Using the UHD USRP2 Block in Simulink

Installing the software

1. Go to: http://www.mathworks.com/discovery/sdr/usrp.html and click on: MATLAB and Simulink Support Package for USRP™ Hardware. 0
2. Complete and submit the form.
3. Click on: Download the UHD interface, demos, and documentation for use with Communications System Toolbox (R2011a or later).
4. A zip file will be downloaded

Installing the USRP/UHD Blocks

1. Extract the contents of the downloaded zip file into a folder of your choice. The folder must not be within the MATLAB installation folder.
2. Start MATLAB and navigate to the folder where you extracted the files in step 1.
3. Type "install" at the MATLAB prompt. On Windows, install must be executed from a non-UNC path.
4. After installation, type "help sdru" for information on using Communications with USRP(TM).
5. In future MATLAB sessions, you must run setupsdru to use Communications with USRP(TM). You can automate this step by adding setupsdru to your startup.m file. If you prefer to run it manually you can use the shortcut, labelled "Add SDRu", that has been provided on the MATLAB shortcut bar.
6. The installer will run setupsdru when you first install Communications with USRP(TM).

Finding the USRP2 Blocks

SDRULIB Open the Communications with USRP(TM) block library.

SDRULIB opens the latest version of the Communications with USRP(TM) block library.

A Simulink sublibrary is also created.

Getting Started with the SDRu

This page serves as a reference site to enable you to most effectively use the MathWorks implementation of SDRu I/O blocks and System objects. Before you use the material on this page, you will benefit from reviewing the material on the USRP™ hardware itself. The hardware manual can be found at http://www.ettus.com/uhd_docs/manual/html/index.html, and provides notes for USRP2™ hardware, N210 hardware, daughter boards, network setup, and troubleshooting communication and connectivity problems.

Downloading Firmware and FPGA Binaries for the USRP™ Hardware

To download firmware for the USRP2™ hardware, follow the instructions at http://www.ettus.com/uhd_docs/manual/html/usrp2.html#load-the-images-onto-the-sd-card-usrp2-only

If you are using the N210 hardware, then follow the instructions at http://www.ettus.com/uhd_docs/manual/html/usrp2.html#load-the-images-onto the-on-board-flash-usrp-n-series-only

These steps require that you have Python installed on your computer. If it is not already loaded on your computer, you can download it from the official Python site, at http://www.python.org/download/.

Configuring Your Host Computer

You need a dedicated gigabit Ethernet card for the USRP™ hardware. If you also want Internet access, your host computer requires a second Ethernet card.

To configure the Ethernet card for your USRP™ hardware, perform the following tasks:

If you are using Windows XP:

1. Select Start -> Settings -> Network Connections.
2. Select the network card to which the USRP™ device will be attached.
3. Select 'Internet Protocol -> TCP/IP', then click the 'Properties' button.
4. Select 'Use the following IP Address'.
5. Set the IP address to 192.168.10.1, and the subnet mask to 255.255.255.0.

If you are using Windows 7:

1. Select 'Control Panel' from the window icon in the lower left corner of your monitor.
2. From the Control Panel, select 'Network and Sharing Center'.
3. Select 'Change adapter settings' on the left sidebar.
4. Select the appropriate network device, then select 'Change settings of this connection'.
5. On the 'Networking' tab of the dialog, select 'Internet Protocol Version 4 (TCP/IPv4), then select 'Properties'.
6. Select 'Use the following IP address:'
7. Set the IP address to 192.168.10.1, and the subnet mask to 255.255.255.0.

If you are using Linux:

1. Set your host's Ethernet interface with a static IP address to enable communication. An address of 192.168.10.1 and a subnet mask of 255.255.255.0 are recommended.

Verifying that Your SDRu Block or System Object is Connected to the USRP™ Hardware

Apart from using blocks or System objects, you can enter 'findsdru' at the MATLAB command line to find the USRP™ devices that are connected to your private network. If findsdru returns 'No devices found' and yet you believe you have the NIC properly configured and a USRP™ device attached, ensure that your host computer does not have any firewall blocking the discovery of the attached device.

Block

You can verify that your SDRu Transmitter or SDRu Receiver block is connected to USRP™ hardware by examining the block mask. If a block is connected, then the Hardware panel will show values for the following parameters:

- Motherboard
- Receiver subdevice
- Transmitter subdevice
- Minimum center frequency
- Maximum center frequency
- Minimum gain
- Maximum gain
- Gain step size

If the block is not connected, the Hardware panel will show the following message:

"Select a connected USRP device using the 'USRP IP address' drop-down parameter or enter in your own IP address to work in off-line mode."

Also, note that a block can only connect to one subdevice (transmitter or receiver) at a time, on the same IP address.

