

## **Intan Technologies RHD2000interface C++/Qt source code version 1.01 release notes**

21 March 2013

### **mainwindow.cpp:**

Added line 1908: `fifoPercentageFull = 0.0;`

This initializes a variable and fixes a problem that shows up on some computers only when running the program in demo mode (i.e., with no USB interface board attached).

## **Intan Technologies RHD2000interface C++/Qt source code version 1.1 release notes**

31 May 2013

Added 'Options' menu to allow users to save temperature sensor data from each chip. (Temperature readings are not displayed in the GUI, but can be plotted in MATLAB after recording data using the GUI.)

Modifications were made to the following files:

**mainwindow.cpp, mainwindow.h**  
**signalprocessor.cpp, signalprocessor.h,**  
**globalconstants.h**

The MATLAB file `read_Intan_RHD2000_file.m` was also updated to read version 1.1 saved data files containing temperature sensor data. (This new m-file will also read older version 1.0 data files.)

## **Intan Technologies RHD2000interface C++/Qt source code version 1.11 release notes**

11 June 2013

Fixed bug that was present since v1.0 and sometimes caused problems if the number of amplifier boards was changed and "Rescan Ports A-D" was clicked. Added lines 1658-1709 to `findConnectedAmplifiers()` in `mainwindow.cpp`.

## **Intan Technologies RHD2000interface C++/Qt source code version 1.2 release notes**

20 June 2013

Fixed bug that was present since v1.0 and sometimes caused problems routing amplifier channels to DAC outputs properly if chips were plugged into SPI ports B, C, or D.

Updated Rhythm C++ API (specifically, `rhd2000evalboard.cpp` and `rhd2000registers.cpp`) to support RHD2164 chip.

Extensive changes in **mainwindow.cpp** to support RHD2164 chip: notably, `findConnectedAmplifiers()`.

Extensive changes in **mainwindow.cpp** and **signalprocessor.cpp** to support multiple data file formats; added **setsaveformatdialog.cpp**. Added “Select File Format” button.

Extensive changes in **mainwindow.cpp** and **signalprocessor.cpp** to support triggered recordings; added **triggerrecorrdialog.cpp**. Added “Trigger” button.

Added check box in “DAC/Audio” tab to lock DAC 1 to currently selected channel (**mainwindow.cpp**).

Changes in **signalprocessor.cpp** to bundle many individual 32- and 16-bit integers into byte arrays and use `QDataStream::writeRawData()` to speed up writing to disk (in functions `loadAmplifierData()` and `saveBufferedData()`).

Changes in **signalprocessor.cpp** to improve running average of temperature sensor data.

### **Intan Technologies RHD2000interface C++/Qt source code version 1.3 release notes**

10 December 2013

Included **qtincludes.h** in all files that use Qt functions to aid in cross-platform compilation.

Changes in **mainwindow.cpp** to add low-latency threshold comparator functions and software/DAC high-pass filter options (in the Bandwidth tab and DAC/Audio tab). Added integer array **ttlOut** to make it easier to set digital output lines. Changed code to set **ttlOut[15]** high during data acquisition and to set **ttlOut[11-14]** high when amplifier boards are present on SPI Ports A-D to support future LEDs.

Changes in **signalprocessor.cpp** to add software high-pass filter capability.

Change in **waveplot.cpp** to draw an X over disabled channel plots.

Added several help dialogs accessible through “?” pushbuttons: **helpdialogchipfilters.cpp**, **helpdialogcomparators.cpp**, **helpdialogdacs.cpp**, **helpdialoghighpassfilter.cpp**, and **helpdialognotchfilter.cpp**.

### **Intan Technologies RHD2000interface C++/Qt source code version 1.4 release notes**

26 February 2014

Changes in **mainwindow.cpp** to add optional real-time control of fast settle function and auxiliary digital output pin of RHD2000 chips. Also added option to adjust SPI cable delay compensation manually since some RHD2164 board/cable combinations seem to require this. All of these changes appear in the “Configure” tab in the GUI.

Added several dialogs used in these new functions: **auxdigoutconfigdialog.cpp**, **helpdialogfastsettle.cpp**, and **cabledelaydialog.cpp**.

Updated Rhythm C++ API (specifically, **rhd2000evalboard.cpp** and **rhd2000registers.cpp**) to support real-time control of fast settle and auxiliary digital output pins.

