Vaunix Technology Corporation

Lab Brick® LMS Family of Signal Generators

Operation Manual
and
Programming Guide

Revision B1

Certification

Vaunix Technology Corporation certifies that this product met its published specifications at the time of shipment from the factory.

Warranty

Lab Brick Signal Generators are warranted against defects in material and workmanship for a period of one year from the date of shipment.

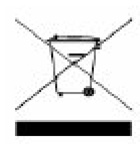
LIMITATION OF WARRANTY

The foregoing warranty does not apply to connectors that have failed due to normal wear. Also, the warranty does not apply to defects resulting from improper or inadequate maintenance by the Buyer, unauthorized modification or misuse, or operation outside of the environmental specifications of the product. No other warranty is expressed or implied, and the remedies provided herein are the Buyer's sole and exclusive remedies. Vaunix Technology Corporation shall not be liable for any direct, indirect, special, incidental, or consequential damages, whether based on contract, tort, or any other legal theory.

NOTICE

Waste Electrical and Electronic Equipment (WEEE) Directive 2002/96/EC

This instruction complies with the WEEE Directive (2002/96/EC) marking requirement. This affixed product label indicates that you must not discard this electrical/electronic product in domestic household waste.



To return an unwanted instrument, contact Vaunix Technology Corporation.

Table of Contents

1.0 GENERAL INFORMATION

This guide contains information on the installation , operation and specifications of the Lab Brick® LMS Family of Signal Generators.

1.1 General Safety Information

To prevent the risk of personal injury and loss related to equipment malfunction, Vaunix Technology Corporation provides the following safety information. For you own safety please read this section before operating the equipment.

Warning

Before connecting your Lab Brick Signal Generator to other instruments ensure that all instruments are connected to earth ground. Any interruption of the earth grounding may cause a potential shock hazard.

Caution

- The Lab Brick Signal Generator contains components which are sensitive to Electro Static Discharge (ESD). Proper ESD precautions must be maintained at all times while using this equipment.
- This equipment has no serviceable parts.
- To prevent the risk of electrical shock or damage to precision components, ***do not*** remove the equipment covers.
- Unauthorized entry into the unit voids all warranties.

2.0 GETTING STARTED

Prior to installing your Lab Brick Signal Generator, verify the contents of the package. The package should contain:

Quantity 1 Lab Brick Signal Generator

Quantity 1 Cable - USB Type A male/ B male

Quantity 1 Flash Drive containing the manual and the Graphical User Interface program

2.1 System Requirements

The Lab Brick Signal Generator runs from a standard PC or lap top computer with the following minimum requirements:

- Operating System - Windows® 7, Windows® 2000, Windows® XP, Windows® Vista, Windows® 8
- A minimum of one USB port

No other AC or DC supply is required as the power for this unit is delivered from a USB port on the computer or a self powered USB hub.

2.2 Installation of the Graphical User Interface (GUI)

The Lab Brick is controlled through the GUI program supplied on the provided USB flash drive. To install the GUI proceed with the following steps:

- Verify computer is loaded with .net framework version 4. If necessary, download and install .net framework 4 from Microsoft Corporation.
- Insert the supplied USB flash drive into an available USB port on the computer
- Run the program “Setup.exe”
- Follow the instructions on the screen
- After Installation is complete, remove the USB flash drive

2.3 Using the Lab Brick Signal Generator

Start the Lab Brick program by selecting the Lab Brick Icon or selecting the Lab Brick program from the Start Menu on the computer. Attach the supplied USB cable to the Lab Brick Signal Generator and the USB port on the computer. The green LED on the Lab Brick will illuminate as communication with the computer is automatically established. The GUI program will recognize the device and display the model number and serial number in the upper left and lower left corners respectively. The Lab Brick is now ready for operation.

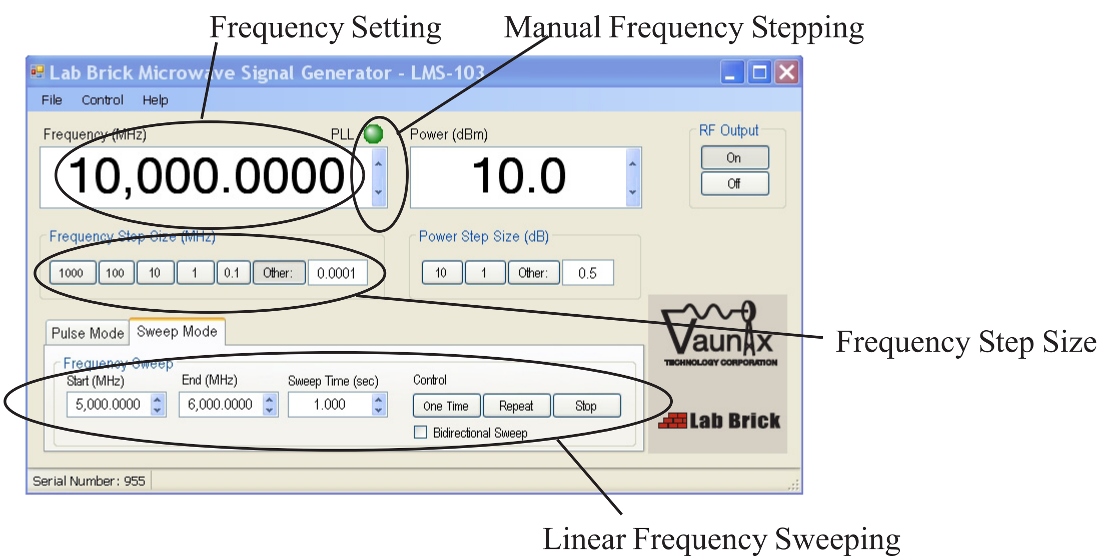
2.4 Using Multiple Lab Brick Signal Generators