System Object

To verify that your SDRu System object is connected to the USRP™ hardware:

2. Construct an object by typing h=comm.SDRuTransmitter or h = comm.SDRuReceiver
3. Enter h.IPAddress = ', then press the Tab key

If the hardware is connected, then an IP address should appear in the resulting dialog. (The default IP address of the hardware is 192.168.10.2.) If no hardware is connected, then the dialog will say 'No devices found'.

A System object can only connect to one subdevice (transmitter or receiver) at a time, on the same IP address. If you change property values of an SDRu System object and display the actual values, the object will connect and then immediately disconnect once the properties are displayed. SDRu System objects connect to a subdevice when you call the step method, and will stay connected until you call the release method.

Maximizing Performance for Your Models and Programs

Simulink Model

Here is a list of performance improvements you can make in your Simulink models that contain SDRu blocks in order to approach, if not achieve, real-time execution:

1. Use frame-based processing, where your model or program processes multiple samples during a single execution call to a block or System object. Consider using frame sizes from roughly 100 to several thousand.
2. In the Configuration Parameters -> Data Import/Export dialog, turn off all logging.
3. Make sure that the model is single-rate. If the model requires resampling, then choose rational coefficients that will keep the model single-rate.
4. Do not add any Buffer blocks to the model. If you want to create convenient frame sizes, it is best to do so in your data sources. Using a Buffer block will typically degrade performance.
5. You should try to run with Rapid Accelerator instead of Normal mode. Be aware that some scopes do not plot data when run in Rapid Accelerator mode.
6. Try to avoid feedback loops. Typically, such loops imply scalar processing, which will slow down the model considerably.
7. Do not use scopes unless absolutely necessary. To visualize your data, send it to a workspace variable and post-process it.
8. If you are using Accelerator or Rapid Accelerator, set the Configuration Parameters -> Optimization -> Compiler optimization level to 'Optimizations on (faster runs)'.
9. If the model has many Constant blocks, it could help slightly if Configuration Parameters -> Optimization (Signals and Parameters) -> Inline parameters is checked on. This will cause the sample time of those Constant blocks to truly become inf -- Simulink will get the values once and only once during a run.
10. If the model generates code, the Solver setting should be Fixed-step/ discrete. The tasking mode should be SingleTasking.
11. You can generate a standalone executable for your Simulink model to improve performance. The generated code runs without Simulink in the loop. To perform any code generation, you must have an appropriate compiler installed. See http://www.mathworks.com/support/compilers/R2011a/win64.html for a list of supported compilers. You can generate generic real-time target (GRT) code if you have a Simulink Coder™ license. To do so, set the Configuration Parameters -> Code Generation -> System target file of your model to 'grt.tlc (Generic Real-Time Target)'.
12. When you generate code for any target (not just GRT), uncheck the Configuration Parameters -> Hardware Implementation -> Emulation hardware (code generation only) checkbox. Once the checkbox is unchecked, set the Device type popup to 'MATLAB Host Computer'.
13. You can create generated code with a smaller stack then the GRT code if you have an Embedded Coder™ license. To do so, set the Configuration Parameters -> Code Generation -> System target file of your model to 'ert.tlc (Embedded Coder)'. Also,

add following lines to the Configuration Parameters -> Code Generation -> Custom Code -> Include custom C code in generated: -> Source file edit field:

```
#include <stdio.h>

#include <stdlib.h>
```

You can also tune your computing environment to improve performance. Here is a list of steps to take:

1. Turn off anti-virus and firewall software.
2. Turn off all non-essential background processes in your computer.

MATLAB Program

If you use SDRu System objects, you do not have access to a Configuration Parameters dialog, so most of the above techniques do not apply. However, you should still use large frames of data to minimize function call overhead and maximize performance.

Setting IP Addresses

Block

If your SDRu I/O blocks are connected to the USRP™ hardware, then the valid IP address(es) will appear in the 'USRP IP address:' parameter in the block mask. You can also type a dotted quad IP address in the combo box.

System Object

To set the IP address for your SDRu System object, you can construct an object h. Then you can type 'h.IPAddress = ', then press the Tab key. The valid IP addresses appear in the resulting dialog.

You can also type 'h = comm.SDRuTransmitter('192.168.10.2')' or 'h = comm.SDRuReceiver('192.168.10.2')' to explicitly set the IP address at construction time. The 192.168.10.2 address is the default address of the USRP™ hardware.

Desired vs. Device Block Parameter Values

Block

When you set mask values for center frequency, gain, interpolation factor, and decimation factor, the block initially performs some rudimentary checks that the values are scalar and real. If your values pass those checks, you can still provide values that are out of range for

the USRP™ hardware. In that case, the hardware will make a best effort to set the requested value, and will report the actual value in the 'Device value' column of the block mask.

