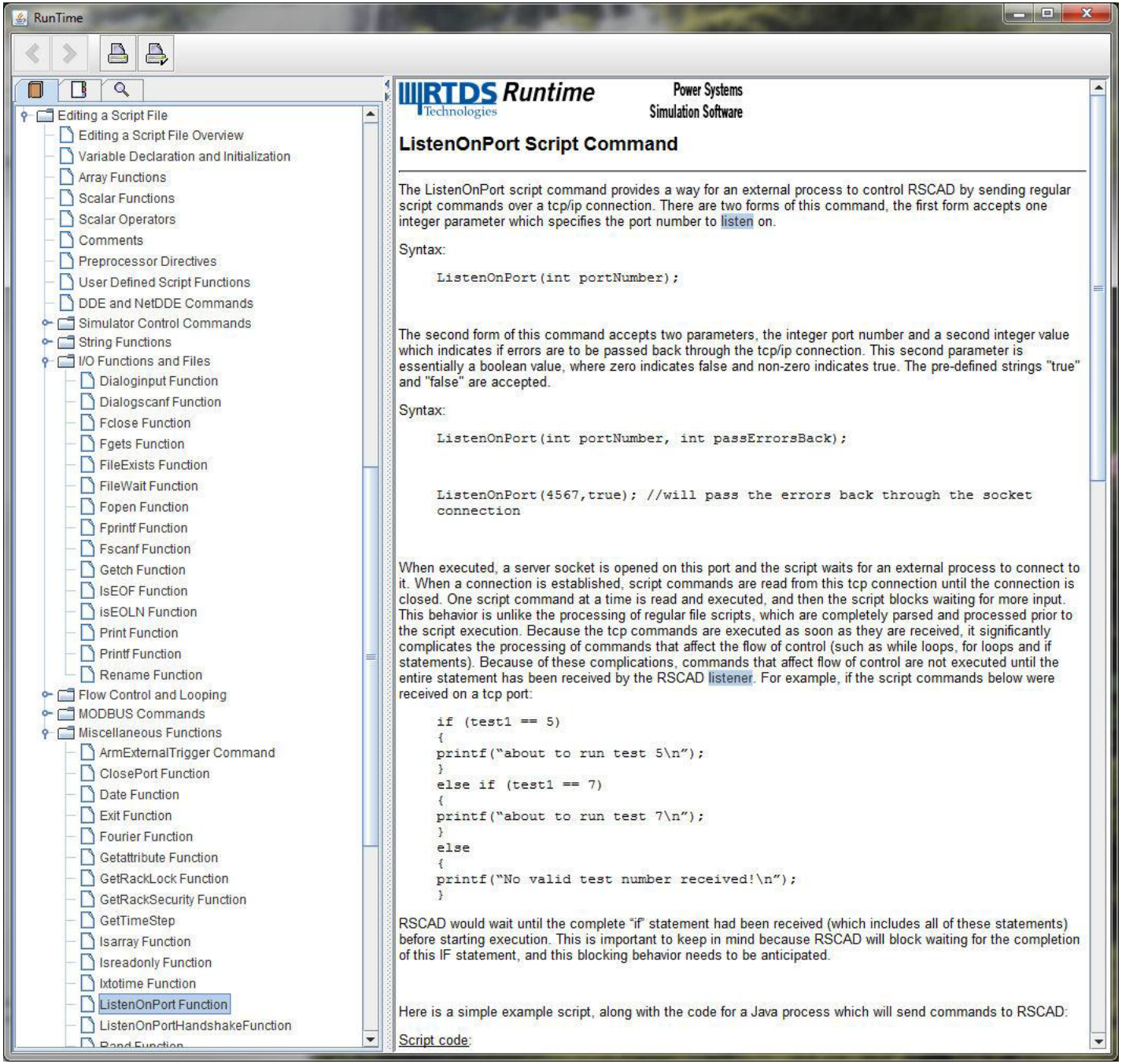
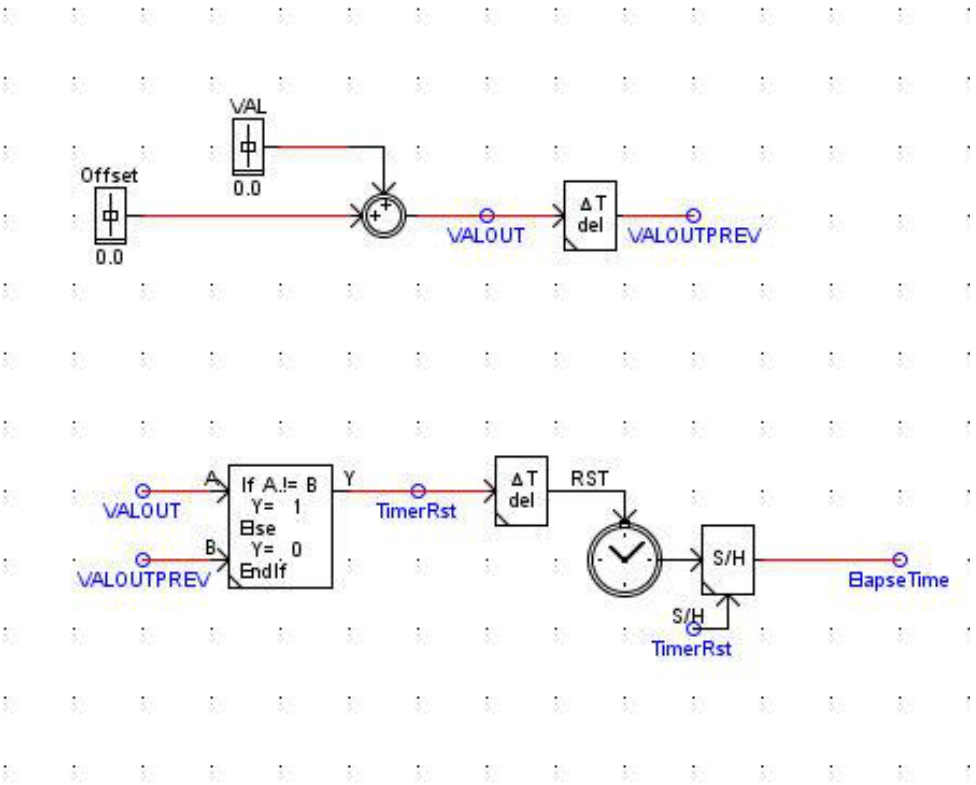