Users may operate and control multiple Lab Bricks from a single computer. Start the Lab Brick GUI as described in section 2.3 for each Lab Brick Signal Generator that you will control from the computer. Connect each Lab Brick either directly to the USB port or through a self powered USB hub to the USB port of the computer. The green LED on each Lab Brick will illuminate as communication with the computer is automatically established. Each GUI application will automatically connect to one Lab Brick. The GUI will display the model number and serial number of the connected device in the upper left and lower left corners respectively.

3.0 OPERATING FEATURES AND CONTROLS

The general operation of the Lab Brick Signal Generator is designed by the Vaunix engineers to be intuitive and easy to use. This section describes the available features of the Lab Brick signal generator.

3.1 Frequency



3.1.1 Fixed Frequency Setting

The frequency is set using the Frequency field found on the left side of the GUI. Simply type the desired frequency into the window and hit the “Enter” key on your computer keyboard. The PLL indicator on the GUI will turn green to indicate that the desired frequency is established.

3.1.2 Configuring the Manual Frequency Step Size

The frequency may also be controlled by using the up and down arrows adjacent to the Frequency field. Use the controls directly below the frequency field to set the desired step size. Quick select buttons are available for fixed step sizes of 1000 MHz, 100 MHz, 10 MHz, 1 MHz and 100 kHz. Custom step sizes may also be used by selecting “Other” and entering the desired step size between 100 Hz and 1000 MHz.

3.1.3 Configuring the Phase Continuous Linear Frequency Sweep Function

The Lab Brick can be configured to linearly sweep through a range of frequencies. The user must specify the start frequency, end frequency and sweep time. The minimum start frequency and the maximum end frequency limits are predefined by the specific model number of the Lab Brick in use. The start and end frequencies may be set anywhere in this range. The sweep time can be set from 1 milliseconds to 1000 seconds.

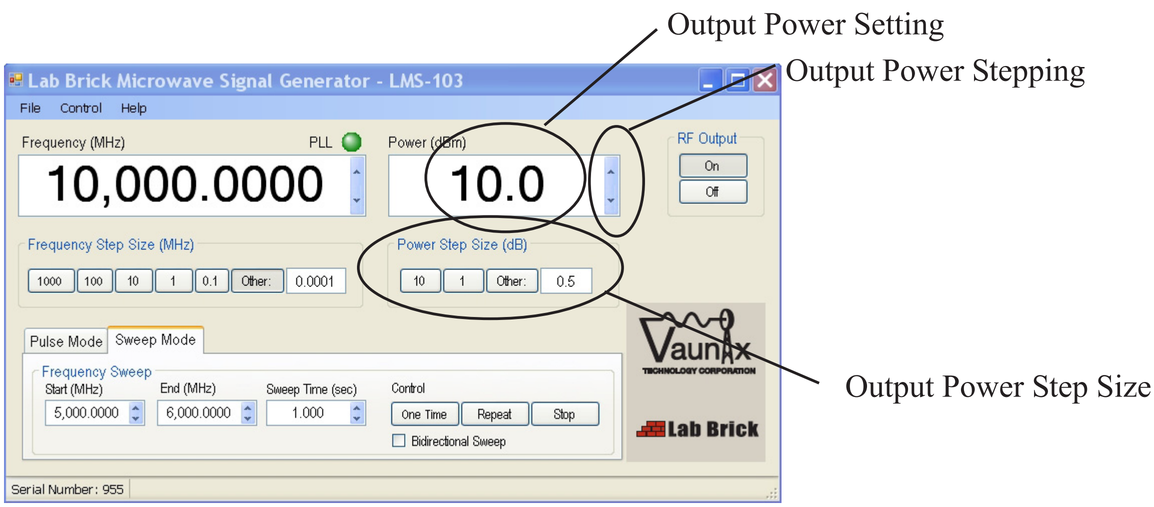
By selecting the “One Time” control button, the Lab Brick frequency will sweep from the start to the end frequency. Upon completing the sweep, the Lab Brick output will stay at the end frequency. Selecting the “Stop” button will disable sweep mode and allow the user to control the frequency from the Frequency field on the top of the GUI.