If your Simulink block is connected to the USRP™ hardware, you can check the 'Hardware' information panel to find the acceptable ranges of center frequency and gain. The acceptable ranges of interpolation/decimation are integer values as follows:

- 1 to 128
- 128 to 256, even
- 256 to 512, divisible by 4

System Object

You can set desired values in your SDRu System objects for center frequency, gain, interpolation factor, and decimation factor. You do this by setting the CenterFrequency, Gain, InterpolationFactor, and DecimationFactor properties, respectively. Due to quantization or range issues, the actual values may be different from the desired values. The actual values are stored in the ActualCenterFrequency, ActualGain, ActualInterpolationFactor, and ActualDecimationFactor properties.

Using Underrun and Overrun Outputs in SDRu Blocks and System Objects

Block

The SDRu Transmitter block has an optional underrun output port. When this port is active, it outputs a signal that indicates whether the block is providing data to the USRP™ hardware in real time. If the block is not keeping up with the hardware, the signal goes high.

Similarly, the SDRu Receiver block has an optional overrun output port. When this port is active, it outputs a logical signal that indicates whether the block is processing data in real time. If the block is not keeping up with the hardware, the signal goes high.

These ports are useful diagnostic tools to determine real time operation of the blocks. If your model is not running in real time, you can increase the interpolation/decimation factors on the block masks to approach or achieve real-time performance. Note that the reporting resolution is dependent on the frame size.

System Object

The corresponding SDRu System objects have optional underrun and overrun output ports as well, to be used for identical purposes. You can type 'help comm.SDRuTransmitter' and 'help comm.SDRuReceiver' to learn more about them.

Setting Sample Times in the SDRu Receiver Block and System Object

Block

The sample time that you set in the SDRu Receiver block is the Simulink sample time. It is not related to the sample time of the data that the board sends to the SDRu Receiver block. However, it is a good practice to set the block's sample time to match the hardware sample time. To do so, simply set the sample time to 1e8/Decimation factor. The 1e8 corresponds to the 100 MHz sampling rate of the A/D on the USRP™ hardware.

The sample time specified by the block mask is the time interval between successive output samples of the SDRu Receiver block. If you turn on the sample time colors of the model, via the "Format -> Sample Time Display -> All" menu item in your model, the sample time value at the output of the SDRu Receiver block corresponds to the Simulink sample time. This is the time interval between block outputs. The Simulink sample time is equal to the sample time specified in the block mask times the vector length at the output of the block.

System Object

The comm.SDRuReceiver System object has a 'SampleRate' property that is the reciprocal of the corresponding sample time of the SDRu Receiver block. That is, the SampleRate property is the sample rate between successive samples of MATLAB data, not successive frames of data.

Setting the Frame Length for the SDRu Receiver Block and System Object

You can set the frame length of the SDRu Receiver block and System object to be any integer value. This flexibility enables more straightforward multirate operation, and easier modeling of standards-based packet communications. The default value is 362 because that value optimally utilizes the underlying Ethernet payloads for a standard 1500-byte MTU.

Output Data Types for the SDRu Receiver Block and System Object

The default output data type of the SDRu Receiver block and System object is int16, with a range of [-32768, 32767]. You can also set the output data type to double or single precision. For those cases, the range is +/-1.

Shifting the Center Frequency Using LO Offset

General

You can shift the center frequency of the USRP™ hardware by using LO (local oscillator) offset. You may want to do this to 1) move the actual center frequency away from some interference, but still tune to the desired center frequency, or 2) compensate for oscillator differences between two devices.

Block

You can use the 'LO offset (Hz)' parameter to perform this shift. If you set the parameter to a positive number, then the center frequency will shift lower.

System Object

You can use the 'LocalOscillatorOffset' property to perform this shift. Once again, if you set the property to a positive number, then the center frequency will shift lower.

Using Conditional Execution in Receiver Processing

Block

The SDRu Receiver block has a static Data Len port that outputs either 0 or the frame length specified in the block mask. When the output is 0, then Simulink is running faster than the hardware, and the hardware did not have any new data to send to Simulink at the instant of the sample time hit. When the output is the specified frame length, then the hardware sent valid data.

Any processing downstream of the SDRu Receiver block must run only on valid data, so it must be controlled by an enabled subsystem. The Data Len output serves as a convenient control signal for such an enabled subsystem. See the sdruFMMono model for an illustration of this modeling technique.

System Object

The SDRu Receiver System object uses a similar 'data length' output of the 'step' method to indicate the length of data packets streaming to MATLAB from the USRP™ hardware. If the length is 0, then the hardware has not sent any valid data. If the length equals the specified FrameLength, then it has sent valid data. Any downstream processing must be conditioned on the presence of valid data. See sdruFMMonoReceiver for an example of this technique.

Available Demos

The following USRP™-based Simulink demos are available:

1. sdruFMMono
2. sdruFMStereo
3. sdruFRSGMRSTx
4. sdruFRSGMRSRx

The following USRP™-based MATLAB demos are available:

1. sdruFMMonoReceiver
2. sdruFMStereoReceiver
3. sdruFRSGMRSTransmitter
4. sdruFRSGMRSReceiver