Figure 1 'ListenOnPort()' command RSCAD/Runtime Help

TU Delft

Built the connection with a MATLAB M file

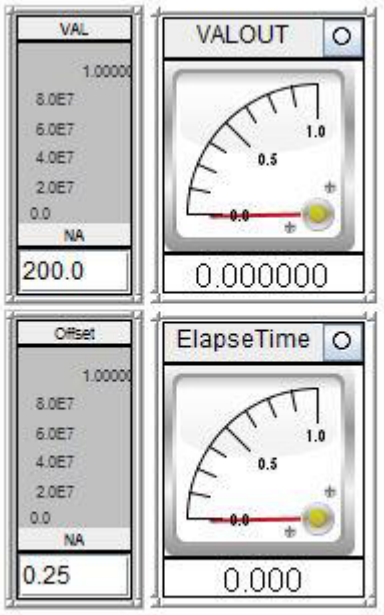
- The MATLAB M file is dependent on an external library regarding the TCP communication capability.
  - The 'JTCP.m' file is the library. (http://www.mathworks.com/matlabcentral/fileexchange/24524-tcpip-communications-in-matlab).



RSCAD Draft Circuit

Built the connection with a MATLAB M file

RSCAD Runtime



- The 'VAL' slider would be used as input to the simulation case
- The external application which connects to the RSCAD/Runtime would control the slider
- The external application would use the meter value as input.
- The second meter ('ElapseTime') displays the time between each change made to the 'VAL' slider.

Built the connection with a MATLAB M file

- *RSCAD Runtime script*

```
1  float temp_float;
2  string temp_string;
3
4  fprintf(stdmsg, "Initialization of RTDS simulation");
5
6  ListenOnPort(4575, true);
7
8  fprintf(stdmsg, "Execution of script is done\n");
```

The most important command in this script file is at the line number 6. It is the 'ListenOnPort()' command with the port number passed as a parameter. Regarding the port number, note that many standard or well-established network applications have already reserved many port numbers, which are known as 'Well-Known Port Numbers' (http://www.tcpipguide.com/free/t_CommonTCPIPApplicationsandAssignedWellKnownandRegi-2.htm).

