AN5796

Application note

STM32WL Series RF bench

Introduction

This application note summarizes the usage of Python scripts for automated RF measurements of STM32WL designs. The scripts are intended mainly as inspiration for specific measurements, which can be customized. There is a large variation in the measuring instruments that can be used. Different instruments may be controlled in different ways.

There are two scripts: one for transmitter measurements, and one for receiver measurements.

The output is in CSV format. It can be easily post-processed in any spreadsheet editor.

1 General information

The STM32WL Series microcontrollers are based on the Arm®Cortex®-M processor.

Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Table 1. Acronyms and terms

Acronym	Definition
COM	Communication port
CSV	Comma-separated values
IDE	Integrated development environment
IP	Internet protocol
DUT	Device under test
LoRa	Long range radio technology
LoRaWAN	LoRa wide-area network
PC	Personal computer
PER	Packet error rate
RF	Radio frequency
Rx	Reception
Tx	Transmission

Reference documents

[1] Application note *LoRaWAN® AT commands for STM32CubeWL* (AN5481)

2 Prerequisites

2.1 Measurement PC

The measurement PC runs the Python scripts and controls the RF instruments and the DUT (device under test).

- The appropriate driver must be installed to control the RF instruments. Follow the instrument vendor recommendations.
- The DUT is controlled by AT commands, which are text commands sent over UART. Refer to document [1] for more details.
- If physical COM port is not available, virtual COM port is used. The appropriate driver must be installed in the PC.

2.2 Python

The scripts run in Python 3. Most of libraries are standard, which are included in the default Python installation. The following additional libraries are used:

- serial
- visa
- numpy

2.3 RF instruments

The following RF instruments are required:

- Spectrum analyzer (tested device R&S FSV)
- Signal generator (tested device R&S SMBV)

If there are different types used than the tested instruments, the scripts may need to be modified.

2.4 VISA address

RF instruments are addressed via the VISA address.

The graphic below shows an example how to find VISA address in NI-MAX software. The Ethernet connection must be added manually (check the IP address in the instrument).