By selecting the “Repeat” control button, the Lab Brick will repeatedly sweep from the start to the end frequency. Upon reaching the end frequency the Lab Brick will blank the retrace and begin the sweep at the start frequency. The user may stop the sweep at any time by selecting the “Stop” button. The Lab Brick will remain in sweep mode until the “Stop” button is selected.

The user also has the option of using the “Bidirectional Sweep” feature. When the bidirectional selection in the sweep mode control box is checked, the Lab Brick will sweep from the start frequency to the end frequency and back to the start frequency. The sweep time during this operation is defined as the time to sweep from the start frequency to the stop frequency. Therefore, the full cycle from the start frequency back to the start frequency is twice the specified sweep time.

Note: The user must stop an active freqeuncy sweep in order for any changes to the sweep parameters to take effect.

3.2 RF Power Output



3.2.1 Fixed Output Power Setting

The output power is set using the Power field found on the right side of the GUI. Simply type the desired output power into the window and hit the “Enter” key on your computer keyboard. The output will immediately go to the desired output power.

Note: The LMS-163 and LMS-203 have a power control range of 40 dB. The GUI will allow the user 50 dB of control range for these models, but the accuracy is not guaranteed.

3.2.2 Configuring the Manual Output Power Step Size

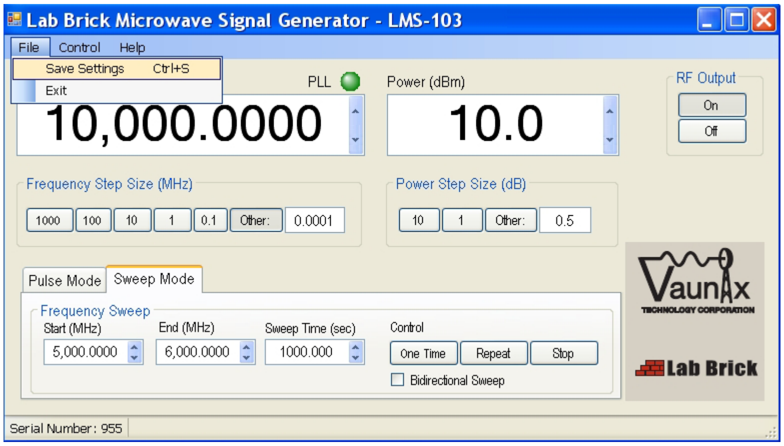
The RF power may also be controlled by using the up and down arrows adjacent to the Power field. Use the controls directly below the Power field to set the desired step size. Quick select buttons are available for fixed step sizes of 10 dB and 1 dB. Custom step sizes may also be used by selecting “Other” and entering the desired step size between 0.5 dB and 40 dB in 0.5 dB increments.

3.2.3 RF On/Off

The output power is enabled and disabled by selecting the RF On or RF Off button as desired. The PLL indicator on the GUI changes from green to red to indicate that the RF output is disabled.

3.3 Setting the Initial Operating State

After configuring the frequency and power, the user may select to save the current settings. From the File menu select Save Current Settings as shown in the figure below.



These settings will be stored within the Lab Brick device. The Lab Brick will now power on in this predefined state when plugged into a USB port on any computer or USB self powered hub. The user may change the saved state at any time by repeating the process.

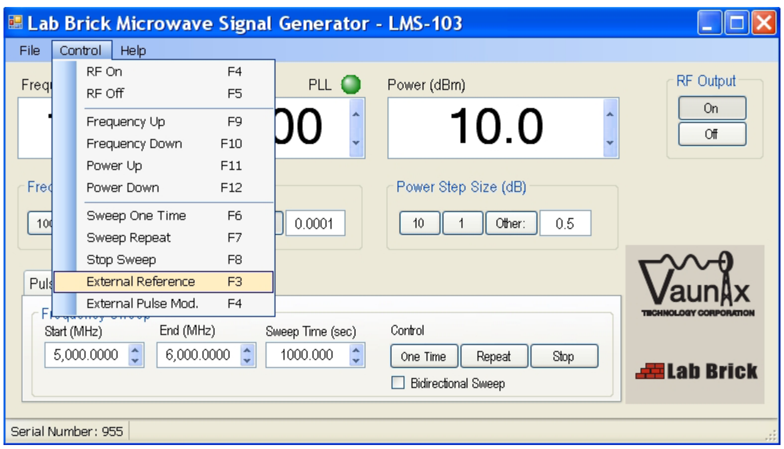
3.4 Autonomous Operation

The autonomous operation feature enables the Lab Brick to operate from a USB power source in the absence of a computer.