As explained in the introductory section, once this command is executed, the port is open and the RSCAD/Runtime becomes a TCP server.

Built the connection with a MATLAB M file

- *Open the related M file in MATLAB*

Pay attention:

a. The command at line number 8 declares a variable with port number. This port number should match the port number given in the RSCAD/Runtime script file

b. The regular expression is utilized as a rigorous measure in order to make sure the transmission is complete and the token string is ready to be parsed at the client

c. The line number 12 tries to connect to the RSCAD/Runtime with connection request. The second and third parameters of this function call are the IP address and port number of the TCP server, which is RSCAD/Runtime in this demonstration

   1. '127.0.0.1' is used under the assumption that the same computer which runs RSCAD/Runtime also runs this MATLAB M file.

d. The 'for' loop, the lines between 17 and 58, send out 100 values with an interval of 0.2 seconds.

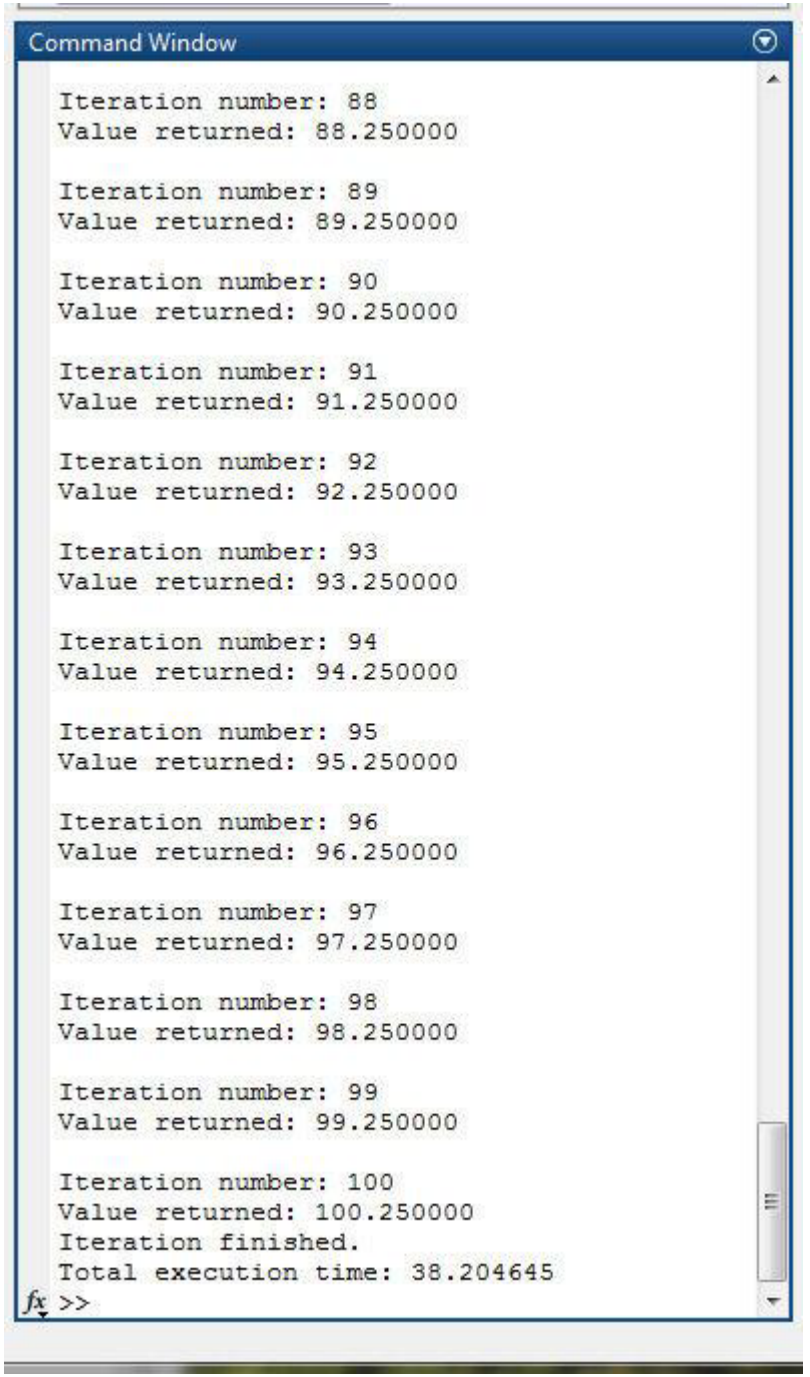
e. The output of the RTDS simulation is captured by the two commands at line number 31 and 32

f. The output of the RTDS simulation is captured by the two commands at line number 31 and 32