### Intan Technologies RHD2000interface C++/Qt source code version 1.41 release notes

8 April 2014

Two minor bug fixes:

Added line 2472 in **mainwindow.cpp** to fix bug when a non-default data file format was selected. The “evaluation board mode” word was not being written to the info.rhd file, and this caused problems for users who wanted to load this file into Matlab using the Intan Matlab code.

In **waveplot.cpp**, modified toggleSelectedChannelEnable() to prevent users from enabling or disabling amplifier channels while recording is taking place. Added isRecording() to **mainwindow.cpp** to facilitate this functionality.

### Intan Technologies RHD2000interface C++/Qt source code version 1.4.2 release notes

9 July 2015

Making the windows resizable, which provides better support for higher resolution monitors. Changes in **mainwindow.cpp**, **setsaveformatdialog.cpp**, **spikeplot.cpp**, **spikeplot.h**, **spikescopedialog.cpp**, **triggerrecorddialog.cpp**, **waveplot.cpp**, **waveplot.h**.

In **mainWindow.cpp**, fixed a bug where saving impedances didn’t work for RHD2216 chips.

In **main.cpp**, added a Window-only ability to display a console with cerr output while running the GUI (commented out). If uncommented (and if **guicon.cpp** is compiled and linked in), this can be useful when debugging certain problems, especially problems where the software can’t contact the RHD2000 Evaluation Board. Added additional logging messages in **okFrontPanel.DLL.cpp**, **rhd2000evalboard.cpp**, **mainwindow.cpp**.

Several necessary enhancements for 64-bit support: fixed compile warnings in **okFrontPanelDLL.cpp** and **rhd2000registers.cpp** when compiling 64-bit, and distributing the 64-bit **okFrontPanel.dll**. Note that you we have only tested 64-bit Rhythm, not 64-bit RHD2000 interface GUI, so there may be other Qt-related issues that need to be resolved.

## Intan Technologies RHD2000interface C++/Qt source code version 1.5 release notes

3 May 2016

We added two forms of error checking to routines that read data from the USB port, to guard against rare, intermittent glitches that may corrupt or drop a few bytes of USB data during long recording sessions:

- (1) A small number of users reported USB interface board freeze-ups during long recording sessions. These freeze-ups seemed to happen randomly, with the software suddenly announcing that the FIFO buffer was full even though the CPU seemed to have no trouble keeping up with the data streaming over the USB interface. We determined that this was likely caused by corrupted data on a USB transfer. In **mainwindow.cpp**, `runInterfaceBoard()` regularly checks two `WireOuts` from the Opal Kelly board that contain the number of words in the FIFO buffer used to regulate USB data transfers (see **rhdevalboard.cpp**, `numWordsInFifo()`). It is important to stop the software just before the FIFO fills up, because it can “roll over” and send corrupted data otherwise. We added a static int *triggerEndCounter* to verify that `numWordsInFifo()` has returned a “nearly full” reading three times in a row before we stop data acquisition. This prevents one USB glitch from halting operation.
- (2) A small number of users reported occasional situations where amplifier channels seemed to shift during recording sessions, with channel 1 ending up at channel 14, for example, and channel 2 at channel 15, etc. We determined that this was caused by several bytes being dropped during USB data transfers over the `PipeOut`, which caused the data frames sent from the Opal Kelly board to become misaligned with the software. We added error-checking routines in the `readDataBlocks()` method in **rh2000evalboard.cpp** that check for the proper “magic number” at the beginning of each USB data frame and fix misalignments. The method `readAdditionalDataWords()` was added to support the new `readDataBlocks()`. We made the `checkUsbHeader()` method in **rh2000datablock.cpp** public instead of private to facilitate the error checking.

We expanded the functionality of triggered recordings to support unsupervised, episodic recording that starts and stops in response to a trigger signal. In **mainwindow.cpp**, we added the private bool *triggered*. We made changes to `runInterfaceBoard()`, including the addition of static int *triggerEndCounter*. We made extensive changes to **triggerrecorrdialog.cpp** to allow users to specify a posttrigger time, and to (optionally) automatically save the signal used for triggering. We modified `createSaveList()` in **signalprocessor.cpp** to support automatic saving of the trigger signal.