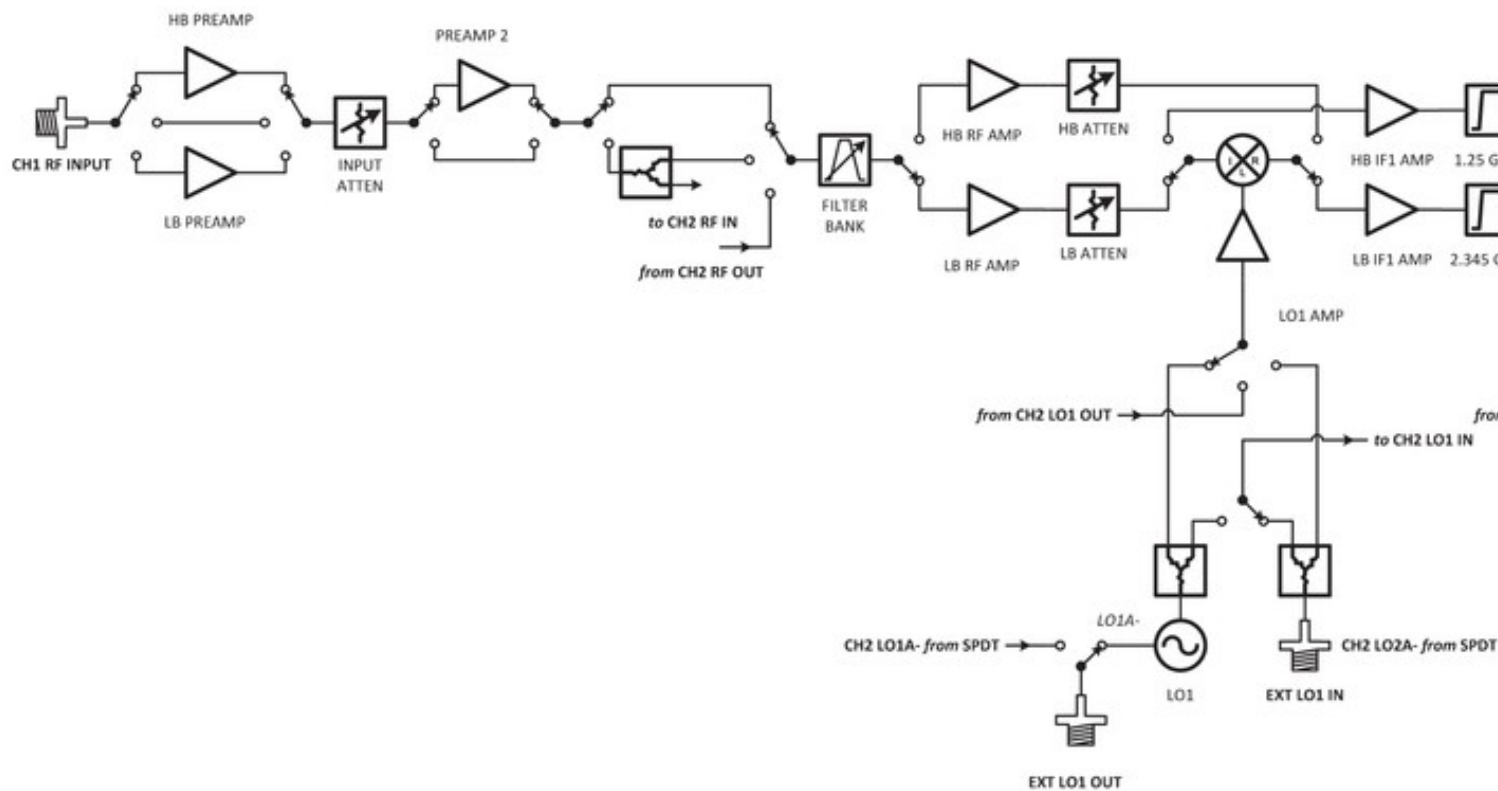


# TwinRX

## Contents

- 1 Device Overview
- 2 Key Features
- 3 Daughterboard Specifications
  - ◆ 3.1 Features
  - ◆ 3.2 Antennas
  - ◆ 3.3 Gains
  - ◆ 3.4 Bandwidths
  - ◆ 3.5 Sensors
  - ◆ 3.6 Ports
  - ◆ 3.7 Preselector Filters
- 4 RF Specifications
  - ◆ 4.1 Freq Range
  - ◆ 4.2 Noise Figure
  - ◆ 4.3 RX Third Order Intercept (dBm)
  - ◆ 4.4 Phase Noise (dBc/Hz)
  - ◆ 4.5 Input/Output Impedance
  - ◆ 4.6 Input Power Levels
- 5 Hardware Specifications
  - ◆ 5.1 TwinRX
- 6 Environmental Specifications
  - ◆ 6.1 Operating Temperature Range
  - ◆ 6.2 Operating Humidity Range
- 7 USRP Compatibility
- 8 Phase Synchronization
- 9 Schematics
- 10 RF Connectors
- 11 Local Oscillator Connectors
  - ◆ 11.1 LO Sharing with Neighbour TwinRXs
- 12 Software API
  - ◆ 12.1 New Multi USRP Functions
  - ◆ 12.2 Antenna Mapping
  - ◆ 12.3 Local Oscillator Control and Routing
  - ◆ 12.4 Exporting Local Oscillators
- 13 Drawings
- 14 Certifications
  - ◆ 14.1 RoHS
  - ◆ 14.2 China RoHS
- 15 Certificate of Volatility
- 16 Downloads

The TwinRX daughterboard for the USRP X300 and X310 SDR platform is a two-channel superheterodyne receiver designed for high performance spectrum monitoring and direction finding applications. The receiver is tunable from 10 MHz - 6 GHz and has 80 MHz of instantaneous bandwidth per channel, providing the versatility necessary to analyze a variety of signals in multiple bands of interest. Each channel has an independent RF signal chain with preamplifiers, preselectors, and two mixer stages for superior selectivity. Users can tune the two channels independently to simultaneously monitor uplink and downlink communication with a combined bandwidth of 160 MHz. The ability to share the LO between channels across multiple daughterboards enables the phase-aligned operation required to implement scalable multi-channel phased-arrays. The receiver is capable of fast frequency hopping to detect frequency agile emitters. Configurable RF attenuation and preamplification allow users to optimize dynamic range in favor of noise figure for faint signals, or IP3 for stronger signals. UHD automatically configures the RF signal path for optimized performance in the pre-defined use cases, and provides the flexibility to adjust settings manually. Support for RFNoC on the X Series motherboard enables deterministic FPGA-accelerated computations for real-time spectrum analysis. The TwinRX daughterboard is supported by the USRP Hardware Driver? (UHD) software API for seamless integration into existing applications.



- Two-Channel Superheterodyne Receiver
- Frequency Range: 10 MHz - 6 GHz
- Bandwidth: 80 MHz per channel ( 160 MHz total )
- RF shielding
- Independent RF signal channels with optional LO sharing



- 2 superheterodyne frontends (2 receive)
- 80 MHz per channel
- Independent tuning
- LO Sharing Capability