How to run the demonstration

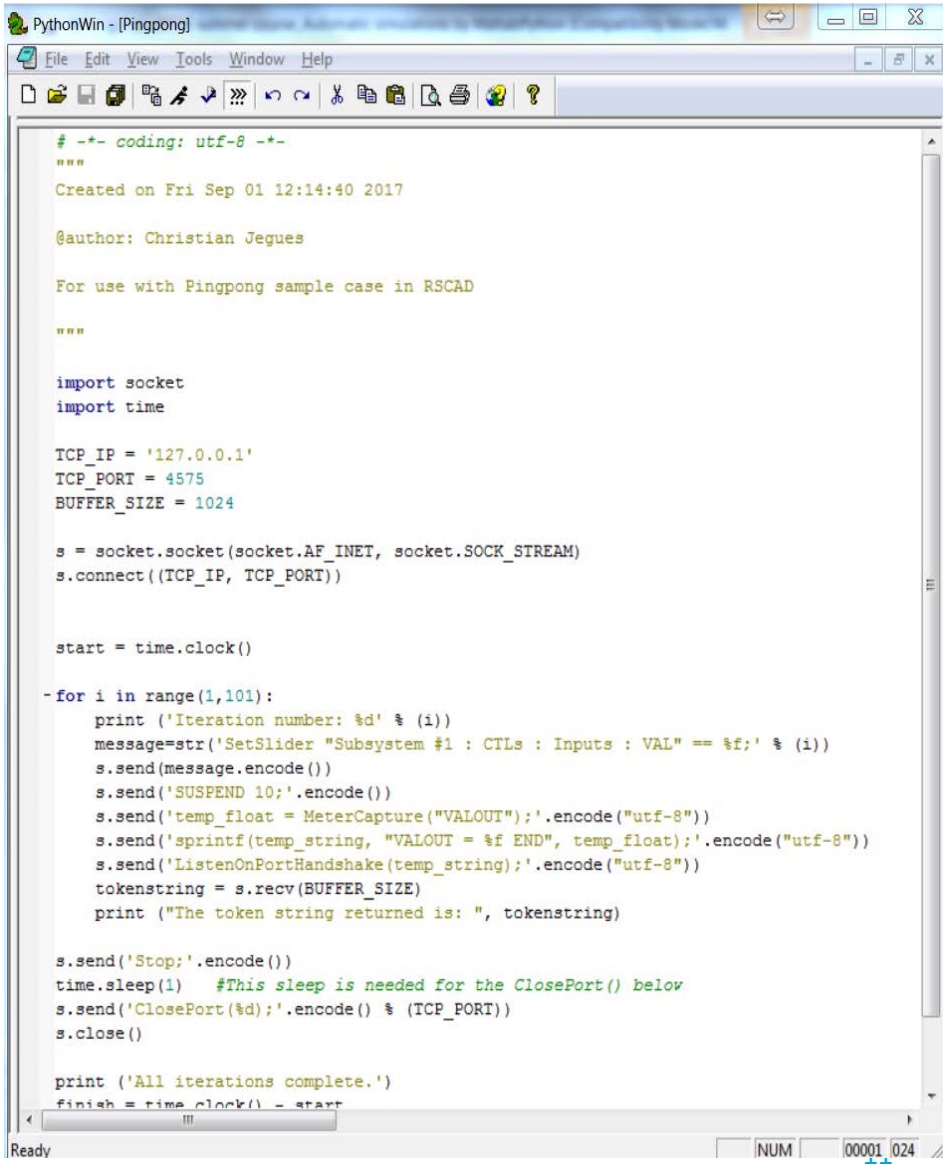
1. Compile the RTDS simulation case (‘Pingpong.dft’) in RSCAD/Draft. Make sure that there is no error or warning. Load the simulation case in RSCAD/Runtime. Don’t start the case yet. The case will be started by the external application (MATLAB M file).
2. Load the script file. Go to the ‘Script/Open’ menu and select the ‘ListenOnPort.scr’ script file. Start the script by selecting the ‘Run Script’ button in the script toolbar. Now, the RSCAD/Runtime becomes a TCP server and listens to the port number given by the script command.
   - In MATLAB, go to the directory where the MATLAB M code files (‘Pingpong.m’ and ‘jtcp.m’)are stored and run the ‘Pingpong.m’ M file from the MATLAB command window.
3. Then it can be seen that the RTDS simulation case starts and bi-directional communication, sending out a value from MATLAB to RTDS and receiving the corresponding result from RTDS to MATLAB, can be monitored at the MATLAB command window

