

Application Note

74-0039-170418

Using the ThinkRF R5500 Real-Time Spectrum Analyzer with MATLAB®

ThinkRF provides MATLAB® drivers for connecting to ThinkRF R5500 Real-Time Spectrum Analyzers and MATLAB® program code examples to get you started towards developing your own.

The ThinkRF R5500 Real-Time Spectrum Analyzer has the performance of traditional high-end lab spectrum analyzers at a fraction of the cost, size, weight and power consumption and is designed for distributed deployment.

MATLAB® is a high-level language and interactive environment that enables you to perform computationally intensive tasks such as matrix computation, plotting of functions, implementation of algorithms, signal processing, and system simulations.

Contents

Required Software	3
The R5500 Real-Time Spectrum Analyzer.....	3
Download and Compile the ThinkRF API and Example Scripts for MATLAB®	3
<i>For MATLAB® 2012a or Later</i>	3
<i>For MATLAB® 2011b or Older on a 32 Bit Platform</i>	4
<i>For MATLAB® 2011b or Older on a 64 Bit Platform</i>	4
<i>Common Instructions for All Versions of MATLAB®</i>	4
MATLAB® API for Windows with Examples	5
Using the ThinkRF R5500 MATLAB® Scripts	6
Implementing Custom MATLAB® Scripts for the R5500	6
<i>Examples (folder)</i>	6
<i>gui-implementation (folder)</i>	6
<i>signal-processing-implementation (folder)</i>	6
<i>utilities (folder)</i>	7
<i>readme (folder)</i>	7
Contact us for more information	7

Required Software

To follow the steps described in this Application Note, the following software is required:

- Windows 7/8/10 32-bit/64-bit operating system;
- MATLAB® versions from v7.11 to v7.14 (released 2010b to 2012a respectively);
- R5500 Software and Firmware Release Package. The Release Package may be download from <http://www.thinkrf.com/firmware-updates/>.

The R5500 Real-Time Spectrum Analyzer

This Application Note also assumes you have access and are connected to a R5500 Real-Time Spectrum Analyzer with firmware version of 1.4 or later.

You can connect via the Internet to the ThinkRF evaluation R5500s. The R5500 is designed for distributed or remote deployment and as such you can use and/or evaluate any of our third-party software applications by connecting to one of ThinkRF's R5500s which we have deployed on the internet. You may access to one of our evaluation units on the internet from <http://www.thinkrf.com/demo/>.

Download and Compile the ThinkRF API and Example Scripts for MATLAB®

ThinkRF's MATLAB® API require the use of MATLAB® MEX-files to provide network interoperability with the ThinkRF R5500. MEX stands for MATLAB® Executable. MEX-files are dynamically linked subroutines produced from C, C++ or Fortran source code that, when compiled, can be run from within MATLAB® in the same way as MATLAB® files or built-in functions. The external interface functions provide functionality to transfer data between MEX-files and MATLAB®, and the ability to call MATLAB® functions from C, C++ or Fortran code.

MEX-files have to be compiled on the target computer and in co-existence with the installed MATLAB® so that the MEX-files work with that specific version of MATLAB®. ThinkRF's MEX-files have been written and tested for compilation by Microsoft Visual C++ 2010. See MATLAB® documentation for a list of supported compilers in the current MATLAB® release and in previous MATLAB® releases.

For MATLAB® 2012a or Later

Use Microsoft Windows SDK 7.1 and see MATLAB® support for detailed instructions. The following is a summary:

1. If you have any Visual Studio 2010 runtime redistributable packages installed, uninstall them through the Control Panel "uninstall program" tool. This is to avoid a known problem with installing the SDK.
2. Install .NET framework 4.0.
3. Install Microsoft Windows SDK 7.1 (note: if running on a 64 bit platform, ensure you select the option to install the x64 Compilers and Tools).
4. Install Visual Studio C++ 2010 SP1 compiler.

5. Install the following patch for MATLAB® 7.10 (R2010a).

See MATLAB® support for detailed instructions.

For MATLAB® 2011b or Older on a 32 Bit Platform

Use Microsoft Visual C++ 2010 Express:

1. Install Visual C++ 2010 Express (note: you don't have to install the optional components such as SQL server or Silverlight).
2. Install the following patch for MATLAB® 7.10 (R2010a).