- Coherent and phase-aligned operation
- Preselection Filters
- RF Shielding

Receive: **RX1** or **RX2**

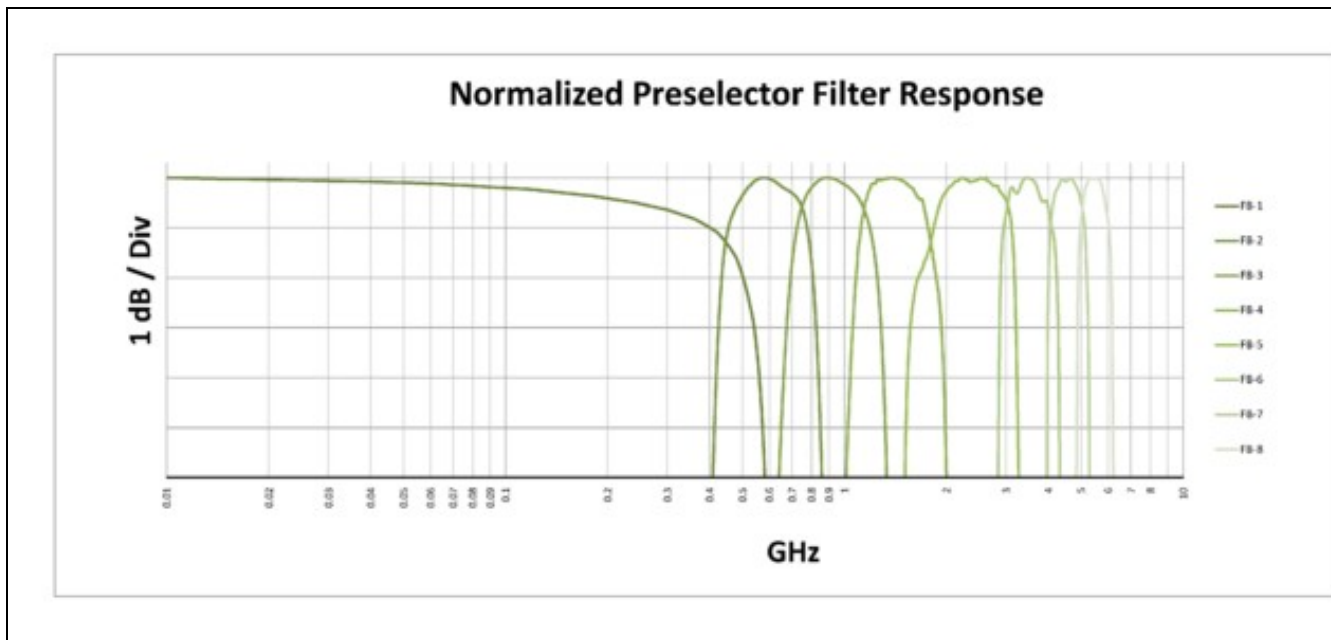
- Receive Gains Range: 0-93dB
- TwinRX: 80 MHz per channel ( 160 MHz total )
- **lo\_locked**: boolean for LO lock state

The TwinRX has six MMCX RF connectors on it.

Connector	Description	Min	Nominal	Damage
J1	LO2 Export	0 dBm	3 dBm	NA (Output)
J2	LO2 Input	0 dBm	2 dBm	20dBm
J3	LO1 Export	-12 dBm	5 dBm	NA (Output)
J4	LO1 Input	-10 dBm	-5 dBm	10dBm
J5	Antenna 1 connector RX 1			10 dBm
J6	Antenna 2 connector RX 2			10 dBm

#### Band Range

LB1	10 - 500 MHz
LB2	500 - 800 MHz
LB3	800 - 1.2 GHz
LB4	1.2 - 1.8 GHz
HB1	1.8 - 3.0 GHz
HB2	3.0 - 4.1 GHz
HB3	4.1 - 5.1 GHz
HB4	5.1 - 6.0 GHz



- Each preselector filter has a  $\pm 40$  MHz band overlap.

- 10MHz - 6GHz

Frequency	Preamp Enabled
10 MHz - 3 GHz	< 5
3 GHz - 5 GHz	< 4
5 GHz - 6 GHz	< 8

Frequency	Full Scale = -45 dBm	Full Scale = -30 dBm	Full Scale = -20 dBm
10 MHz - 1.8 GHz	-8	-2	16
1.8 GHz - 3 GHz	-10	-1	14
3 GHz - 6 GHz	-13	-1	12

Frequency Offset	0.9 GHz	2.4 GHz	5.8 GHz
10 kHz	-88	-86	-82
100 kHz	-105	-107	-103
1 MHz	-124	-127	-127

- All RF Ports are matched to 50 Ohm with -10dB or better return loss generally. Detailed test is pending.
- The maximum input power for the TwinRX is +10 dBm.
- Ettus Research recommends to always use the latest stable version of UHD. Minimum UHD version is 3.10.0.0