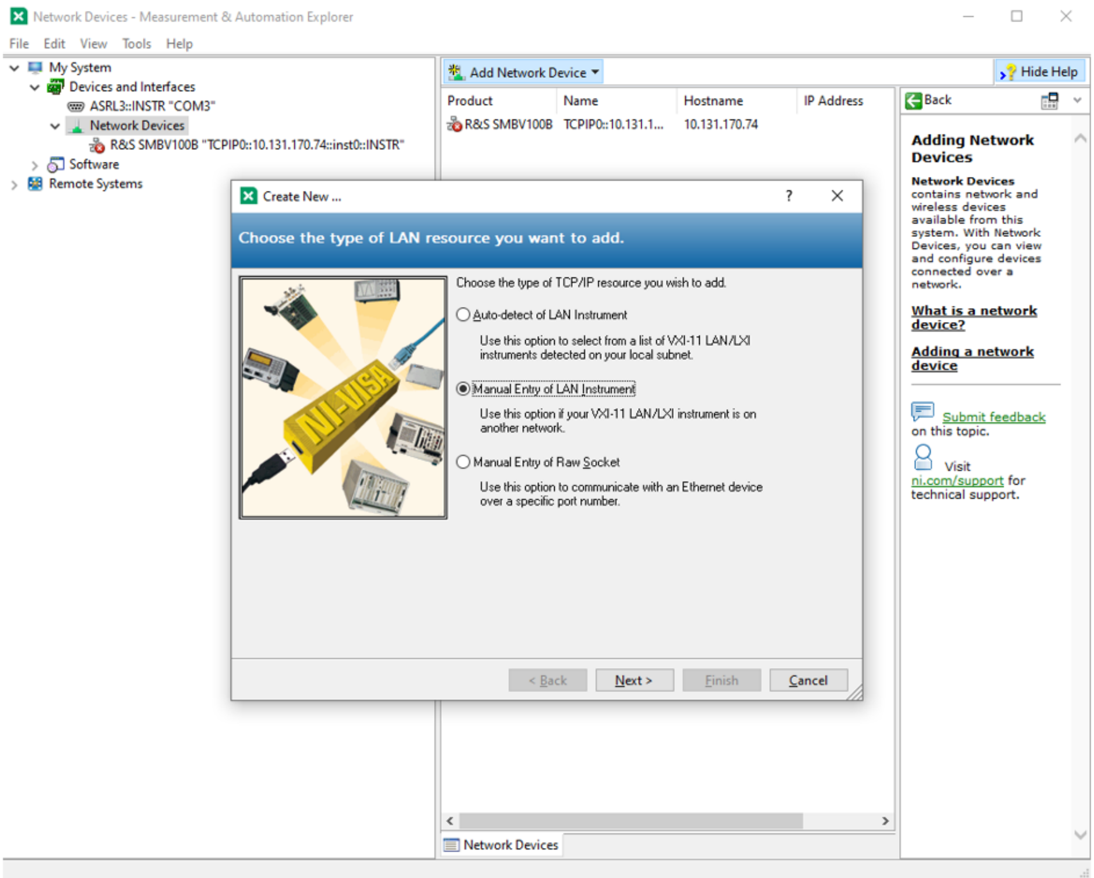


Figure 1. Manual entry of LAN instrument

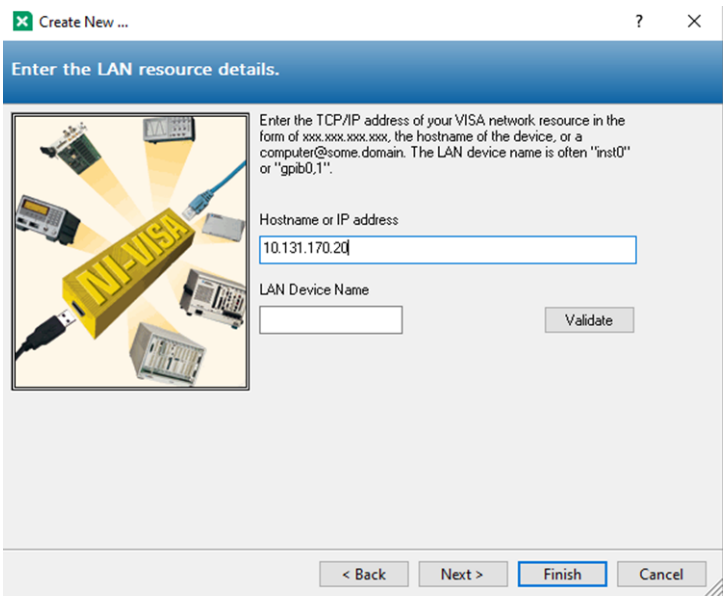
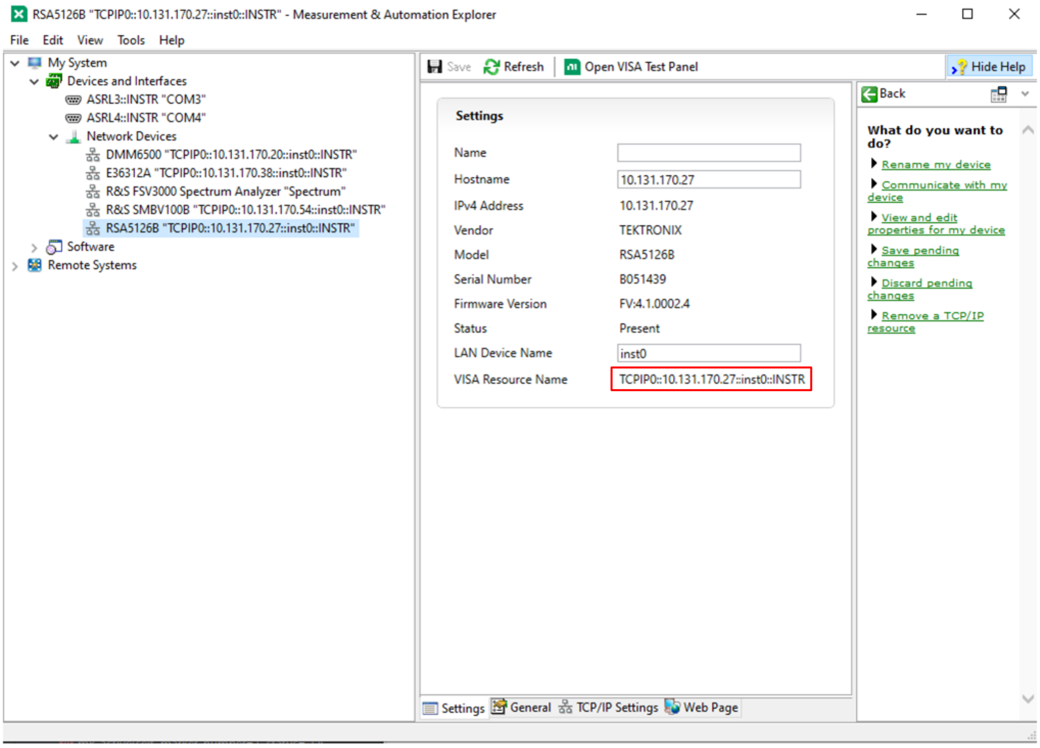


