



808 Gilman Avenue Berkeley, CA., 94710 phone:925.253.2960
E-mail: support@ultraviewcorp.com Visit us at: www.ultraviewcorp.com

SYNTH1000x2,
And SYNTH1000x2SQRWAVE
PCIe Dual 50 kHz – 800 MHz Direct Digital Synthesizer

Covers Boards With Firmware rev 1.00 (4/9/14)
With software packages for 64-bit Windows 7 and CentOS 6.4

Product Specifications and User Manual

v1r06 – February 2, 2018

TABLE OF CONTENTS

TABLE OF CONTENTS.....	3
LIMITED WARRANTY.....	4
MODEL DESCRIPTIONS.....	5
MODEL: SYNTH1000x2.....	5
MODEL: SYNTH1000x2sqrwave.....	5
SPECIFICATIONS.....	6
Synthesizer outputs (Synth1000x2, Synth1000x2sqrwave).....	6
Phase Resolution.....	6
Profile select Inputs.....	7
General.....	7
Physical.....	8
HARDWARE ARCHITECTURE	9
Synthesizer A & B Outputs (DDS A OUT, DDS B OUT).....	9
TTL0-TTL5 PROFILE SELECT INPUT Lines.....	9
External REFERENCE clock INPUT LINE.....	9
External Ramp mode.....	9
LED INDICATORS.....	10
Enable acq LED (frequency locking enabled indicator).....	10
DMA LED.....	10
Acquiring LED.....	10
Mode LED.....	10
Clock bad LED.....	10
Power bad LED.....	11
Mode B LED.....	11
HARDWARE INSTALLATION AND SETUP.....	12
WINDOWS 7™ SOFTWARE SETUP.....	12
Software Package Contents.....	12
Windows 7™ Installation.....	13
The Example GUI Programs.....	13
The Example Command Line Programs.....	13
Power Settings.....	13
Additional Driver Considerations.....	14
RUNNING SYNTH1000 PROGRAMS.....	15
USING THE SYNTH1000X2 LABVIEW GUI	16
Example #1 – Two independent sinusoids – Profile Mode.....	17
Example #2 – FREQUENCY-locked sinusoids – Profile Mode.....	18
Example #3 – Two fully independent swept sinusoids – sweep mode.....	19
Example #4 – External Modulation example – external profile mode.....	20
USING THE QT GUI UNDER WINDOWS.....	21
SOFTWARE INSTALLATION AND USE UNDER CENTOS 6.....	22
CUSTOM APPLICATION PROGRAMMING OF THE SYNTH1000 IN WINDOWS.....	23
Synth1000 Operational Overview.....	23
“Update Event”.....	23
Using the uvAPI Object	24
Instantiating a uvAPI Object.....	24
uvAPI Object Member Functions.....	24
LOW LEVEL SOFTWARE INTERFACE (MOST USERS NEED NOT READ).....	27

LIMITED WARRANTY

Ultraview Corporation hardware products are warranted against defects in materials and workmanship for a period of two (2) years from the date of shipment of the product. During the warranty period, Ultraview Corporation shall, at its option, either repair or replace hardware, software or firmware products which prove to be defective. This limited warranty does not cover wear on the top PCIe connector and damage caused by misuse or abuse by customer, and specifically excludes damage caused by the application of excessive voltages or currents to any part of the board. Also specifically excluded is ESD damage to the bus switches, so all users are advised to use wrist straps and other ESD protection while inserting or removing boards in any system.

While Ultraview Corporation hardware, software and firmware products are designed to function in a reliable manner, Ultraview Corporation does not warrant that the operation of the hardware, software or firmware products will be uninterrupted or error free. Ultraview products are not intended to be used as critical components in life support systems, aircraft, military systems or other systems whose failure to perform can reasonably be expected to cause significant injury to humans. Ultraview expressly disclaims liability for loss of profits and other consequential damages caused by the failure of any product which would cause interruption of work or loss of profits, such as shipboard or military attachment.

THIS LIMITED WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED. THE WARRANTIES PROVIDED HEREIN ARE BUYER'S SOLE REMEDIES. IN NO EVENT SHALL ULTRAVIEW CORPORATION BE LIABLE FOR DIRECT, SPECIAL, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES SUFFERED OR INCURRED AS A RESULT OF THE USE OF, OR INABILITY TO USE THESE PRODUCTS. THIS LIMITATION OF LIABILITY REMAINS IN FORCE EVEN IF ULTRAVIEW CORPORATION IS INFORMED OF THE POSSIBILITY OF SUCH DAMAGES.

Some states do not allow the exclusion or limitation on incidental or consequential damages, so the above limitation and exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

MODEL DESCRIPTIONS

The SYNTH1000 series of Direct Digital Synthesizer (DDS) boards contain two complete low noise precision waveform synthesizers on a single PCIe bus card. Each channel's frequency may be programmed from 50KHz to 800MHz (with reduced performance to 1000MHz). Amplitude may be independently programmed from 0 to 800mV p-p into 50 ohms. Designed for low jitter operation in communications, industrial and scientific applications, these boards function in PCIe bus systems using supplied drivers for 64-bit Windows 7™ or CentOS 6. In addition to dual synthesizers, the SYNTH1000x2 board has three external TTL inputs for each synthesizer channel, which can be used to allow external equipment to dynamically switch each synthesizer between up to eight frequency profiles, for a variety of dynamic waveform synthesis applications (frequency/phase/amplitude modulation, high speed quadrature modulation, triggered frequency sweeping, etc.). The TTL inputs can also be used to switch ramping direction, or pausing the ramp.

Both synthesizer channels are driven from a common 3.5GHz CRO timebase, frequency locked to a sub-1ppm-accurate TCXO, for precise channel-to-channel frequency tracking.

All models have two independent output channels, each driven by the filtered output of a dedicated Analog Devices AD9914 DDS IC, and include additional firmware to allow advanced functionality such as software selectable internal or externally triggered profile switching capability. Further information for each model is given below.

MODEL: SYNTH1000X2

Model Synth1000x2 is a Dual-channel DDS board with two independent Direct Digital Synthesizers, each separately programmable for any frequency between 50kHz and 800MHz, in increments of approximately 0.816 Hz. The two channels may each be independently programmed for frequency, amplitude and phase, or may be externally controlled via three TTL inputs (PS0-PS2) to dynamically select up to eight frequency/phase profiles per synthesizer, via hardware control. The two synthesizer outputs are conveniently provided by SMA output connectors.

MODEL: SYNTH1000X2SQRWAVE

Model Synth1000x2SQRWAVE is similar to the Synth1000x2 model, but instead output a 700mV peak to peak square wave on each channel, instead of a sine wave. Unlike the sine wave outputs whose frequency may be programmed from 50KHz to 800MHz (with reduced performance to 1000MHz), the square wave outputs are limited to 1MHz to 1000MHz. All other functionality is equivalent.

SPECIFICATIONS

SYNTHESIZER OUTPUTS (SYNTH1000X2, SYNTH1000X2SQRWAVE)

Number of Output Channels:	2 independently programmable analog outputs
Analog output voltage (into 50 ohms):	446mV RMS (6dBm) +/-10% from 2-500MHz 446mV RMS +/-20% from 50KHz-800MHz 446mV RMS +/-30% from 10KHz to 1GHz
Output impedance:	50 ohms 10pF
Output connectors:	SMA connectors.
Analog Output Specifications (preliminary):	
Wideband SFDR (DC to Nyquist):	
40MHz Fout	55dB
100MHz Fout	65dB
400MHz Fout	60dB
Narrowband SFDR (over 10MHz frequency span):	
20 MHz Fout	>80dB
100MHz Fout	>85dB
200MHz Fout	>95dB
400MHz Fout	>95dB
400MHz Fout	>95dB
Harmonic Distortion:	
100KHz to 400MHz):	-50dB or better
Absolute frequency accuracy, after 10 minute warm-up:	+/-0.000001 (1ppM) of specified freq. +/-0.816Hz
Software profile switching time:	Approx 1 microsecond, system dependent
Externally controlled profile switching time:	Under 30 nanoseconds.