- Current Hardware Revision: 2
- Minimum version of UHD required: 3.10.0.0
- Minimum version of GNU Radio required: 3.7.10

- 0-40 °C

- 10% to 90% non-condensing

- X300 and X310 USRP only

- `master_clock_rate` of 200e6 only. The TwinRX daughterboard does not operate with the master clock rate of 184.32e6; operation with this master clock rate will result in UHD errors such as the following:

```
[ERROR] [DBMGR] The daughterboard manager encountered a recoverable error in init.
Loading the "unknown" daughterboard implementations to continue.
The daughterboard cannot operate until this error is resolved.
ValueError: TwinRX clock rate 92160000.000000 is not a multiple of the pfd freq 12500000.000000.
```

The TwinRX daughterboard is capable of phase-synchronous operation, and is recommended for phase-coherent applications. Please note that the TwinRX is receive-only. The SBX and UBX daughterboards are also recommended for phase-coherent applications, and are capable of both transmit and receive operations.

The TwinRX daughterboard is composed of two PCBs, the Intermediate Frequency (IF) board and the RF board. There are two functionally identical revisions of TwinRX at the moment. Please check the back of your TwinRX to see which you have. The schematics for TwinRX revision B should answer most possible questions. If you have any further questions please email [support@ettus.com](mailto:support@ettus.com).

TwinRX Revision A - (159685A-01)

- IF Board Revision B - (158671B-01L)
- RF Board Revision C - (156263C-01L)

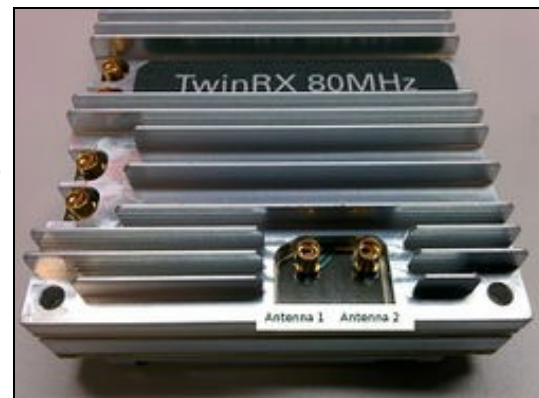
TwinRX Revision B - (159685B-01)

- IF Board Revision C - (158671C-01L)
- RF Board Revision D - (156263D-01L)

File:TwinRX IF Board Rev C.pdf

File:TwinRX RF Board Rev D.pdf

The antenna ports are MMCX connectors with 50 ohm input impedance. By default Antenna 1 (RX1) is routed to Channel 1 and Antenna 2 (RX2) to Channel 2. This routing can be changed to swap the antennas or to share a single antenna to both channels. The damage threshold for the antenna inputs is 10 dBm. In practice the available gain makes much lower input powers recommended for achieving the best dynamic range and noise figure.



Note: LO sharing cables are not required for a single TwinRX setup. LO sharing cables are only required with two TwinRX daughterboards in a single USRP X300/X310.

Connector	Description	Min	Nominal	Damage
J1	LO2 Export	0 dBm	3 dBm	NA (Output)
J2	LO2 Input	0 dBm	2 dBm	20dBm
J3	LO1 Export	-12 dBm	5 dBm	NA (Output)
J4	LO1 Input	-10 dBm	-5 dBm	10dBm
J5	Antenna 1 connector RX 1			10 dBm
J6	Antenna 2 connector RX 2			10 dBm



#### TwinRX (A Slot) TwinRX (B Slot)

J1 LO2 Export	J2 LO2 Input
J2 LO2 Input	J1 LO2 Export
J3 LO1 Export	J4 LO1 Input
J4 LO1 Input	J3 LO1 Export

The advanced functionalities of the TwinRX will be exposed through new functions implemented in [Multi-USRP](#).