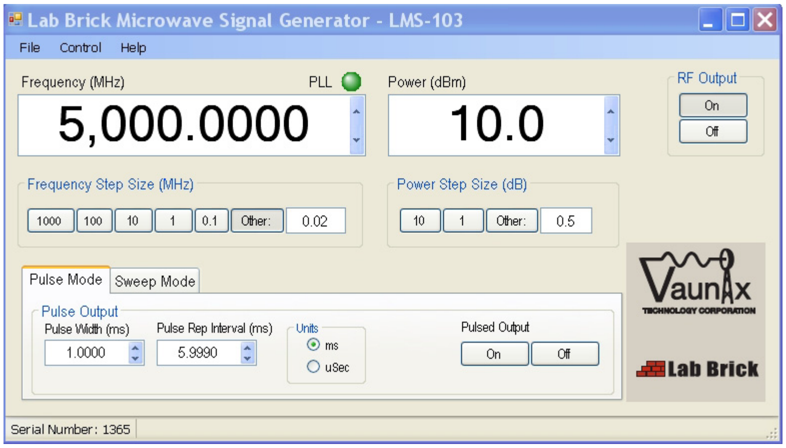
Configure the Lab Brick to the desired state and select the Save Current Settings command from the File menu. The power must now be removed from the Lab Brick by disconnecting the device from the computer or USB hub. When power is applied to the Lab Brick either from an unmanaged USB hub or USB battery pack the unit will begin operation at the saved setting. Please note that there is approximately 5 seconds of delay before the Lab Brick begins autonomous mode operation after power is applied.

3.5 External Reference

The Lab Brick can operate from the internal reference or it can be syncronized to an external 10 MHz source. To use an external reference select External Reference from the Control menu. Once selected, the GUI will display a check mark adjacent to the External Reference command in the Control menu. Alternatively, the user may toggle the reference source using the F3 key on the keyboard. The user can then attach the 10 MHz signal to the BNC connector located on the side of the Lab Brick.



3.6 Pulse Modulation (Option 003)

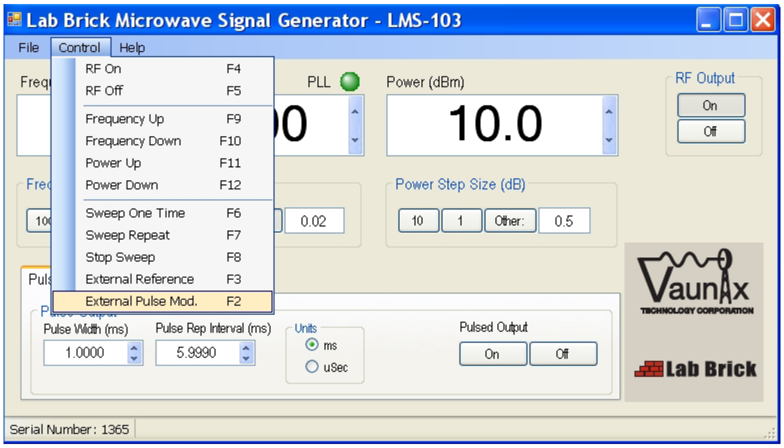


3.6.1 Internal Pulse Mode

The LMS series offers the ability to internally generate pulse modulation. Internal pulse modulation can be configured from the Pulse Mode window on the lower section of the GUI. The GUI allows control of the Pulse Width and the Pulse Repetition Interval (PRI). Internal pulse modulation is activated upon selection of the pulsed output "On" button. The minimum Pulse Width for internally generated pulse modulation is 100 ns. The minimum Pulse Repetition Rate is equal to the Pulse Width plus 100 ns.

The internally generated pulse trigger is available on the Trigger input/output BNC connector located on the side of the Lab Brick Signal Generator. Internal modulation will remain active until the Pulsed Output "Off" button is selected from the pulse mode window.

3.6.2 External Pulse Mode



To use the external pulse modulation feature, the user must select External Pulse Mode from the Control menu. Once selected, the GUI will display a check mark adjacent to the External Pulse Mode command in the Control menu.

The Trigger input/output BNC connector is located on the side of the Lab Brick Signal Generator. The external trigger must be TTL compatible with a minimum Pulse Width of 100 ns and a minimum Pulse Repetition Rate of 100 ns plus the Pulse Width.

3.7 Frequency Sweep Trigger (Option 004)

With the frequency sweep trigger option, the user can control the start time of the linear frequency sweep. The frequency sweep is configured as defined in section 3.1.3 of this manual. From the Control menu on the GUI, select External Sweep. A TTL level control signal present on the BNC connector labeled Sweep Trigger Input will start the specified sweep parameters.

Note: The repeat sweep function is not a valid setting when using the external sweep trigger mode. In Bidirectional Sweep mode, it is recommended that the trigger pulse “on” time be longer than the desired sweep time.