PHASE RESOLUTION

Each DDS has 16-bit phase resolution, allowing fine phase adjustment of each DDS core. Further, the Synth1000x2 boards incorporate dedicated circuitry and firmware to frequency synchronize the two DDS cores to a common 3.5GHz master timebase.

PROFILE SELECT INPUTS

Profile selection may optionally be externally controlled for rapid frequency and/or phase and/or/amplitude switching.

Connector: 26 pin serial external TTL profile select and ramp control, with external boardlet with 8 SMA input connectors.

Number of TTL Input lines: 6, Standard TTL ($V_{il} < 0.8V$, $V_{ih} > 2.4V$). Do not exceed +4V on any of these inputs.

GENERAL

Operating Temperature Range: 0 to +55 Degrees Celsius

Storage Temperature Range: -25 to +85 Degrees Celsius

Power Requirements (from PCIe bus) +3.3V +/-5% at 2A Maximum
+12V +/-5% at 0.5A Max.
<10W Total power consumption

PHYSICAL

SYNTH1000 boards are full size PCIe bus boards designed to be inserted into a PCIe x16 (physical) slot, but only requires x8 PCIe connectivity. It also can be installed into PCIe x8 open-ended PCIe sockets, provided there is enough vertical room between boards. Due to the protrusion of its cooling fan, the SYNTH1000 requires a blank slot between it and the neighboring board to its component side. The figure below shows the locations of the SMA signal output and digital I/O connectors, and LED indicators.

To avoid overheating, all SYNTH1000x2 boards must be installed in a well-cooled workstation, PC or server chassis.



Figure 1. SYNTH1000x2 PCIe Dual 50KHz to 800MHz Direct Digital Synthesizer Board

HARDWARE ARCHITECTURE

SYNTH1000 series boards are comprised of a digital section and an analog section. The digital section includes a high speed programmable logic device which implements the PCI interface and embedded firmware for controlling the DDS cores of the analog section. The analog section contains a 3.5GHz low-jitter clock, and two AD9914 single-chip Direct Digital Synthesizer (DDS) ICs. The current outputs of each DDS are fed, via a broadband RF transformer and a 900MHz low-pass filter, to two SMA jacks.

SYNTHESIZER A & B OUTPUTS (DDS A OUT, DDS B OUT)

The two analog synthesizer outputs are provided on SMA connectors with amplitude (nominally 0-450mV RMS into 50 ohms). The frequency, phase, amplitude, frequency sweep rate (if specified), and other parameters are software programmable.

TTL0-TTL5 PROFILE SELECT INPUT LINES

The external TTL input pod is connected to the Synth1000x2 board via a 26-pin cable, and the interface is used for external profile selection inputs in external profile mode and as ramping controls in external ramp mode. TTL 0-2 control profiles for SynthA and TTL 3-5 control profiles for SynthB on the board.

EXTERNAL REFERENCE CLOCK INPUT LINE

The SMA jack labeled GC_P on the external TTL input pod may be optionally used for an external clock input, if desired. This external clock, which may be any frequency between 10 and 50MHz can make it easier to sync multiple synth1000x2 devices together in a multi-board system. The labview interface currently supports a 10mhz external clock, if another frequency is required additional steps are needed.

EXTERNAL TTL RAMP/OSK MODE

The external TTL input pod can also function to control the sweep. External Ramp mode enables hardware ramp direction change (on TTL 0,3 for synth A/B), hardware ramp pause (on TTL 1,4 for synth A/B) and OSK (on TTL 2,5 for synth A/B) when external OSK is enabled. Enabling external OSK disables the output unless a high signal is present on TTL2/5.

LED INDICATORS

There are seven Red/Green LEDs on the top edge of the SYNTH1000x2 board that are useful during system integration for monitoring the board status. The functions of the LEDs are outlined below.

ENABLE ACQ LED (FREQUENCY LOCKING ENABLED INDICATOR)

This LED displays whether the frequency locking loop is enabled.

Green: CRO (3.5GHz Coaxial Resonant Oscillator) is successfully locked to <1ppm TCXO reference or external reference clock.

OFF: CRO is not successfully locked to a TCXO or external reference frequency. Usually this condition occurs during warm-up.

DMA LED

This LED indicates the FLL (Frequency Locking Loop) DAC has reached its low limit without locking. This is an error condition.

Green: DAC at zero, tuning direction for lock is down.

OFF: DAC not at zero.

ACQUIRING LED

This LED indicates the FLL (Frequency Locking Loop) DAC has reached its high limit without locking. This is an error condition.

Green: DAC at Full Scale, tuning direction for lock is up.

OFF: DAC not at Full Scale.

MODE LED

This LED displays whether the device is to be locked to internal clock or external clock.

Green: Set up to be locked to internal clock

Off: Set up to be locked to external clock

CLOCK BAD LED

This LED displays the current status of the 3.5GHz internal clock:

Off: Internal clock locked to high accuracy reference oscillator.

Red: Internal clock not locked correctly. This means one of three things:

- 1) Not Locked.

- 2) Missing Reference Clock.
- 3) Missing Output from SynthA (CRO/24).

POWER BAD LED

This LED displays the status of the power supply to the board:

OFF: Board within normal operating parameters.

Red: Board is outside of specs, check power supply levels.

MODE B LED

This LED displays the status of the board loading sequence.

Green: FPGA loaded correctly.

Off: Board did not power up correctly. Shut down system and then power on again.

HARDWARE INSTALLATION AND SETUP

To avoid overheating, the SYNTH1000x2 must be installed in a well-cooled workstation or server chassis, or alternatively in an industrial chassis PC. Installation in a standard desktop PC without fans at the front end of the card cage may cause the SYNTH1000x2 to overheat, and resulting damage is not covered by warranty.

1. Use the shutdown command on your system and then turn OFF the power to the system.

BEFORE REMOVING THE COMPUTER SYSTEM COVER OR REMOVING ANY BOARD, BE SURE THAT THE POWER TO THE COMPUTER, AS WELL AS TO ALL PERIPHERAL DEVICES IS OFF. WEAR A STATIC-DISSIPATING WRISTBAND WHICH IS GROUNDED TO THE SYSTEM CHASSIS WHILE OPENING OR WORKING ON YOUR SYSTEM.

2. Remove any screws that attach the computer system cover and remove the cover.
3. Remove the filler bracket (if applicable) from the PCIe bus slot into which you wish to install your SYNTH1000 board. For details, refer to the hardware manual for your computer system.
4. Hold the SYNTH1000 board by the top of the metal PCIe bracket. Then hook the tab on the bottom edge of the SYNTH1000's metal bracket into the corresponding slot in the computer's rear panel. Carefully push the SYNTH1000 down so its PCIe bus connector mates with the PCIe bus connector on the motherboard. Be sure that the SYNTH1000 is seated firmly into the motherboard PCIe bus connector. Check that no other boards have become unseated when the SYNTH1000 was installed, as motherboards may flex slightly when installing PCIe boards.
5. Plug coaxial cables for the analog outputs into the appropriate SMA connectors on the SYNTH1000's rear bracket at the rear of the system. Please refer to the diagram on page 8 of this manual.
6. We recommend that synthesizer output channels A and B (the 3rd and 4th SMA connectors from the bracket top) initially be connected to an oscilloscope set for 200mV/div., so you can observe the two synthesized waveforms operation when running the example software.
7. Replace the computer system cover, installing all screws you had removed. Reconnect the power cables to the system and peripherals.
8. Power up and reboot the system. The system will then be ready for software installation.