Figure 2. Entering the IP address

Figure 3. **Assigned VISA address**



2.5 Device under test (DUT)

It is a measured device. The LoRaWAN_ATSlave firmware is loaded in it. It is a project in the STM32CubeWL package. It is originally for STM32WL Nucleo board (MB1389). It must be ported to the target platform. The supported version of AT_Slave is v1.2.0.

3 Scripts

3.1 Common features

Follow the steps below for the scripts usage:

1. Load DUT with the AT_Slave firmware
2. Copy I/Q pattern to the RF generator (needed only for RX measurements)
3. Connect instruments and DUT, power on
4. Open the Python script in a text editor or Python IDE
5. Configure the script
6. Save the script and run
7. Process the results saved in the CSV file. The columns are separated by semicolons.

There are two RF instrument drivers:

- Spectrum_FSV_Driver.py for spectrum analyzer R&S FSV
- SigGen_SMBV_Driver.py for signal generator R&S SMBV

They separate specific commands for the instruments from the application code. This can help with customization if different instruments are used, for example, from different vendors.

3.2 Configuration

The scripts have user configuration area where parameters of the measurement are set.

Figure 4. Example of user configuration area

```
#####################################################
# USER CONFIGURABLE AREA                            #
# Measurement settings                              #
#####################################################
...
SpectrumAdress = "USB0::0x0BBE::0x02D6::201005::INSTR"
...
#####################################################
# END OF USER CONFIGURABLE AREA                     #
#####################################################
```

Table 2. Common configuration variables

Variables	Specifications
projectName	Name of the project, is used in the CSV file (optional)
projectOperator	Name of the operator, is used in the CSV file (optional)
outputFolder	Name of the folder where the CSV files with the results are stored
SpectrumAdress	VISA address of the spectrum analyzer
SMBVAddress	VISA address of the signal generator (only for RX script)
DUTPort	COM port of the DUT is found in Windows (device manager) or by using the hyperterminal software.
cableAttenuation	Insertions loss between the DUT and the instrument. It is added to the measured results. If not used, is set to 0 (default state).

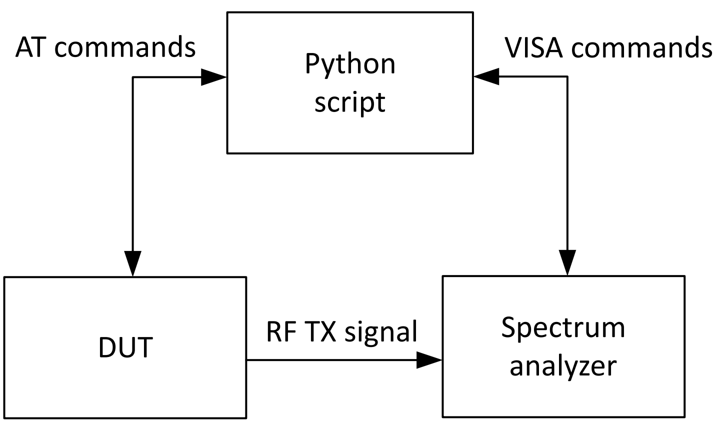
Specific configurations are described in Table 3. Specific configuration variables.