4.0 SPECIFICATIONS

Models

- LMS271D: .5 to 270 MHz
- LMS152D: .25 to 1.5 GHz
- LMS322D: .6 to 3.2 GHz
- LMS602D: 1.5 to 6 GHz
- LMS-103: 5 to 10 GHz
- LMS-163: 8 to 16 GHz
- LMS451D: 70 to 450 MHz
- LMS232D: .5 to 2.3 GHz
- LMS402D: 1 to 4 GHz
- LMS-802: 4 to 8 GHz
- LMS-123: 8 to 12 GHz
- LMS-203: 10 to 20 GHz

Electrical

- Frequency Resolution: 100 Hz
- Frequency Accuracy: +/- 2ppm
- Frequency Switching : 100 microseconds
- Output Power Range: LMS-163, LMS-203 +10 to -30 dBm; all other models +10 to -40 dBm
- Output Power Resolution: 0.5 dB
- Output Power Accuracy: +/- 0.75 dB at +10 dBm
  +/- 2.0 dB full range
- Spurious: -80 dBc typical, -70 dBc max
- Harmonics: -40 dBc typical, -15 dBc max
- SubHarmonics: LMS-802, LMS-103, LMS-123 -60 dBm typical; LMS-163, LMS-203 -25 dBc typical
- Output VSWR: 1.5:1 typical
- Phase Noise (typical):

	LMS-232D/402D	-602D	-802/103	-123	-163/203
1 kHz	-95 dBc/Hz	-85 dBc/Hz	-80 dBc/Hz	-77 dBc/Hz	-74 dBc/Hz
10 kHz	-98 dBc/Hz	-85 dBc/Hz	-81 dBc/Hz	-77 dBc/Hz	-75 dBc/Hz
100 kHz	-105 dBc/Hz	-94 dBc/Hz	-89 dBc/Hz	-86 dBc/Hz	-83 dBc/Hz
1 MHz	-129 dBc/Hz	-120 dBc/Hz	-118 dBc/Hz	-116 dBc/Hz	-114 dBc/Hz

- Internal/External Reference: Selectable
- Reference Frequency: 10 MHz
- Reference Input Level: 500mV to 3V Peak to Peak
- Phase Continuous Linear Frequency Sweep (LFM)
  Frequency Range: Full Band[1]
  Sweep Time: 1 ms to 1000 seconds
  Sweep Direction: Up, Down, Bidirectional

[1] Model numers with the ending D have linear sweep in 2 bands to cover the full frequency range. Consult factory for details

[2]Specifications are subject to change without notice

Electrical - continued

- Pulse Modulation: Optional
- Pulse Depth: LMS-163, LMS-203 all other models
  -45 dBc typical, -35 dBc min -70 dBc typical, -60 dBc min
- Rise/Fall Time: 30 ns typical
- Internal Pulse Mode
  Pulse Width: 100 ns min
  PRI: 100 ns plus Pulse Width min
  Resolution: 100 ns
  Trigger Output: 0 - 5 Volts
- External Pulse Mod
  Pulse Width: 100 ns min.
  PRI: 100 ns plus Pulse Width
  Trigger Input: 0 - 5 Volts nom, 0-3 Volts min
  Pulse Delay: 70 ns typical

Power Requirements

- Powered from USB Connection
- 5 Volts - 500 mA (550 mA for LMS-163 and LMS-203)

Environmental

- Operating Temperature: 0°C to 50°C
- Relative Humidity: <95% (non-condensing)

Dimensions

- Length: 4.90 in. (124mm)
- Width: 3.14 in. (80mm)
- Height: 1.59 in. (40mm)
- Weight: <1 lbs (<0.45Kg)

Physical Connections

- Power & Control: USB Type B - female
- RF Output: SMA - female
- External Reference: BNC - female
- Pulse Modulation: BNC - female
- Mounting: Counter-bore Through Holes (2)

5.0 OPTIONAL ACCESSORIES

Vaunix offers the following optional accessories for the Lab Brick signal generator family. Please consult your sales representative or visit Vaunix.com for up to date pricing and availability.

LPH-204B High Reliability, Low Noise 4 Port USB Hub with external power adapter

6.0 PROGRAMMING GUIDE

The Lab Brick Signal Generators are designed to be easily controlled from either their included control software or from applications programs that directly access the signal generators. The Lab Bricks use the USB HID class so that applications software can send commands and receive responses and status messages without the need to install any drivers or other special software components.

For developers creating software that will operate in the Microsoft Windows or Linux environments it is generally easier to use the interface DLLs or libraries supplied by Vaunix as a part of the software development kits for each of the Lab Brick products. The DLLs or libraries perform all the low level interface functions, allowing you to concentrate on the functionality of your application. This documentation is intended for programmers working in other environments, or who desire to control the device directly.

As with any USB HID device, there are two phases to working with the Lab Bricks. The first phase is the process of identifying the device you want to work with, and then opening the device to send and receive commands and status messages from it. The second phase is communicating with the device, using its commands to control it and reading its responses and status messages to determine the state of the Lab Brick.

This documentation includes examples from the Microsoft Windows™ environment. Similar strategies are used to communicate with USB HID devices under other operating systems, and this documentation will provide you with a general understanding of how to control the Lab Bricks under any operating system which supports USB HID class devices.

6.1 Identifying the Lab Brick Signal Generators

The Lab Brick Signal Generators are identified by their Vendor ID (“VID”) and Product ID (“PID”). Each Lab Brick also has a unique serial number, so that individual signal generators can be identified and selected in situations where multiple, otherwise identical Lab Bricks are connected to one computer.