WINDOWS 7™ SOFTWARE SETUP

SOFTWARE PACKAGE CONTENTS

The software for the Synth1000 series board includes 64-bit Windows device drivers supported for Windows 7™ and later operating systems, a ready-to-use NI Labview GUI for simple synthesizer applications, a ready to use Qt GUI for simple synthesizer applications, a command line C program showing how to implement software loops for advanced functionality, a command line C program suitable for batch file access, and several Microsoft Visual Studio 2012 projects containing example code necessary to allow rapid development of powerful custom applications. This software package can be downloaded from:

<http://www.ultraviewcorp.com/downloads>

WINDOWS 7™ INSTALLATION

Run setup.bat **AS ADMINISTRATOR** (right click) in the base folder and follow the prompts. First the user is asked to install the driver for the synth. Then the user is asked to install MSVC redistributable. These are both required for basic synth operation if not already installed. Then Qt and the Labview runtime redistributable should be installed from their respective web locations (<http://www.ni.com/download/labview-run-time-engine-2014/4889/en/> and http://download.qt-project.org/archive/qt/5.1/5.1.1/qt-windows-opensource-5.1.1-msvc2012_opengl-x86_64-offline.exe). These are required only if the user plans to use the associated GUIs and/or those environments for custom applications. It is recommended to use at least one of the GUIs to get a feel for the Synth's functionality. It is best to restart after the installation process has completed, and the synth will initialize automatically on the next startup. If you do not wish the synth to start when windows does, simply remove startup.exe from the startup folder (start menu, all programs, startup). The synth will still start when a program is executed, but the clock may take longer to sync. Two shortcuts, one to the main synth folder, and the other to the labview executable are placed on the desktop for easy access.

THE EXAMPLE GUI PROGRAMS

The source code and project files for the example GUI programs are contained in two folders: "QtGUI" for Qt, "LabviewGUI" for labview. There is both a Labview based and a Qt based GUI project and executable. The executables are located in the release and build folders respectively. Both can be run as they are, or modified with the appropriate development environment. Qt is freely available on the net, but requires an additional compiler. Labview must be licensed, although the example LabView executable may be run unmodified, in any system in which the free Labview 2014 Run-Time-Engine has been installed.

THE EXAMPLE COMMAND LINE PROGRAMS

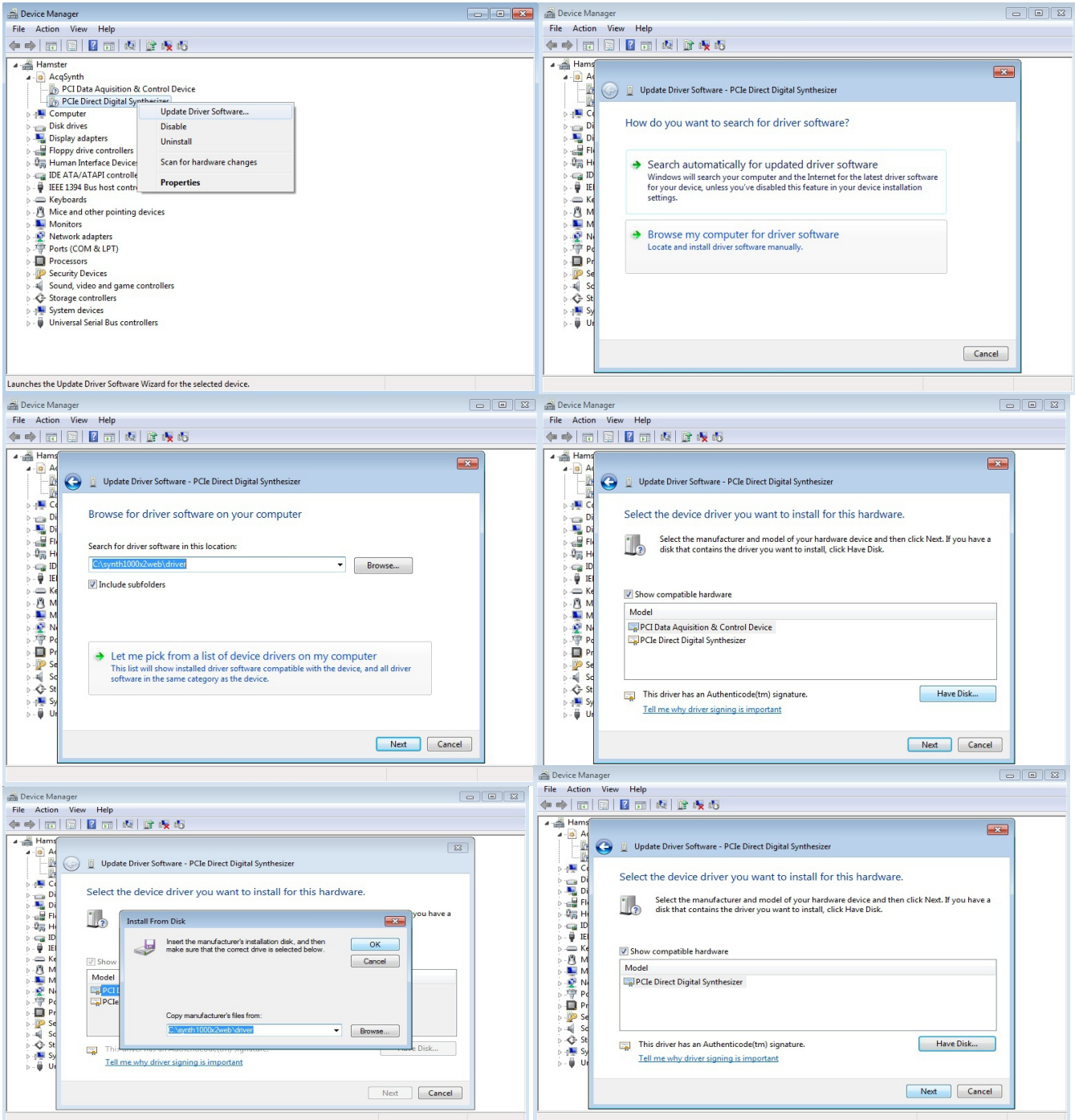
The C source code and project files for the example program are contained in the folders titled "command line examples\command_line_utilities\acquire" and "command line simple control\command_line_utilities\acquire". The project files are associated with Microsoft Visual Studio version 12.0. Each project builds a simple Windows console application (acquire.exe). One of the programs is ready to use simply by executing it and performs examples of common uses of the synths. The other requires several command line arguments and provides easy access to basic synth functionality that can be called from a batch file or from other programs. Modification of either of these programs can be achieved by editing acquire.cpp and rebuilding the application. Additional information is contained in the C code comments.

POWER SETTINGS

Please turn off computer sleep mode in control panel, power settings. The synth1000x2 does not support hibernation/sleep and has unexpected behavior when recovering from sleep mode.

ADDITIONAL DRIVER CONSIDERATIONS

If using both a synth1000x2 and an ultraview data acquisition board, additional steps may be required to get the drivers correctly loaded. Both products are similar and as such windows may want to install the wrong driver for one of the products. The product with the wrong driver will need to have a manual install of the driver as follows: Update, Browse, Pick, Have Disk, Okay, Next.



RUNNING SYNTH1000 PROGRAMS

The labview based GUI provided (Synth1000x2.exe) allows the user to specify each of the frequency and phase profiles for each synthesizer, and select the active profiles easily. Additionally, frequency sweeping capabilities can be enabled and specified, and the board can be set for external modulation mode. This program will allow the user to easily access most, but not all, of the features of the Synth1000 board. Additional functionality can be programmed with a Labview development environment. The board is set up automatically when the program loads, and should (with the startup program running when windows loads) immediately be locked to the extremely precise secondary clock and be ready for use. If there is considerable air flow or temperature fluctuation near the board, it may unlock temporarily, but should quickly lock again. While unlocked, the board will still function correctly, but the inaccuracy of the exact clock frequency delivered may be more than 1 part per million.

The Qt based GUI provided (qtGUI.exe) also allows the user to specify each of the frequency and phase profiles for each synthesizer, and select the active profiles easily. The board is setup automatically when the program is run. External modulation mode is also available in the Qt GUI with a checkbox in the bottom left corner. Additional functionality can be programmed with the included source files and the free for download Qt development environment at qt-project.org/downloads. Qt also requires a compiler on the system.