The command window: Communication between RTDS and MATLAB: Result



Connecting with Python

- Any programming environment which supports TCP socket communication can be used when writing the necessary external application
- The programming environment selected for this demonstration is Python 3 programming environment (Pythonwin)
- The procedures of demonstration is as follows:
  1. Find the Pingpong.py in the related folder, and open it.
  2. Compile the RTDS simulation case (Pingpong.dft) in RSCAD/Draft without errors.
  3. Load the simulation case in RSCAD/Runtime and start it to run.

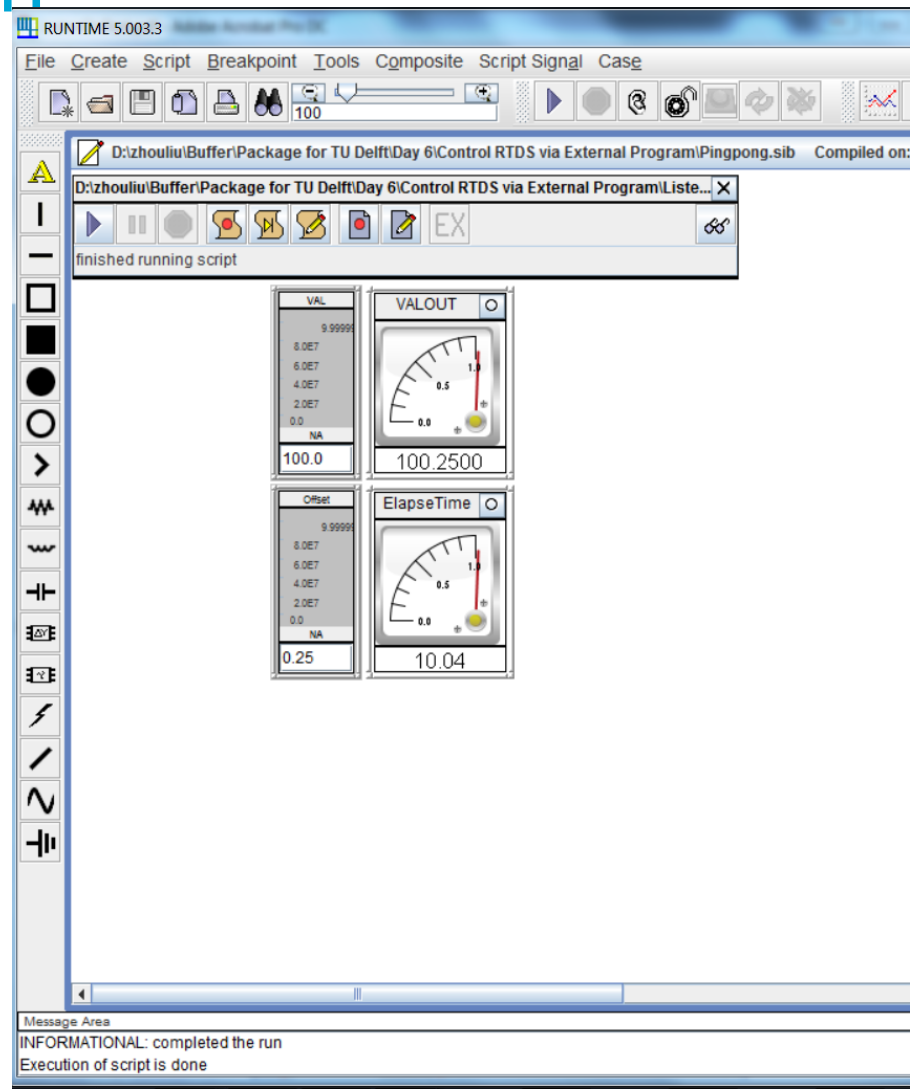


TU Delft

Connecting with Python