[1] Actually, due to the architecture of USB HID devices, Interfaces on the device are opened and closed. A single HID device can have multiple Interfaces, and it is the Interfaces that are exposed by the HID Class driver for user level processes to interact with.

Model Name	Frequency Range	VID	PID
LMS-271D	0.5 to 270 MHz	0x041F	0x122a
LMS-451D	70 to 450 MHz	0x041F	0x1229
LMS-152D	0.25 to 1.5 GHz	0x041F	0x122b
LMS-232D	0.5 to 2.3 GHz	0x041F	0x1226
LMS-322D	0.6 to 3.2 GHz	0x041F	0x1225
LMS-402D	1 to 4 GHz	0x041F	0x1228
LMS-602D	1.5 to 6 GHz	0x041F	0x1227
LMS-802	4 to 8 GHz	0x041F	0x1221
LMS-103	5 to 10 GHz	0x041F	0x1220
LMS-123	8 to 12 GHz	0x041F	0x1222
LMS-163	8 to 16 GHz	0x041F	0x1224
LMS-203	10 to 20 GHz	0x041F	0x1223

Normally, in the Microsoft Windows environment, USB devices are identified by repeatedly calling the SetupDiEnumDeviceInterfaces function and then getting the symbolic link name for the HID device's interface with the SetupDiGetDeviceInterfaceDetail function. There are a number of publications that explain this technique, Writing Windows WDM Device Drivers by Chris Cant is a good starting point. Also, the Wiimote library by Brian Peek, (http://www.codeplex.com/WiimoteLib) is a good example of code for identifying and communicating with a USB HID device in a Microsoft Windows environment.

Use the SetupDiGetDeviceInterfaceDetail function to get the symbolic link name for the interface, which the operating system uses to encode its enumeration information describing the device. The string contains the VID and PID of the device found by the operating system. Test each string to find the one (or more if you have multiple Lab Bricks attached) that contains the VID and PID values in it. For the Microsoft Windows environment, the portion of the device strings containing the VID and PID are in the format:

sDevSubstring1 = “vid_041f&pid_1220”; // VID and PID for LMS-103

Once you have identified a Lab Brick, open it by using the DevicePath from the Interface Device Detail Data structure, using the normal CreateFile function. Once you have opened the device[1] you can read the Lab Brick's serial number using the HidD_GetSerialNumberString function.

```
WCHAR *pBuffer = new WCHAR [32];    // this buffer must be large enough to hold
                                    // any serial number
HidD_GetSerialNumberString(hDevice, pBuffer, 32);
```

If you are using multiple Lab Bricks you will need to identify the Lab Bricks, open them all, and then use the serial numbers returned by each of the devices to map the device handles to the specific Lab Bricks.

6.2 Controlling the Lab Brick Signal Generators

6.2.1 Commands

The Lab Brick Signal Generators use a simplified HID based set of commands. The commands, and the responses from the Lab Brick, are designed so that they can be easily created or parsed directly by your applications software. It is not necessary to use the normal HID API parsing functions.

Each Lab Brick command consists of an eight byte packet with the following format:

```
typedef struct
{
  BYTE command;
  BYTE count;
  BYTE byteblock[6];
} HID_REPORT_OUT;
```

(Note that in the Microsoft Windows environment, the HID driver stack requires a pre-pended 0 byte on packets written, and pre-pends a 0 byte to packets received, so your applications software needs to use a structure which has an additional BYTE before the command, and is therefore 9 bytes long.)

[2] The timing of the status reports will vary somewhat depending on the timing of the USB bus transactions, and software processes within the operating system, particularly for dwell times less than 100 milliseconds.

The command byte determines the meaning of the bytes within the byteblock. The count byte contains a count of the number of valid bytes in the byteblock. The values and the meaning of the bytes in the byteblock are set forth in the table below. For most commands the byteblock contains a 32bit DWORD quantity, several instances use a single byte quantity.

The most significant bit of the command byte determines whether the command gets or sets the parameter. To set the parameter, set the most significant bit. For example, to set the Lab Brick LMS-103 to 5.430GHz you would send the following command:

Command Byte	Count	Byteblock Contents
0xC4	4	0xC0, 0x80, 5D, 20, 00, 00

Frequency is set in 10Hz units, so 5.430GHz is 543,000,000 units, or 0x205D80C0. Note that the resolution of the integer used to represent the frequency does not necessarily represent the actual frequency resolution of the device. Refer to the specifications for each product to determine its actual output frequency resolution.

The command to get the current frequency is:

Command Byte	Count	Byteblock Contents
0x44	0	xx, xx, xx, xx, xx, xx

The Lab Brick responds with a report that contains the command byte in its status field, along with a count of 4 bytes and a DWORD representing the current frequency in the byteblock of the response. The format of responses will be described in more detail in the next section.

