Comparative features list - B200/B210/B200mini

- Hardware Capabilities:
    - Integrated RF frontend (70 MHz - 6 GHz)
    - External PPS reference input
    - External 10 MHz reference input
    - Configurable clock rate
    - Variable analog bandwidth (200 kHz - 56 MHz)
    - GPIO header
    - [B200/B210] Internal GPSDO option (see **Internal GPSDO Application Notes (USRP-B2x0 Models)** for details)
    - [B210/B200mini] JTAG Connector
    - [B210] MICTOR Debug Connector
- FPGA Capabilities:
    - Timed commands in FPGA
    - Timed sampling in FPGA

Table of Contents

Power

In most cases, USB 3.0 bus power will be sufficient to power the device. If using USB 2.0 or a GPSDO, an external power supply or a cable designed to pull power from 2 USB ports (USB 3.0 dual A to micro-B or B) must be used.

Specifying a Non-standard Image

UHD software will automatically select the USRP B2X0 images from the installed images package. The image selection can be overridden with the `fpga` and `fw` device address parameters.

Example device address string representations to specify non-standard images:

```
fpga=usrp_b200_fpga.bin

-- OR --

fw=usrp_b200_fw.hex
```

Custom FPGA images and accessing user settings

The FPGA image is provided in source code and can thus be modified and rebuilt to serve custom purposes. For example, additional filtering or other DSP operations can be inserted into the FPGA before or after the DAC or ADC stages, respectively. Refer to the **FPGA Manual** for further information on how to rebuild the FPGA.

To control user-defined IP, the `USER_SETTINGS` parameter for the `radio_legacy` block in `b200_core.v` (for B210 and B200) or `b205_core.v` (for the mini series) need to be set to 1, and radio_legacy.v can then be modified to include user-defined read and write registers (the file includes examples on how to do that). Any customizations will most likely also be applied in radio_legacy.v.

For FPGA images that include user settings registers, UHD provides APIs to interface with those. To enable access to user settings, apply the `enable_user_regs` device argument on initialization. Then, use the uhd::multi_usrp::get_user_settings_iface API call to access an object that allows modifying the registers. Example:

```
auto usrp = multi_usrp::make("type=b200,enable_user_regs");
auto user_regs = usrp->get_user_settings_iface();
user_regs->poke32(addr, data);
```

Changing the Master Clock Rate

The master clock rate feeds the RF frontends and the DSP chains. Users may select non-default clock rates to achieve integer decimation rates or interpolations in the DSP chains. The clock rate can be set to any value between 5 MHz and 61.44 MHz (or 30.72 MHz for dual-channel mode). Note that rates above 56 MHz are possible, but not recommended.

The user can set the master clock rate through the usrp API call **uhd::usrp::multi_usrp::set_master_clock_rate()**, or the clock rate can be set through the device arguments, which many applications take:

```
uhd_usrp_probe --args="master_clock_rate=52e6"
```

The property to control the master clock rate is a double value, called `tick_rate`. Example:

```
import uhd
my_usrp = uhd.usrp.MultiUSRP("type=b200")
# The following two lines are synonymous:
my_usrp.set_master_clock_rate(16e6)
my_usrp.get_tree().access_double('/mboards/0/tick_rate').set(16e6)
# Note that if MCR changed, then all time-synchronization is lost!
```

Note that changing the master clock rate will reconfigure the entire clocking chain, thereby losing previous synchronization configurations.

Automatic Clock Rate Setting

The default clock rate setting is to automatically set a clock rate depending on the requested sampling rate. The automatic clock rate selection is disabled when either `master_clock_rate` is given in the device initialization arguments, or when **uhd::usrp::multi_usrp::set_master_clock_rate()** is called.

Note that the master clock rate must be an integer multiple of the sampling rate. If a master clock rate is chosen for which this condition does not hold, a warning will be displayed and a different sampling rate is used internally.