3.3 TX measurement

The script STM32WL_tx_power_harm.py is intended for automated measurement of the STM32WL TX output power and harmonics. It performs a frequency range sweep for each output power setting.

The use of the script is shown in the figure below:

Figure 5. TX measurement



Specific settings of the script:

- Frequency range:
  - There are two ways to perform TX measurements:
    - Use Fstart/Fstop/Fstep that creates a list of frequencies to measure.
    - Force Flist array with desired frequencies.
  - All values are in Hz.
- Power range:
  - As frequency, create a range of power to measure or specific list of power link with frequency (Plist).
  - All values are in dBm.

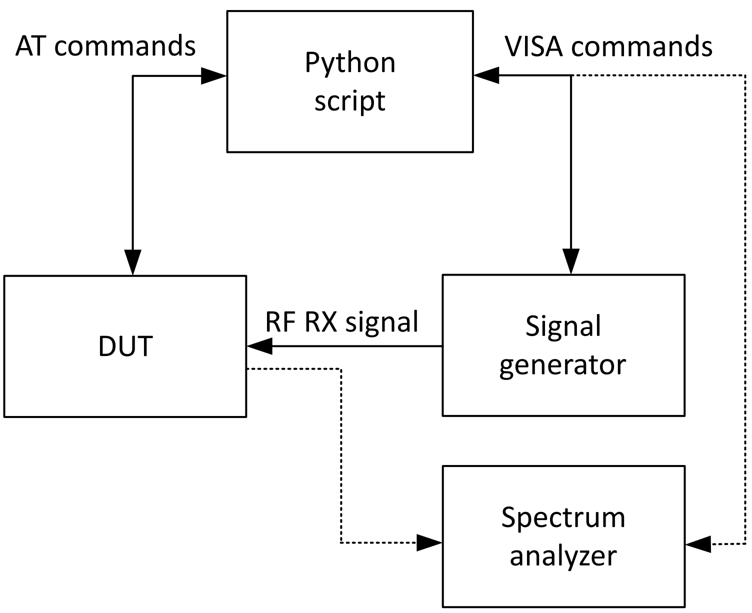
Note: Values Fstart/Fstop/Fstep must be integer and not float (mandatory for numpy.arange()), but can be forced, in Flist, for own frequencies.

3.4 RX measurement

The script STM32WL_rx_sensitvity_lora.py is intended for automated measurement of the STM32WL RX sensitivity of LoRa modulated signal. The sensitivity can be measured in a selected frequency range.

The use of the script is shown in the figure below:

Figure 6. RX measurement



The signal generator generates a LoRa modulated test signal. It is generated in arbitrary modulator based on LoRa waveforms. The waveforms are stored in the directory "Patterns_Lora" and must be manually copied into the signal generator (pattern_path setting in the script). Appropriate waveforms are used according to the settings in the script.

The spectrum analyzer is optional. It is needed only for frequency offset measurement.

Table 3. Specific configuration variables

Variables	Specifications
pattern_path	Path to LoRa waveforms in the signal generator
board_config	Can be set to HP (high power) or LP (low power)
frequency_list_Hz	List of frequencies for sensitivity measurements. Values are in Hz.
nb_packet	Number of packets to perform the sensitivity measurement. Should be between 100 and 1000 for high data rate.
limitPER	PER value to be achieved for sensitivity measurement
Precision_dB	Precision of the dichotomy algorithm
Power_min/max_dBm_init	Limits for the dichotomy algorithm
Lora_Bw(1)	Bandwidth of LoRa modulation. Value is in kHz.
LoRa_SF	The spread factor of the LoRa modulation can be set from 5 to 12.
Frequency_calibration	• If **True**, the DUT should be connected to the spectrum analyzer at first, to measure frequency offset. This offset is used in the signal generator. • If **False**, the script uses force_delta_frequency_Hz (zero by default) to correct signal generator frequency.

1. At the moment, only 125 kHz is supported.

Revision history

Table 4. **Document revision history**

Date	Version	Changes
12-May-2022	1	Initial release.

Contents

List of tables

List of figures