The command line programs can be accessed by opening a command prompt and running the executables in the folders "command line examples\command_line_utilities\acquire" and "command line simple control\command_line_utilities\acquire". The first program simply demonstrates some of the features of the synths and should be viewed with an oscilloscope on several different time, channel, and triggering settings. The second requires that you pass arguments to the program. It can be very quickly setup to work from a user created batch file to call the program several times with different arguments that even a non-programmer can put together. Frequency, amplitude, phase, and profile settings can all be passed in to allow anyone to setup the board with their own settings. The setup argument should be called once, and external select can also be called after all the profile information is set. The included batch file will need to be edited to include your unique serial number as the first parameter passed. Sweep modes are not enabled in the command line by default for simplicity, but they can be added if needed. Loop statements are demonstrated in the example, which are mostly equivalent to the sweep modes as long as a very high level of precision is not needed. Other programs can also call this program to pass in settings to the board.

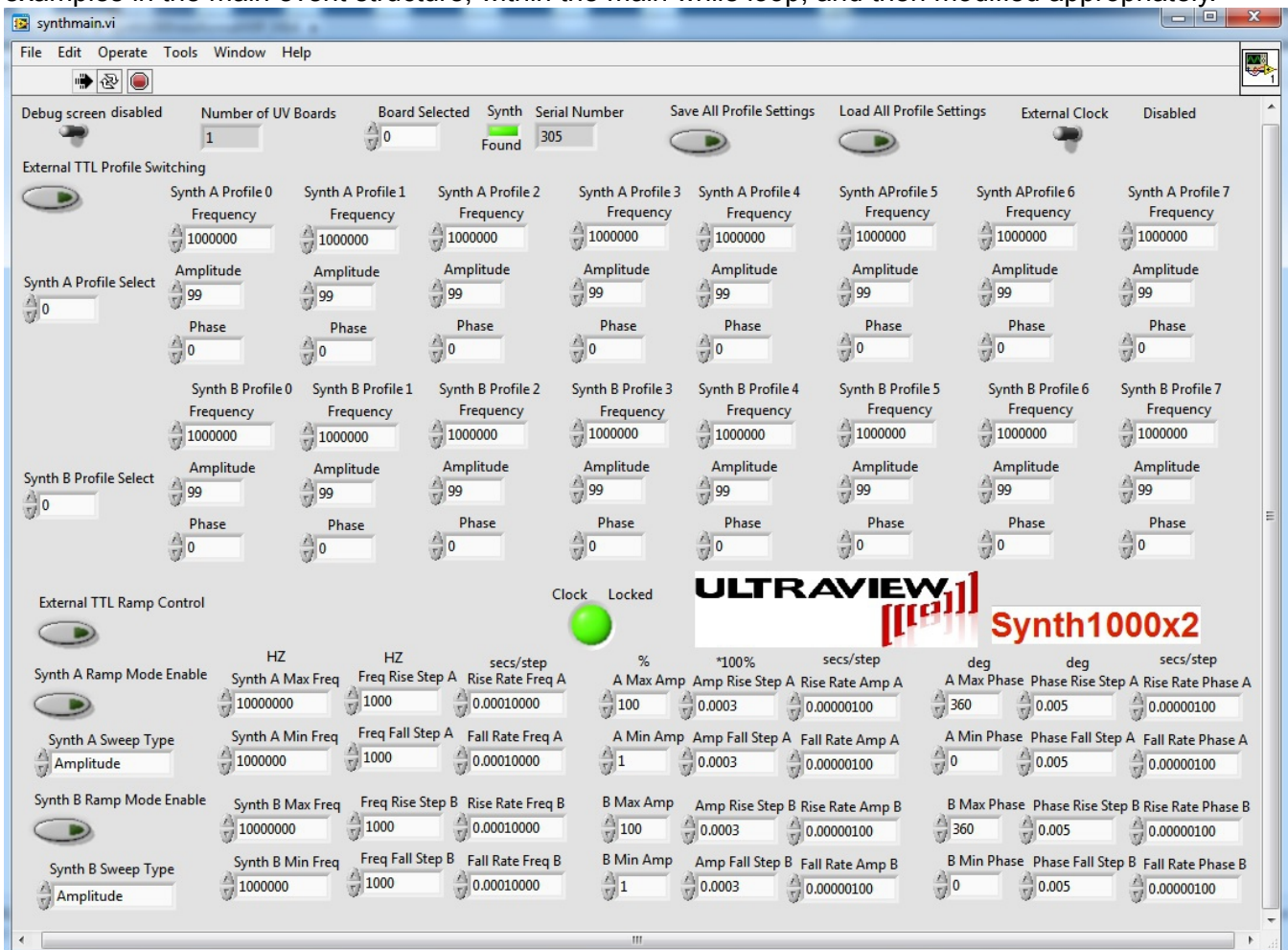
If a higher level of control is desired, the user may modify the C code provided in the example program directories which contains a MS Visual Studio project (acquire.sln) containing all files necessary to create custom user programs. The user library is provided in simple to use function calls, and the compiled driver is supplied. Please refer to page 23 of this document for further detail.

USING THE SYNTH1000X2 LABVIEW GUI

The Synth1000 Labview GUI has all the functionality commonly required by most users. In profile mode, the profiles can be individually selected by software, or alternatively externally toggled by an input TTL signal through the serial interface. The frequency, amplitude, and phase of each profile can be individually modified to suit user requirements. Toggling the external profile mode enable switch will change the profiles rapidly based on TTL input. In external profile mode, each individual profile can still be modified and the changes will happen at the next clock cycle. Sweep mode allows frequencies, amplitudes, or phases to be swept through at a user defined step rate. Sweep mode must be disabled to return to profile mode. External ramp control allows TTL input to pause the sweep, or switch the direction of the sweep. The external clock select in the upper right disables the internal clock and accepts a 10mhz external clock input on the GC_P SMA connector on the TTL input pod. The debug switch in the upper left enables rarely needed monitors below the main window.

When working with multiple boards in a system, each time the board is switched a temporary ini file with all settings is saved/loaded. These settings can be manually copied to the save/load settings ini file if desired. The manual save and load buttons work for either a single board, or multiple boards.

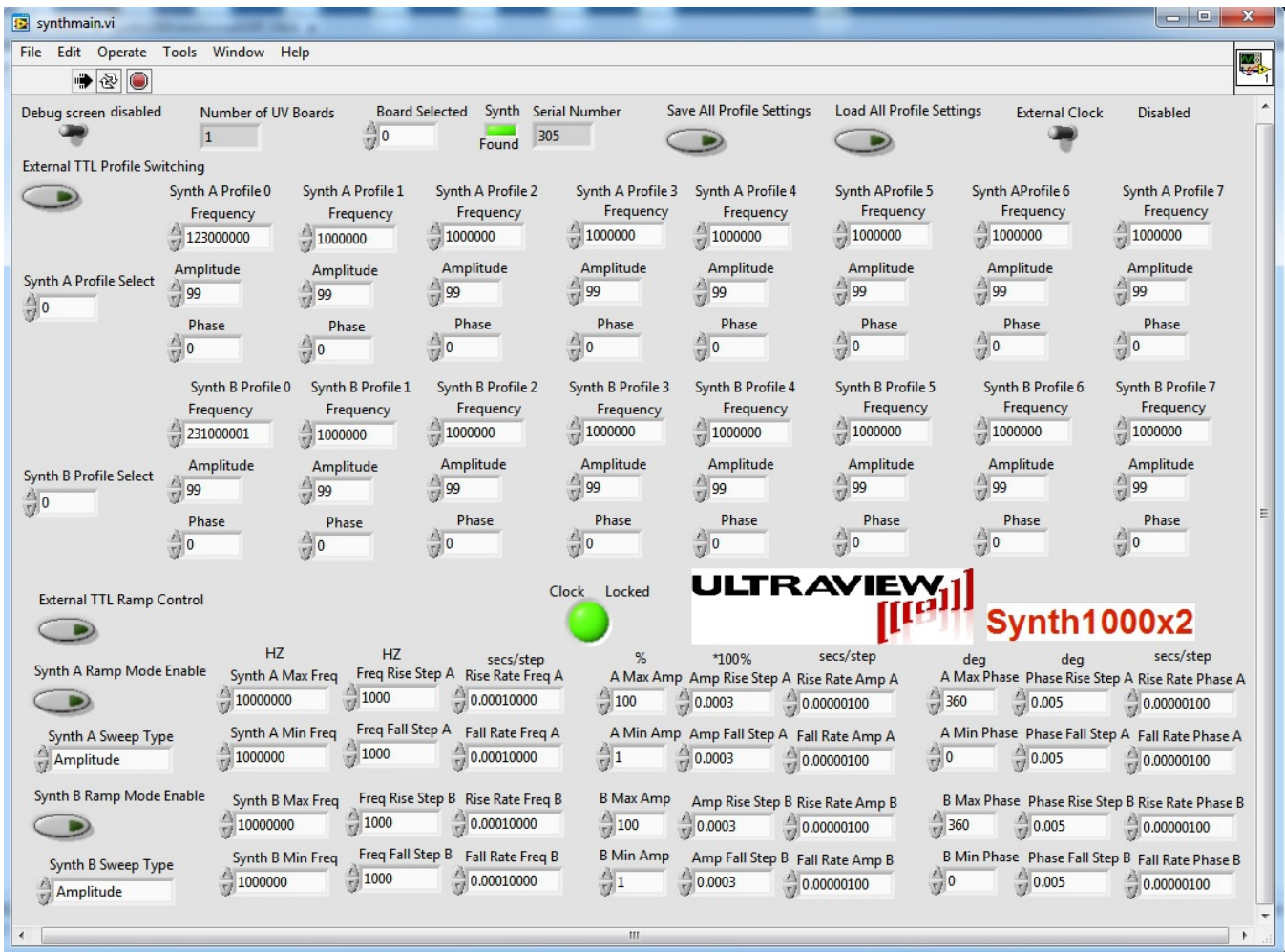
The labview project file synth1000x2.lvproj can be opened, edited, and recompiled for custom functionality to suit every application. Additional calls to the DLL can be set up by looking at the examples in the main event structure, within the main while loop, and then modified appropriately.