Nevertheless, there are multiple valid values for the master clock rate for most sampling rates. The auto clock rate selection attempts to use the largest possible clock rate as to enable as many half-band filters as possible. Expert users might have cases where a more fine-grained control over the resampling stages is required, in which case manually selecting a master clock rate might be more suitable than the automatic rate. Manual selection of master clock rate may also be required to synchronize multiple B200 units in time.

The property to dis- or enable the auto tick rate is a boolean value, `auto_tick_rate`. Example:

```
>>> import uhd
>>> my_usrp = uhd.usrp.MultiUSRP("type=b200")
>>> my_usrp.get_tree().access_bool('/mboards/0/auto_tick_rate').get()
True
>>> my_usrp.set_master_clock_rate(16e6) # This will lock the MCR
>>> my_usrp.get_tree().access_bool('/mboards/0/auto_tick_rate').get()
False
>>> U.set_rx_rate(7e6) # Not possible! We locked the MCR.
>>> U.get_rx_rate() # The actual rate is coerced:
8000000.0
>>> my_usrp.get_tree().access_bool('/mboards/0/auto_tick_rate').set(True)
>>> my_usrp.set_rx_rate(7e6) # This requires a new MCR...
>>> my_usrp.get_master_clock_rate() # ...which is chosen automatically.
56000000.0
```

RF Frontend Notes

The B200 features an integrated RF frontend.

Frontend tuning

The RF frontend has individually tunable receive and transmit chains. On the B200 and B200 mini, there is one transmit and one receive RF frontend. On the B210, both transmit and receive can be used in a MIMO configuration. For the MIMO case, both receive frontends share the RX LO, and both transmit frontends share the TX LO. Each LO is tunable between 50 MHz and 6 GHz.

Frontend gain

All frontends have individual analog gain controls. The receive frontends have 76 dB of available gain; and the transmit frontends have 89.8 dB of available gain. Gain settings are application specific, but it is recommended that users consider using at least half of the available gain to get reasonable dynamic range.

Frontend bandwidth

The analog frontend has a seamlessly adjustable bandwidth of 200 kHz to 56 MHz.

Generally, when requesting any possible master clock rate, UHD will automatically configure the analog filters to avoid any aliasing (RX) or out-of-band emissions whilst letting through the cleanest possible signal.

If you, however, happen to have a very strong interferer within half the master clock rate of your RX LO frequency, you might want to reduce this analog bandwidth. You can do so by calling uhd::usrp::multi_usrp::set_rx_bandwidth(bw).

The property to control the analog RX bandwidth is `bandwidth/value`.

UHD will not allow you to set bandwidths larger than your current master clock rate.

Power API

The B200 series support the UHD power calibration API (see: **Power Level Controls**). The TX path and the two RX paths have their own calibration data, resulting in 6 sets of calibration data total for the B210, and 3 for all the others.

Devices have to be manually calibrated using a calibrated power meter or signal generator.

Hardware Reference

LED Indicators

Below is a table of the B200/B210 LED indicators and their meanings:

Component ID	Description	Details
LED600	Power Indicator	off = no power applied (before rev6, some rev6) / external power applied and not yet initialized (most rev6) blue = USB power applied (before rev6) / power supplied (USB) OR power supplied (external) (rev6) red = external power applied (before rev6) / external power applied and device initialized (rev6)
LED800	Channel 2 RX2 Activity	off = no power applied green = receiving
LED801	Channel 2 TX/RX Activity	off = no activity green = receiving red = transmitting orange = switching between transmitting and receiving
LED802	Channel 1 TX/RX Activity	off = no activity green = receiving red = transmitting orange = switching between transmitting and receiving
LED803	Channel 1 RX2 Activity	off = no power applied green = receiving
LED100	GPS lock indicator	off = no lock green = lock

Below is a table of the B200mini LED indicators and their meanings:

Component ID	Description	Details

PWR LED	Power Indicator	off = no power applied on = power applied (external or USB)
TRX LED	TX/RX Activity	off = no activity green = receiving red = transmitting orange = switching between transmitting and receiving
RX2 LED	RX2 Activity	off = no activity green = receiving
S0 LED	Reference Lock	off = no activity green = locked
S1 LED	Reference Present	off = reference level low or not present green = reference level high

TX LED indicators are on when transmitting data and off when no samples are available to transmit. RX LED indicators are on when sending samples to the host and off when unable to do so. This means that TX/RX activity LED indicators will blink off in a temporary transmit underflow or receive overflow condition, indicating that the host is not sending or receiving samples fast enough. The host will be notified of the condition and output a "U" or "O" as well.