Command	Command Byte	Count	Byteblock Contents
Set/Get Frequency	0xC4/0x44	4	DWORD = Frequency in 10Hz units.
Set/Get Sweep Time	0xC5/0x45	4	DWORD = The length of the sweep in 1 millisecond intervals.
Set/Get Sweep Start Frequency	0xC6/0x46	4	DWORD = Lower limit of the sweep in 10Hz units.
Set/Get Sweep Stop Frequency	0xC7/0x47	4	DWORD = Upper limit of the sweep in 10Hz units.
Start/Stop Sweep	0xC8/0x48	1	Byte = 00 to stop a sweep, active sweeping modes are controlled by the lower 4 bits in the
			Bit 0 = 1 for a single sweep.
			Bit 1 = 1 for continuous sweeping
			Bit 2 = 1 to sweep downward in frequency, 0 to sweep upward.
			Bit 3 = 1 for a bi-directional triangle shaped frequency envelope, 0 for a single directional sweep.
Get/Set Pulse On Time	0xCA/0x4A	4	DWORD = to the on time of the RF output in pulsed mode operation, see the section below regarding units.
Get/Set Pulse Off Time	0xC9/0x49	4	DWORD = to the off time of the RF output in pulsed mode operation see the section below regarding units.
Pulsed Output Mode On/Off	0xCB/0x4B	1	Byte = 01 to enable internally pulsed output, 00 to disable internally pulsed output. Byte = 02 to enable external hardware control of the pulsed mode. External and internal modulation can be used simultaneously.
RF On/Off	0x8A/0x0A	1	Byte = 01 to enable RF output, 00 to disable RF output.
Power Level	0x8D/0x0D	1	Byte = power level relative to the maximum power of the Lab Brick Microwave Signal Generator, in .25db steps. 00 is maximum power, 02 is .5db less than full power. Note that the resolution of the power output setting is only .5db, the least significant bit of the value is ignored.
Restore Defaults	0x8F	1	Resets all of the parameters to their factory default settings.
Get Minimum Frequency	0x20	0	This is a read only value, only the “Get” command is supported.
Get Maximum Frequency	0x21	0	This is a read only value, only the “Get” command is supported.
Save User Parameters	0x8C	3	The first three bytes of the byteblock must be set to 0x42, 0x55, 0x31 as a key to enable the save operation. Save User Parameters records the frequency, power and sweep settings into non-volatile memory in the Lab Brick. The Lab Brick will reload these parameters when it is powered on.

As an example, this C code function sets the beginning frequency for a sweep to 5GHz:

```
static long FStart = 50000000;                // start sweep at 5Ghz

void SetSweepStart(HANDLE hDevice)
{
        unsigned char *ptr = (unsigned char *) &FStart;
        if (SendReport(hDevice, VNX_FSTART | VNX_SET, ptr, 4)){
                printf(" sending the sweep start frequency\n");
        }

}

SetSweepStart(hDevice);
```

Applications programs should ensure that commands are sent with a minimum delay of 30ms between commands, in order that the Lab Brick can generate and send its responses.

Pulse Modulation Mode

Lab Brick LMS Signal Generators can be ordered with an optional pulse modulation feature. For LMS devices with the pulse modulation option, the modulation can be internal, where the Lab Brick controls the modulation specified by the pulse on time and pulse off time, or external where an external signal controls the modulation.

The pulse on or off times can range from 100ns to more than 16 seconds. Two ranges are used. For times less than 1 millisecond, the pulse on and off time values are represented in units of 1/48MHz or approximately 2.08×10^{-6} seconds. For times greater than 1 millisecond the pulse on and off time values are represented in units of 1 microsecond. The range must be the same for the pulse on and off times and it is set with a range code in the high order nibble of the DWORD containing the pulse on time. The high order nibble of the pulse off time must be cleared.

Byte 3 (MSB) Byte 2 Byte 1 Byte 0 (LSB)

Range 28 bit (maximum) unsigned time value

The 4 bit range code is 0000b for microseconds, and 0001b for 1/48MHz units. For the 1/48MHz range the time value must be between 5 and 50,000. For the microsecond range the time value must be between 1 and 268,435,455. Practically speaking few applications will use values larger than 16×10^6.

For the off time the range code is ignored, and should be set to 0.

6.2.2 Responses

The Lab Brick Signal Generators send status reports to the host computer periodically while they are operating, and in response to some commands. Applications programs should normally set up and maintain a read thread to capture responses and status reports from the device. The status reports are designed to be easily parsed directly by the applications program.

Each Lab Brick response consists of an eight byte packet with the following format:

```
typedef struct
{
 BYTE status;
 BYTE count;
 BYTE byteblock[6];
} HID_REPORT1;
```

The status byte contains a value indicating the type of status report, like the commands the contents of the byteblock varies depending on the value in the status byte. For command responses, the value of the status byte is equal to the command. So for example, the response to the Get Frequency command shown above would have a status byte of 0x04, a count of 4 corresponding to the 4 bytes used by the DWORD in the byteblock, and a value of 0xC0805D20, or 543,000,000.