A few examples of common applications follow:

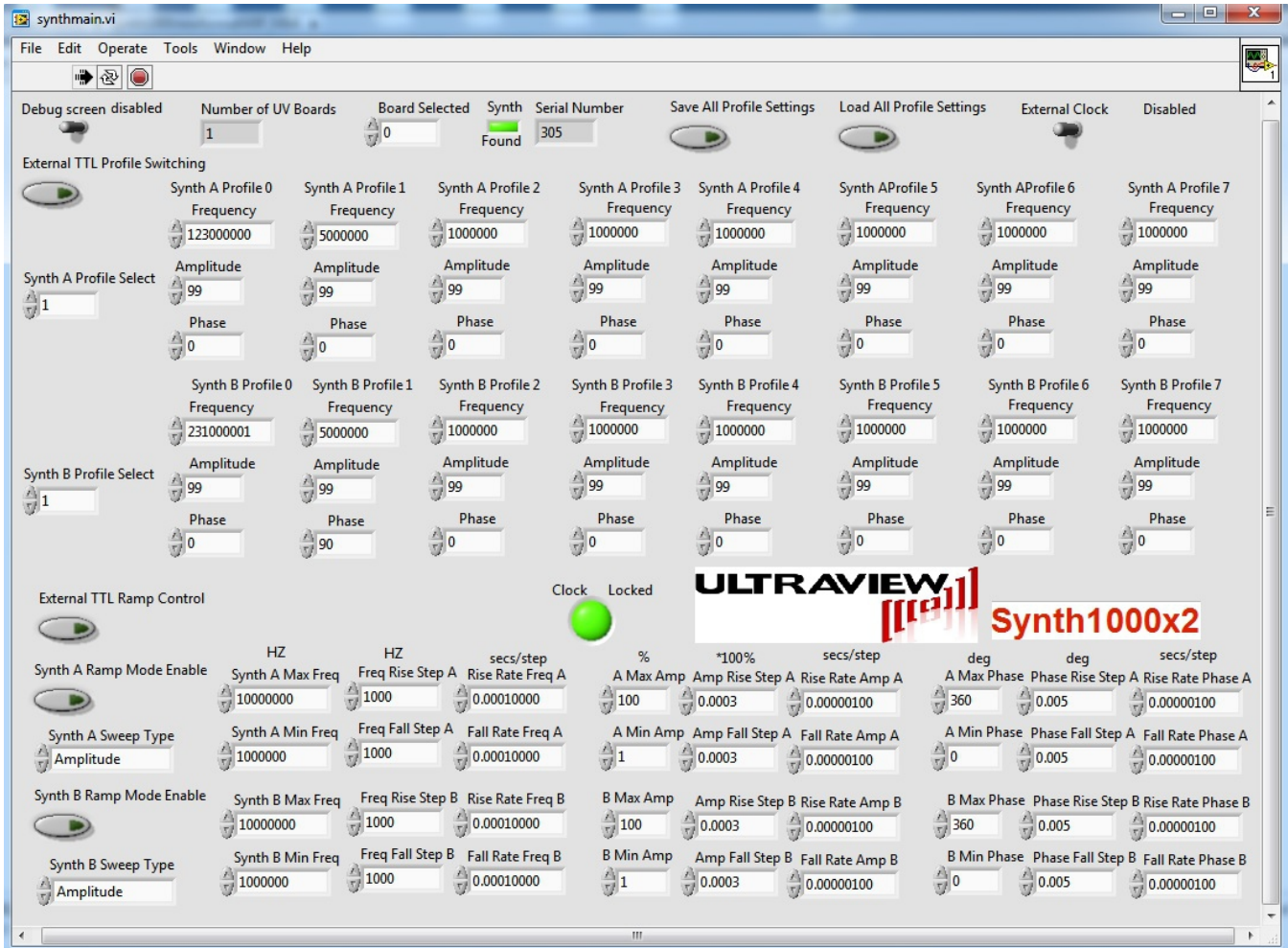
EXAMPLE #1 – TWO INDEPENDENT SINUSOIDS – PROFILE MODE

123.000000 MHz is generated on Synth A, and 231.000001MHz on Synth B. Notice that Profile “0” is selected on each synthesizer by use of the selection box. Press Enter or choose a new selection box to accept these settings.



EXAMPLE #2 – FREQUENCY-LOCKED SINUSOIDS – PROFILE MODE

5.000000 MHz is generated on Synth A, and 5.000000 MHz on Synth B with a 90 degree phase offset. Profile 1 was used for both synthesizers, while retaining the Profile 0 settings. Each synthesizer has eight independent frequency, amplitude, and phase profiles.