External Connections

Below is a table showing the B200/B210 external connections and respective power information:

Component ID	Description	Details
J601	External Power	6 V 3 A
J701	USB Connector	USB 3.0
J104	External PPS Input	1.8 V - 5 V
J101	GPS Antenna	GPSDO will supply nominal voltage to antenna.
J100	External 10 MHz Input	+15 dBm max
J800	RF B: TX/RX	TX power +20 dBm max RX power -15 dBm max
J802	RF B: RX2	RX power -15 dBm max
J803	RF A: RX2	RX power -15 dBm max

J801	RF A: TX/RX	TX power +20 dBm max RX power -15 dBm max

Below is a table showing the B200mini external connections and respective power information:

Component ID	Description	Details
USB3	USB Connector	USB 3.0
J1	TRX	TX power +20 dBm max RX power -15 dBm max
J2	RX2	RX power -15 dBm max
J3	External 10 MHz/PPS Reference	+15 dBm max

On-Board Connectors and Switches

Below is a table showing the B200/B210 on-board connectors and switches:

Component ID	Description	Details
J502[1]	Mictor Connector	Interface to FPGA for I/O and inspection.
J503[1]	JTAG Header	Interface to FPGA for programming and debugging.
J504[1,2]	GPIO Header	Header connected to the FPGA for GPIO purposes.
J400[3]	GPIO/UART Header	Header connected to the FPGA for GPIO purposes, or as a serial interface to the FX3 (requires custom FPGA bitfile)
S700	FX3 Hard Reset Switch	Resets the USB controller / System reset
U100	GPSDO socket	Interface to GPS disciplined reference oscillator

[1] Only on the B210

[2] Only since rev. 6 (green board). Default voltage is 3.3V. GPIO pinout is 1=GPIO_0, 2=GPIO_1, 3=GPIO_2, 4=GPIO_3, 5=GPIO_4, 6=GPIO_5, 7=GPIO_6, 8=GPIO_7, 9=GND, 10=GND

[3] Only since rev. 6 (green board). Default voltage is 1.8V. GPIO pinout is 1=GPIO_8, 2=GND, 3=GPIO_9. When compiling a bitfile to support a UART on this header, pinout is 1=TX, 2=GND,

3=RX.

Below is a table showing the B200mini on-board connectors and switches:

Component ID	Description	Details
J5	JTAG Header	Interface to FPGA for programming and debugging.
J6[1]	GPIO Header	Header connected to the FPGA for GPIO purposes.
SW1	FX3 Hard Reset Switch	Resets the USB controller / System reset

[1] GPIO pinout is 1=3.3V, 2=GPIO_0, 3=GPIO_1, 4=GPIO_2, 5=GPIO_3, 6=GND, 7=3.3V, 8=GPIO_4, 9=GPIO_5, 10=GPIO_6, 11=GPIO_7, 12=GND

Known issues

- When synchronizing multiple USRPs in time, it is strongly recommended to specify a master clock rate instead of relying on **Automatic Clock Rate Setting**.
- The B200 and B210 cannot support an external 10 MHz reference if a GPSDO is already present on the motherboard. If an external 10 MHz reference is to be used, the GPSDO needs to be physically removed from the device beforehand.
- The default streaming settings do not work optimally for all use cases. If there are issues with performance or stability, it can help to modify the `recv_frame_size` values, e.g., by setting `recv_frame_size=1024` as part of the device args.