The two channels of the TwinRX can be independently configured to use either of the two antenna ports, **RX1** and **RX2** using the standard antenna selection function in `multi_usrp`.

```
virtual void set_rx_antenna(const std::string &ant, size_t chan = 0) = 0;
```

Select the RX antenna on the frontend.

- **ant** the antenna name
- **chan** the channel index 0 to N-1

The TwinRX has two channels, **CH1** and **CH2** and each channel has two local oscillators, **LO1** and **LO2**. The local oscillators for a channel can be sourced from that channel's **internal** synthesizers, the **companion** channel's synthesizers, or **external** inputs. The value `multi_usrp::ALL_LOS` can be used to specify that the command be run on both synthesizers for a channel. The defaults are to operate on

```
virtual std::vector<std::string> get_rx_lo_names(size_t chan = 0) = 0;
```

Get a list of possible LO stage names

- **chan** the channel index 0 to N-1
- Returns a vector of strings for possible LO names

```
virtual void set_rx_lo_source(const std::string &src, const std::string &name = ALL_LOS, size_t chan = 0) = 0;
```

Set the LO source for the usrp device. For USRPs that support selectable LOs, this function allows switching between them. Supported options for source: internal, external, companion.

- **src** a string representing the LO source
- **name** the name of the LO stage to update
- **chan** the channel index 0 to N-1

```
virtual const std::string get_rx_lo_source(const std::string &name = ALL_LOS, size_t chan = 0) = 0;
```

Get the currently set LO source.

- **name** the name of the LO stage to query
- **chan** the channel index 0 to N-1
- Returns the configured LO source

```
virtual std::vector<std::string> get_rx_lo_sources(const std::string &name = ALL_LOS, size_t chan = 0) = 0;
```

Get a list of possible LO sources.

- **name** the name of the LO stage to query
- **chan** the channel index 0 to N-1
- Returns a vector of strings for possible settings

```
virtual double set_rx_lo_freq(double freq, const std::string &name, size_t chan = 0) = 0;
```

Set the RX LO frequency.

- **freq** the frequency to set the LO to
- **name** the name of the LO stage to update
- **chan** the channel index 0 to N-1
- Returns a coerced LO frequency

```
virtual double get_rx_lo_freq(const std::string &name, size_t chan = 0) = 0;
```

Get the current RX LO frequency.

- **name** the name of the LO stage to query
- **chan** the channel index 0 to N-1
- Returns the configured LO frequency

```
virtual freq_range_t get_rx_lo_freq_range(const std::string &name, size_t chan = 0) = 0;
```

Get the LO frequency range of the RX LO.

- **name** the name of the LO stage to query
- **chan** the channel index 0 to N-1
- Returns a frequency range object

```
virtual void set_rx_lo_export_enabled(bool enabled, const std::string &name = ALL_LOS, size_t chan = 0) = 0;
```

Set whether the LO used by the usrp device is exported For USRPs that support exportable LOs, this function configures if the LO used by chan is exported or not.

- **enabled** if true then export the LO
- **name** the name of the LO stage to update
- **chan** the channel index 0 to N-1 for the source channel

```
virtual bool get_rx_lo_export_enabled(const std::string &name = ALL_LOS, size_t chan = 0) = 0;
```

Returns true if the currently selected LO is being exported.

- **name** the name of the LO stage to query
- **chan** the channel index 0 to N-1

- [File:cu usrp twinrx cca.pdf](#)

As of December 1st, 2010 all Ettus Research products are RoHS compliant unless otherwise noted. More information can be found at <http://ettus.com/legal/rohs-information>

## Management Methods for Controlling Pollution Caused by Electronic Information Products Regulation

### Chinese Customers

National Instruments is in compliance with the Chinese policy on the Restriction of Hazardous Substances (RoHS) used in Electronic Information Products. For more information about the National Instruments China RoHS compliance, visit [ni.com/environment/rohs\\_china](http://ni.com/environment/rohs_china).

[TwinRX Letter of Volatility](#)

[FPGA Resources](#)

[UHD Stable Binaries](#)

[UHD Source Code on Github](#)