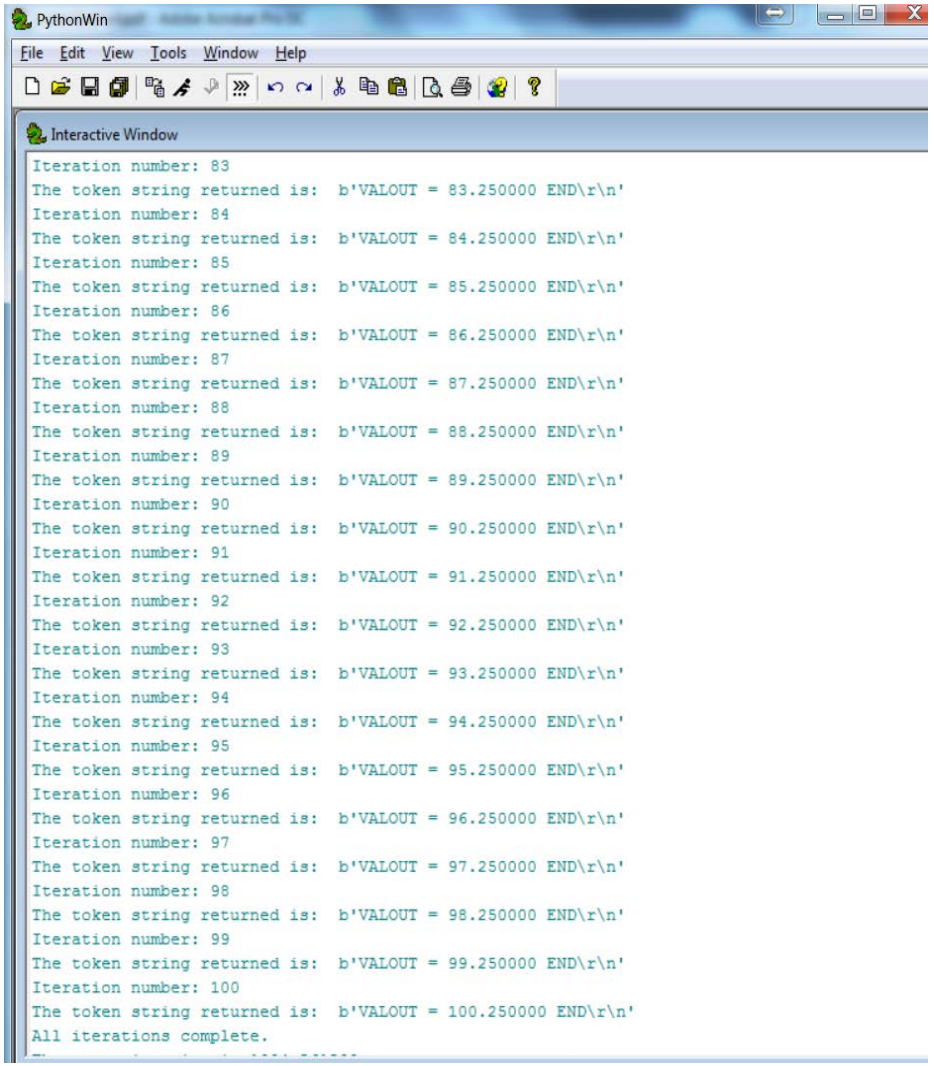
4. Load the script file. Go to the ‘Script/Open’ menu and select the ‘ListenOnPort.scr’ script file.
5. Start the script by selecting the ‘Run Script’ button in the script toolbar

Now, the RSCAD/Runtime becomes a TCP server and listens to the port number given by the script command.



Connecting with Python

6. Then let the program in Pythonwin run by click the related button
7. The slider of variable “VAL” has been changed every 0.2s from 1 to 100.
8. The returned results can be seen from the shell window of related Python programming environment.



Thank you for your attention!

Questions ?

IEPG, ESE, TU Delft