For MATLAB® 2011b or Older on a 64 Bit Platform

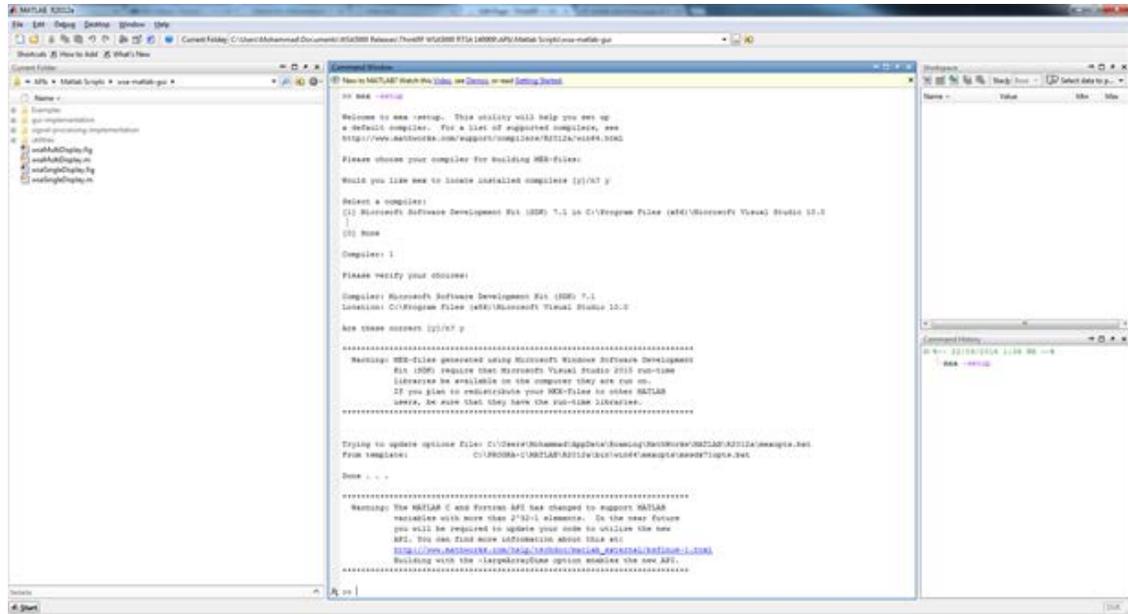
Use Microsoft Visual C++ 2010 Express with Microsoft Windows SDK 7.1. See MATLAB® support for detailed instructions. The following is a summary:

1. If you have any Visual Studio 2010 runtime redistributable packages installed, uninstall them through the Control Panel "uninstall program" tool. This is to avoid a known problem with installing the SDK.
2. Install .NET framework 4.0.
3. Install Microsoft Windows SDK 7.1 (note: if running on a 64 bit platform, ensure you select the option to install the x64 Compilers and Tools).
4. Install Visual C++ 2010 Express (note: you don't have to install the optional components such as SQL server or Silverlight).
5. To overcome a known issue with Visual C++ 2010 Service Pack 1, it may be necessary to install Visual Studio C++ 2010 SP1 compiler.
6. Install the following patch for MATLAB® 7.10 (R2010a).

Common Instructions for All Versions of MATLAB®

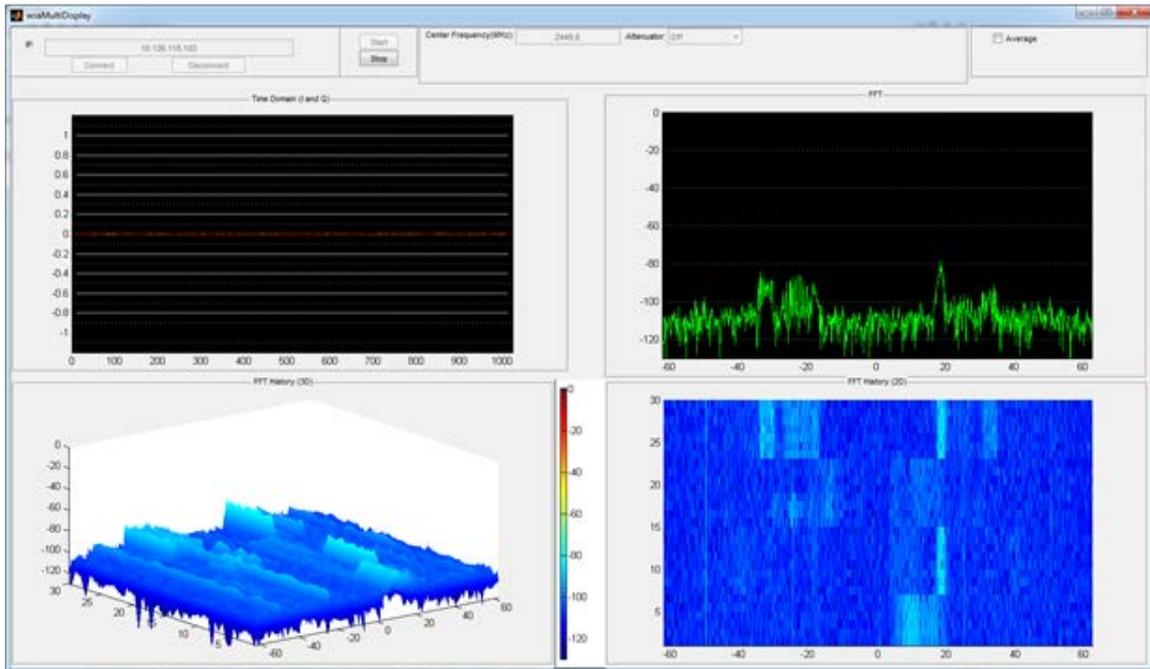
The following instructions are common for all versions of MATLAB® and provide for compilation of the MEX-files:

1. Type "*mex -setup*" within MATLAB®'s command window.
2. Follow the displayed instructions to select the appropriate compiler.
3. Download the latest R5500 release package.
4. The MATLAB API is located in the "*Software\MATLAB Scripts\wsa-MATLAB-api*" directory inside the release package.
5. The MATLAB Example scripts are located in the "*Software\MATLAB Scripts\wsa-MATLAB-gui*" directory inside the release package.
6. Copy the MATLAB API and example scripts to your preferred MATLAB style="*font-size:small;*"> workspace.
7. Launch MATLAB®.
8. Navigate to the MATLAB® workspace that contains the example script (*wsa-MATLAB-api*).
9. Execute *buildWsaMex.m* to compile the mex files.



MATLAB® API for Windows with Examples

In the *wsa-MATLAB-gui* directory, there are two MATLAB® applications that can be used to interface with the R5500 (*wsaMultiDisplay.m*, and *wsaSingleDisplay.m*). When you launch *wsaMultiDisplay.m*, the following window will appear.



Using the ThinkRF R5500 MATLAB® Scripts

In addition to the *wsaMultiDisplay.m* script, we also provide the *wsaSingleDisplay.m* script which is fairly self-explanatory and provides additional control and display options.

The following assumes that you have already invoked the ThinkRF *wsaMultiDisplay.m* script by double-clicking it within MATLAB®'s *Current Folder* window and pressing the *F9* key to run the script (or by right-clicking *wsaMultiDisplay.m* and selecting *Run*, or by pressing the *F5* key within MATLAB®'s *Editor* window).

1. Enter the IP address of the R5500 within the *IP* dialog box in the upper banner of the *wsaMultiDisplay* window, for example `"192.168.215.104"` or `"kanata.demo.thinkrf.com"` excluding the quotes.
2. Click the *Connect* button. You are now connected to the R5500.
3. Enter the frequency of interest in the *Center Frequency* dialog box, for example `"2450"` excluding the quotes.
4. Select the Attenuation state (On or Off).
5. Click the *Start* button. The R5500 will now stream IQ data for visualization.
6. Please note that you must click the *Stop* button in order to change R5500 settings.

Implementing Custom MATLAB® Scripts for the R5500

The following is a description of the files and directories contained with the release bundle's *SoftwareMATLAB scripts* directory.

There are two different example scripts of the GUI application, each with slightly different controls and visualization examples and both of which are in the top level folder. The *.m* files can be executed to start the application while the *fig* files are just helper files that store the GUI layout.

1. *wsaMultiDisplay.m* and *wsaMultiDisplay.fig*
2. *wsaSingleDisplay.m* and *wsaSingleDisplay.fig*

Examples (folder)

Contains simple scripts that demonstrate how to interact with the WSA and perform data processing in MATLAB®.

gui-implementation (folder)

Contains helper scripts for the GUI application. These files may be disregarded.

signal-processing-implementation (folder)

Contains helper scripts for doing the data processing (windowing, FFT). These files may be disregarded or used as examples.

utilities (folder)

Some other helper scripts for the GUI application, such as constant definitions and tests. These files may be disregarded.

readme (folder)

Contains a version and license files. The following provides some tips on interacting with the R5500 using custom developed MATLAB® scripts

1. Add the "*wsa-MATLAB-api*" directory to MATLAB®'s path to enable interaction with the R5500 from the custom MATLAB® scripts.
2. The script *wsaBasic.m* in the "*samples*" directory provides an example.
3. Start by creating a new *R5500 Interface* class handle using:
 - o *wsaHandle = wsaInterface*
4. Resources are cleaned up automatically, even if an error occurs (such as: forgetting to clean the handle, unintentionally closing MATLAB®, etc.). Regardless, it is good practice to close the handle manually at the end of the script using:
 - o *clear wsaHandle.*

Upon creation of the *wsaHandle*, you can now call functions such as *connectToWSA()*, *setFrequency()*, etc.

Contact us for more information

Please contact us via email at sales@thinkrf.com or via phone at +1.613.369.5104.

© 2017 ThinkRF Corporation, Ottawa, Canada, thinkrf.com

Trade names are trademarks of the owners.

These specifications are preliminary, non-warranted, and subject to change without notice.