Status Byte	Count	Byteblock
0x04	4	0xC0, 0x80, 5D, 20, 00, 00

The Lab Bricks report their status periodically, at an interval equal to approximately 50 milliseconds when a sweep is not active. During a sweep, the status report occurs when the frequency changes. This allows an applications program to track the frequency output of the Lab Brick during the sweep. Note that the report of the new frequency may be received before the synthesizer stabilizes at the new frequency, so if the change in frequency is used to trigger a measurement process that process should be delayed until the synthesizer output is stable. The format for the periodic status report is:

```
typedef struct
{
 BYTE pkt_status;          // = 0x0E
 BYTE count;               // = 6
 DWORD frequency;
 BYTE dev_status;
 signed char power;
} VNX_STATUS_REPORT;
```

Status Byte	Count	Byteblock
0x4E	6	Frequency, dev_status, power

The dev_status byte contains a set of flags which describe the current state of the Lab Brick:

```
#define STATUS_PLL_LOCK 0x80       // MASK: PLL lock status bit, 1 = locked
#define STATUS_NEW_PARAM 0x40      // MASK: A parameter was set since the last
                                   // “Save Settings” command
#define STATUS_OK 0x20             // MASK: A command completed
#define STATUS_RF_ON 0x10          // MASK: The RF HW is on = 1 (note this bit
                                   // is in a different position than it is in the
                                   // LSG series of Lab Bricks
```

// Bit masks and equates for the Sweep command byte (stored in Sweep_mode, and reported also in Status)

```
#define SWP_BIDIR 0x08               // MASK: bit = 0 for ramp style sweep,
                                     //       1 for triangle style sweep
#define SWP_DIRECTION     0x04       // MASK: bit = 0 for sweep up, 1 for
                                     // sweep down
#define SWP_CONTINUOUS 0x02          // MASK: bit = 1 for continuous sweep
#define SWP_ONCE 0x01                // MASK: bit = 1 for single sweep
```

Note that the PLL lock status bit may reported as 0 during sweeps if the status report is generated at the instant that the frequency is changed, particularly in sweeps with significant frequency changes such as may occur when a repeating uni-directional frequency ramp retraces to the starting frequency.

Power is the current value of the power level as set by the Power Level command. It is relative to the maximum output power of the Lab Brick, a value of 00 represents full power, a value of 04 represents an output power level 1db less than full power.

Since Lab Brick status reports occur asynchronously with respect to command responses, the applications code handling reports from the Lab Brick should be able to accept either a command response report or a status report.

Command Response	Status Byte	Count	Byteblock Contents
Frequency	0x04	4	DWORD = Frequency in 10Hz units.
Sweep Time	0x05	4	DWORD = Length of the sweep in 1 millisecond intervals.
Sweep Start Frequency	0x06	4	DWORD = Lower limit of the sweep in 10Hz units.
Sweep Stop Frequency	0x07	4	DWORD = Upper limit of the sweep in 10Hz units.
Sweep Mode	0x09	1	Byte = 00 to stop a sweep, active sweeping modes are controlled by the lower 4 bits in the byte:
			Bit 0 = 1 for a single sweep.
			Bit 1 = 1 for continuous sweeping.
			Bit 2 = 1 to sweep downward in frequency, 0 to sweep upward.
			Bit 3 = 1 for a bi-directional triangle shaped frequency envelope, 0 for a single directional sweep.
Pulse On Time	0x4A	4	DWORD = to the on time of the RF output in pulsed mode operation, in scaled units as described in the section entitled Pulse Modulation Mode.
Pulse Off Time	0x49	4	DWORD = to the off time of the RF output in pulsed mode operation, in scaled units as described in the section entitled Pulse Modulation Mode.
Pulsed Output Mode	0x4B	1	Byte = 01 if the internal pulsed output is enabled, 00 if the internal pulsed output is disabled. Byte = 02 if external hardware control of the pulsed mode switching is enabled. External and internal modulation can be used simultaneously. An LMS device that does not have the pulse modulation option will return a value of 0x80 always.
RF On/Off	0x0A	1	Byte = 01 if RF output enabled, 00 if RF output disabled.
Power Level	0x0D	1	Byte = power level relative to the maximum power of the Lab Brick Signal Generator, in .25db steps. 00 is maximum power, 02 is .5db less than full power. Note that the resolution of the power output setting is only .5db, the least significant bit of the value is ignored.
Minimum Frequency	0x20	0	DWORD = The minimum frequency which the Lab Brick can generate.
Maximum Frequency	0x21	0	DWORD = The maximum frequency which the Lab Brick can generate.

6.3 Tips and Suggestions

Remember to handle the error cases for device removal, and close the device handles when you are done interacting with the device.

Remember that in the Microsoft Windows environment, the operating system pads the reports with a byte at the beginning of the report. Make sure to adjust your structures accordingly.

If you are programming in C or C++ you can create a set of unions to allow for convenient access to and conversion of the fields of the reports. For other languages, ensure that the byte order of a 32 bit unsigned integer is the same as the byte order used in the reports, which has the least significant byte stored in the lowest address.