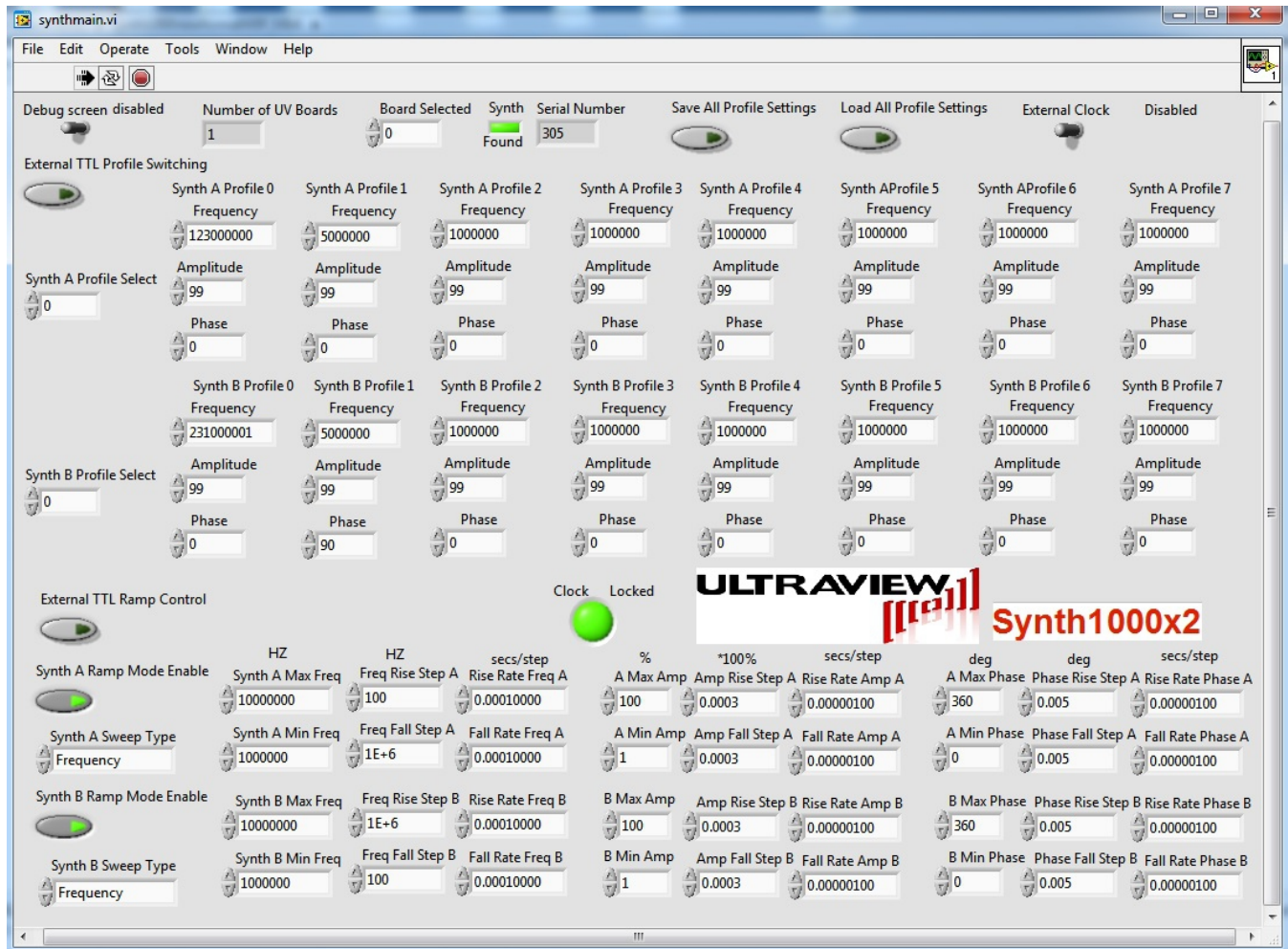


EXAMPLE #3 – TWO FULLY INDEPENDENT SWEEP SINUSOIDS – SWEEP MODE

Synthesize two swept sinusoids, one with increasing frequency and the other with decreasing frequency.

- 1.000000 MHz from Synth A, increasing at 1Mhz/sec (Sweep Rate = $1e+002$ hz/step * 10k steps/sec) for 9 seconds, then repeating (fall time nearly instant)
- 10.000000MHz from Synth B, decreasing at 1MHz/sec (Sweep Rate = $1e+002$ hz/step * 10k steps/sec) for 9 seconds, then repeating (rise time nearly instant)

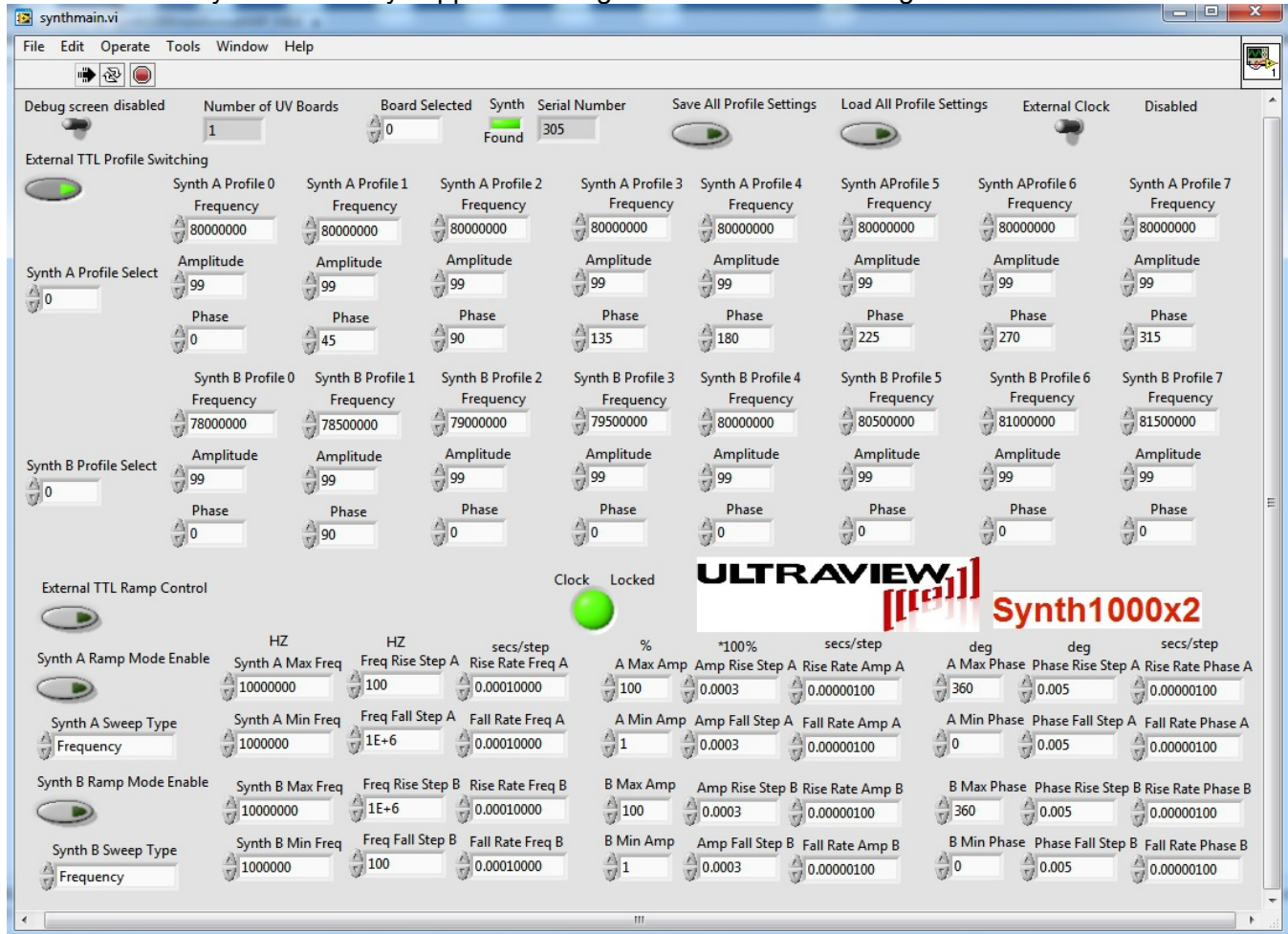
Notice the “SynthA sweep mode enable” and “SynthB sweep mode enable” switches are both enabled, but either can function individually.



EXAMPLE #4 – EXTERNAL MODULATION EXAMPLE – EXTERNAL PROFILE MODE

Externally modulated QPSK (8-PSK Phase Shift Key) signal (IF = 80MHz) on Synthesizer A. Externally modulated 8-FSK signal (f = 78, 78.5, 79, 79.5, 80, 80.5, 81, and 81.5MHz) on Synthesizer B. In this case the eight profiles are set for the corresponding desired frequency/phase relationships for the two synthesizers.

Enabling external modulation disables the user control of the profile selection, as profile selection is now controlled by the externally supplied TTL signals on PS0-PS2 through the serial interface.



If the external profile mode switch is enabled the TTL signals on PS0-PS2 will control the selection of frequency/phase profiles for each of the synthesizers (positive logic: TTL_High = logic '1', TTL_Low = logic '0').

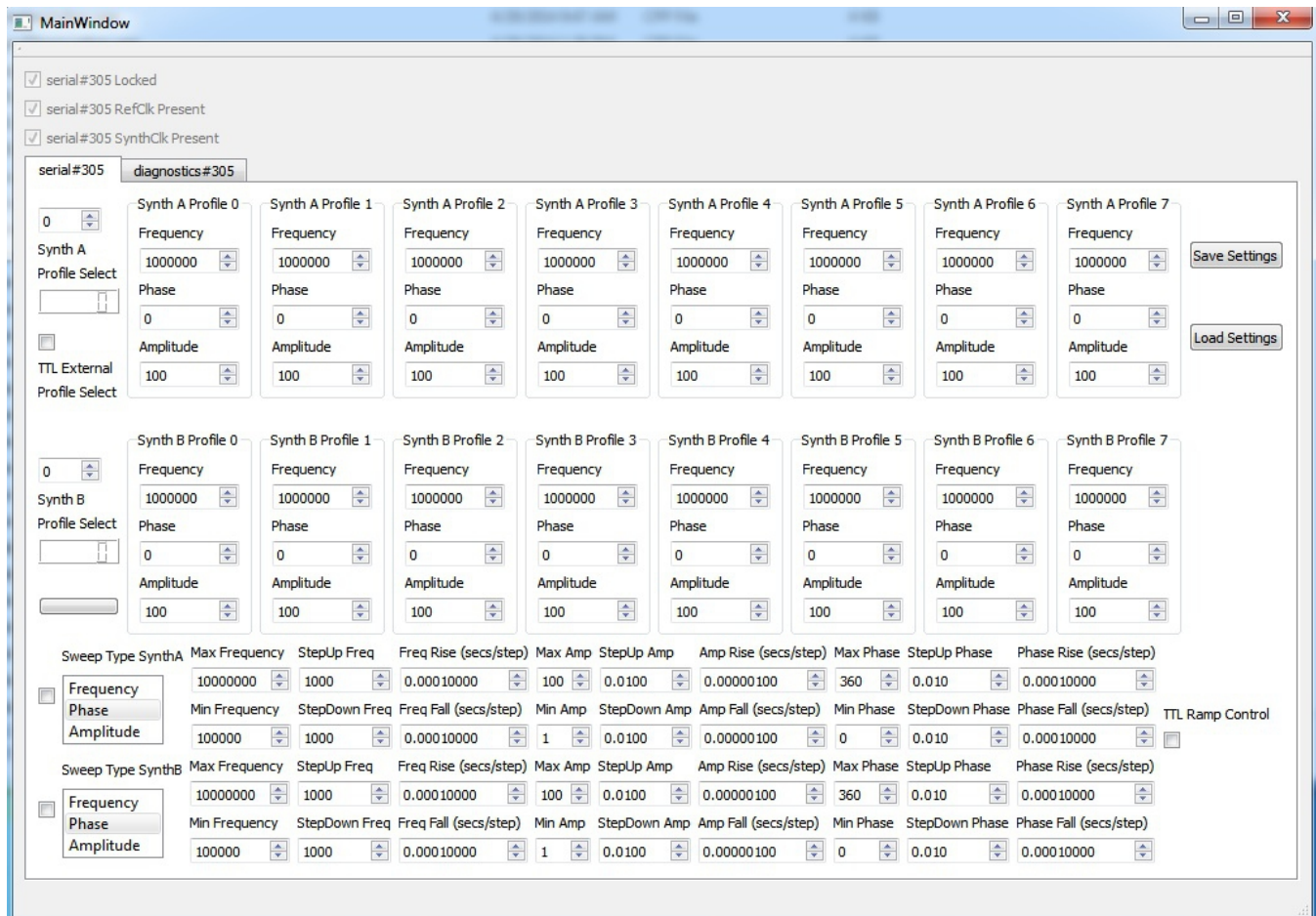
Table 8.1 external mode (“external profile mode” switch in enabled position)

PS0 (TTL2,TTL5)	PS1 (TTL1,TTL4)	PS2 (TTL0,TTL3)	Synth A Profile	Synth B Profile
0	0	0	0	0
0	0	1	1	1
0	1	0	2	2
0	1	1	3	3
1	0	0	4	4
1	0	1	5	5
1	1	0	6	6
1	1	1	7	7

USING THE QT GUI UNDER WINDOWS

The Qt based GUI is very similar to the Labview based GUI. Setup Board is called as the program initializes, and all the values are read in from the program and automatically fed to the board. Both profile mode and sweep mode are available in the Qt GUI, and there are 8 profiles for each synth. Several boards can be used at the same time in the Qt GUI, and each is given its own tab labeled by a serial number unique to each board. There is an external mode check box at the left side of the GUI for enabling external TTL profile mode select for both synths to rapidly switch between profiles through the serial interface after the profile settings are configured. The External Ramp mode check box is on the right and applies to both synths. The settings can be saved or loaded from the buttons on the far right, these are separate from the labview settings.

This program can be easily modified by downloading the current version of Qt, configuring a compiler on the system (MSVS 2012 is what was used to originally compile) opening up the qtGUI.pro file, then editing and recompiling.



SOFTWARE INSTALLATION AND USE UNDER CENTOS 6

Install the driver and software from the rpm file by double clicking and follow the prompts. All the software is installed under Filesystem:\uvdma.

The Qt GUI program is contained in Filesystem:\uvdma\example_programs\qtGUI and is almost identical to the windows version. Basic and intermediate functionality is provided, and advanced functionality can be programmed with the Qt development environment which is free for download. Additional details are in the windows section on the previous page.

The command line example program is contained in Filesystem:\uvdma\example_programs\synth_examples and shows many of the programmatic features of the synth as it does in windows, and should be viewed with an oscilloscope on several different time, channel, and triggering settings. This program can be quickly and easily edited to allow all kinds of advanced looping functionality. Simply edit the main.cpp file, and then make and run.

The command line argument program is contained in Filesystem:\uvdma\example_programs\synth_command_line and is the same as the windows version as well. It takes several parameters and writes to the synth. The program can be called from another program, or with a simple bash file to setup all the profile information and then turn on external triggering, as shown in the example bash file "profilesetup.sh". The first call has a "1" on setupboard (to initialize the board and setup the default parameters), and the last call has a "1" on external select (for TTL switching). This can be modified to suit any application, either by modifying the bash file, or by modifying the program code. The bash file will need to be edited to include your unique serial number as the first argument passed for this program. Multiple synths can be set up this way by including the appropriate serial number in the arguments. If additional functionality is required from this program, simply look at the uvAPI.h and uvAPI.cpp files for the prototype of the functions required, and then call them from main.cpp according to how they are needed.

CUSTOM APPLICATION PROGRAMMING OF THE SYNTH1000 IN WINDOWS

Powerful custom applications (beyond the functions incorporated in the GUI) for the Synth1000 device can be quickly programmed with source code provided. The easiest method for creating a custom application is to modify either of the GUI programs with the appropriate development environment, or either of the supplied command line software projects, `acquire.sln`, which are contained in the “command line simple control\command line utilities\acquire” and “command line examples\command line utilities\acquire” folders. This project is a Microsoft Visual Studio project and can be immediately built using Microsoft Visual Studio 2012. The main example file, `acquire.cpp`, uses the `uvAPI` object (described below) in order to configure a Synth1000x2 device, resulting in the executable `acquire.exe`. Modifications to either of these programs is simple for anyone proficient in the `c/c++` programming language.

SYNTH1000 OPERATIONAL OVERVIEW

The Synth1000 device uses a combination of AD9914 DDS-core implemented and firmware (FPGA-implemented) functionality. This combination results in an augmentation of the basic DDS core functions to enable capability such as triggered frequency hopping, frequency dwell/sweep/dwell capability, external TTL-controlled profile switching and ramp control, and other advanced capabilities.

In programming the board, one must remember an important difference between the CPLD and DDS core functional partition. Any CPLD-implemented function takes effect immediately upon the corresponding write to the board. However, DDS-core implemented functions take effect only after a qualifying “update event” as described below. This allows the user to configure a number of parameters in the DDS cores before an “update event” occurs, to allow event-initiated behavior such as triggered frequency/phase-hopping, dwell/sweep/dwell sinusoids, and other functionality.

“UPDATE EVENT”

As described above, the DDS cores of the Synth1000x2 device require an “update event” in order for any programmed changes following the most recent “update event” to take effect. Thus a programmer can configure the DDS cores for one type of operation while the current configuration is active and running. This is important for advanced operation, such as updating profile parameters based on sweep results.

The DDS cores recognize the following as qualifying “update events”:

A change in the enabled mode, profile, external, sweep, etc

A change in active profile on either `synthA` or `synthB`

A change in the frequency/amplitude/phase of the active profile on either `synthA` or `synthB`

A change in the sweep type

A change in the sweep rise/fall/rate/step/limits

Calling the `IOUpdate()` function (described below) will cause an “update event” for both DDSs, regardless of the state of the Internal/External profile select bit.

In either internal or external mode changing to a new profile will cause an “update event”. Note, in the software function `Set_Synth_Profile_Freq`, `Set_Synth_Profile_Phase`, and `Set_Synth_Profile_Amplitude` an explicit call to the `IOUpdate()` function is included to avoid confusion caused by changes that may not cause a profile change of one DDS.

In External Mode (External bit set to 1) any change to either of the 2 hardware (TTL-level) inputs will cause a qualifying update event for the DDS(s) that are affected by that/those bits.

USING THE UVAPI OBJECT

The API used to configure a Synth1000 device is defined in uvAPI.h and uvAPI.cpp as a C++ object called uvAPI. The uvAPI object provides a useful abstraction to the programmer. The collection of functions that define the API allow full access to the capabilities of a Synth1000 device without the need to understand the low level software interface to the control registers on the Synth1000.

INSTANTIATING A UVAPI OBJECT

In order to exploit the functionality provided by the uvAPI object the programmer must include the appropriate files into their project and instantiate a uvAPI object. The files that must be added to the programmer's project are located in the folder "Synth1000 API Object Source Code". All the files in this folder must be placed in the project directory and the files must be added to the project. (If the programmer is using the MS Visual Studio v12.0 project "acquire.sln" these files are already added to the project). These files are as follows:

Pcie5vDefines.h
uvAPI.h
acquire.cpp
uvAPI.cpp
get_usercode.svf
DllLibLoadSvf.cpp

Further description of the function of each file is included as comments inside each file.

UVAPI OBJECT MEMBER FUNCTIONS

As mentioned above the member functions of the uvAPI object provide the means for connecting to the driver service and configuring a Synth1000 device. The following is a brief explanation of these functions. Further explanation of each of the member functions is given within the code itself.

A brief description of the functions of the API are listed below. SetupBoard should be the first function called in an application, as this initializes all the registers to their default values, and loads some other necessary defaults for standard operation.

```
bool SetupBoard(unsigned short BoardNum);
```

A valid "update event" is required for certain functions to take effect. One method of generating a valid "update event" is through use of the IOUpdate function. Both Synths share an update, so SynthB passed can be either value with no difference in functionality. For further information refer to the AD9914 data sheet.

```
void IOUpdate(unsigned short BoardNum, bool SynthB);
```

The primary configuration of a Synth1000 device is facilitated by the Set_Synth_Profile_Select

function. Each DDS has 8 separate freq/phase profiles, and this function sets the corresponding profile in the corresponding Synth (DDS) with the given frequency, phase, and amplitude associated with that profile. SynthA is targeted by setting the first parameter to 0 (FALSE), and SynthB is targeted by setting the first parameter to 1 (TRUE). A profile within the targeted synth is targeted by setting the second argument to either 0, 1, 2, 3, 4, 5, 6, or 7.

```
void Set_Synth_Profile_Select(unsigned short BoardNum, bool SynthB, unsigned int Profile);
```

The 8 profiles can have their values modified by calling the appropriate function with the synth number (0 or 1 for A or B), the profile number (0,1,2,3,4,5,6, or 7), the appropriate argument for the modifier, and in the case of frequency, the system_clock, nominally 3.5Ghz unless using an external clock source.

```
void Set_Synth_Profile_Freq(unsigned short BoardNum, bool SynthB, float system_clock, unsigned int Profile, double Freq);
```

```
void Set_Synth_Profile_Phase(unsigned short BoardNum, bool SynthB, unsigned int Profile, float phase);
```

```
void Set_Synth_Profile_Amplitude(unsigned short BoardNum, bool SynthB, unsigned int Profile, float Amplitude);
```

By default, the board is set to internal software select profile mode (External bit set to 0). Enabling External Mode is accomplished by setting the External bit to 1. External Modulation is discussed previously. Setting the value back to 0 will disable the external modulation capability. Note that changing the External bit will cause an “update event”, as will every subsequent TTL event that changes the active profile externally.

```
void Set_Synth_External_Select(unsigned short BoardNum, bool SynthB, unsigned int Enable);
```

By default, the board is set up to control amplitude only through internal registers. External OSK (output shift keying) can be enabled to control the output on/off by a TTL pin.

```
void DIISetOSKExtEnable(unsigned short BoardNum, bool SynthB, bool Enable);
```

Frequency/amplitude/phase sweeping is enabled with the function SetDigitalRampEnable. Setting the enable bit to 1 will allow the sweep to take precedence over profile mode.

```
void SetDigitalRampEnable(unsigned short BoardNum, bool SynthB, unsigned int Enable);
```

The sweep type is configured with the function SetDigitalRampDestination. Frequency, amplitude, and phase can all be swept. FSK_ASK_PSK must pass in Frequency_Ramp (0), Phase_Ramp (1), or Amplitude_Ramp (2) to specify the sweep mode. SynthB selects which synth with 0 or 1.

```
void SetDigitalRampDestination(unsigned short BoardNum, bool SynthB, unsigned int FSK_ASK_PSK);
```

The speed and range of the sweep is determined by several functions. There is an upper limit, a lower limit, a step size for both rise and fall, and a step rate for both rise and fall. As usual, SynthB is passed a 0 or 1 indicating the synth to be accessed, the system_clk is 3.5ghz unless otherwise changed. FSK_ASK_PSK is specified by frequency (0), phase (1), or amplitude(2). The functions are listed below:

```
void SET_DIGITAL_RAMP_LOWER_LIMIT(unsigned short BoardNum, bool SynthB, float system_clk,  
unsigned int FSK_ASK_PSK, float Value)  
void SET_DIGITAL_RAMP_UPPER_LIMIT(unsigned short BoardNum, bool SynthB, float system_clk,  
unsigned int FSK_ASK_PSK, float Value)  
void SET_RISING_DIGITAL_RAMP_STEP_SIZE(unsigned short BoardNum, bool SynthB, float  
system_clk, unsigned int FSK_ASK_PSK, float Value)  
void SET_FALLING_DIGITAL_RAMP_STEP_SIZE(unsigned short BoardNum, bool SynthB, float  
system_clk, unsigned int FSK_ASK_PSK, float Value)  
void SET_POSITIVE_SLOPE_RAMP_RATE(unsigned short BoardNum, bool SynthB, float  
system_clock, float Value)  
void SET_NEGATIVE_SLOPE_RAMP_RATE(unsigned short BoardNum, bool SynthB, float  
system_clock, float Value)
```

LOW LEVEL SOFTWARE INTERFACE (MOST USERS NEED NOT READ)

The SYNTH1000 board is easy to communicate with. In most cases, this section may be skipped, as the driver supplied with the board will automatically handle all communication with the board registers. The best way to develop your own custom software is simply to modify any of the included example programs in either Labview, Qt, or visual c++ and then rebuild the application. With all the framework in place in multiple environments, new applications can be quickly developed with a minimal amount of work. Each has similar functionality and can be used based on what is most appropriate and familiar.

The analog outputs are created by two Analog Devices AD9914 synthesizers. In the event that the user would like to learn more about this device refer to the datasheet on the Analog Devices website at http://www.analog.com/static/imported-files/data_sheets/AD9914.pdf . For further information about custom applications, please contact us.