LibreVNA USB Protocol

Version 12

October 26, 2022

Contents

1 USB device 2

2 General packet structure 2

3 Packet types 2

3.1 SweepSettings . . . 4
3.2 ManualStatusV1 . . . 5
3.3 ManualControlV1 . . . 5
3.4 DeviceInfo . . . 7
3.5 FirmwarePacket . . . 8
3.6 Ack . . . 8
3.7 ClearFlash . . . 8
3.8 PerformFirmwareUpdate . . . 8
3.9 Nack . . . 8
3.10 Reference . . . 8
3.11 Generator . . . 9
3.12 SpectrumAnalyzerSettings . . . 9
3.13 SpectrumAnalyzerResult . . . 10
3.14 Nack . . . 11
3.15 RequestSourceCal . . . 11
3.16 RequestSourceCal . . . 11
3.17 SourceCalPoint . . . 11
3.18 ReceiverCalPoint . . . 11
3.19 SetIdle . . . 11
3.20 RequestFrequencyCorrection . . . 11
3.21 FrequencyCorrection . . . 12
3.22 RequestAcquisitionFrequencySettings . . . 12
3.23 AcquisitionFrequencySettings . . . 12
3.24 DeviceStatusV1 . . . 12
3.25 RequestDeviceStatus . . . 13
3.26 VNADatapoint . . . 13
3.27 SetTrigger . . . 14
3.28 ClearTrigger . . . 14

1 USB device

The LibreVNA implements a "custom class" USB device. It uses a VID of 0x0483 and a PID of 0x4142. The custom class contains a single interface with three bulk endpoints:
- **Endpoint 0x01:** Communication data from the USB host to the LibreVNA
- **Endpoint 0x81:** Communication data from the LibreVNA to the USB host
- **Endpoint 0x82:** Debug messages from the LibreVNA

Endpoint 0x82 is exclusively used for debug messages. They are transmitted in ASCII format. All protocol packets described in this document are always transmitted over endpoints 0x01 and 0x81.

2 General packet structure

The USB traffic through bulk endpoints can be viewed as a stream of bytes. The communication between the LibreVNA and the USB host is done in packets. To detect the packets within the data stream, some framing is needed. This general package structure is described in this section.

Each packet consists of the following fields:
1. **Header:** 1 byte, always 0x5A
2. **Length:** 2 bytes, length of the overall packet in bytes, including the header and the checksum
3. **Type:** 1 byte, defines the type of packet and subsequently the data encoding within the payload
4. **Payload:** Any amount of bytes, content depends on the packet type
5. **CRC:** 4 bytes, CRC32 over all other packet bytes (header, length, type and payload)

All values in the USB protocol are little-endian.

3 Packet types

The following packet types are available:

Type	Name	Dir[a]	Description	Answer[a]
2	SweepSettings	H→D	Sets the sweep paramaters and starts the sweep in VNA mode	27[c]
3	ManualStatusV1	D→H	Contains the hardware status when in manual control mode	None
4	ManualControlV1	H→D	Transfers the manual control configuration, switches the device into manual control mode	3[c]
5	DeviceInfo	D→H	Contains the device information (firmware/hardware version, capabilities,...)	None
6	FirmwarePacket	H→D	Contains a piece of firmware data	None
7	Ack	D→H	Sent as a response to every successfully received and handled packet	None
8	ClearFlash	H→D	Triggers the flash erase procedure. Must be issues before transferring firmware data	None
9	PerformFirmwareUpdate	H→D	Triggers the firmware update once all firmware data has been transferred	None
10	Nack	D→H	Sent as a response to every unknown command or failure to execute the requested command	None

Type	Name	Dir[a]	Description	Answer[a]
11	Reference	H→D	Configure the external/internal reference	None
12	Generator	H→D	Switches the VNA into generator mode and configures the generator output	None
13	SpectrumAnalyzerSettings	H→D	Sets the sweep parameters and starts the sweep in spectrum analyzer mode	14[c]
14	SpectrumAnalyzerResult	D→H	Sent for every sampled frequency within the sweep in spectrum analyzer mode	None
15	RequestDeviceInfo	H→D	Makes the device send the DeviceInfo packet	5
16	RequestSourceCal	H→D	Makes the device send the source calibration packets	18[c]
17	RequestReceiverCal	H→D	Makes the device send the receiver calibration packets	19[c]
18	SourceCalPoint	D↔H	Contains a single source amplitude calibration point	None
19	ReceiverCalPoint	D↔H	Contains a single receiver amplitude calibration point	None
20	SetIdle	H→D	Stops all device activity	None
21	RequestFrequencyCorrection	H→D	Makes the device send the frequency calibration packet	22
22	FrequencyCorrection	D↔H	Contains the frequency calibration factor	None
23	RequestAcquisition-FrequencySettings	H→D	Makes the device send the acquisition frequency settings	24
24	AcquisitionFrequencySettings	D→H	Contains the configuration of IF and sample frequencies	None
25	DeviceStatusV1	D→H	Contains the hardware device status (lock, temperatures,...)	None
26	RequestDeviceStatus	H→D	Makes the device send the device status	25
27	VNADatapoint	D→H	Sent for every sampled frequency within the sweep in VNA mode	None
28	SetTrigger	D↔H	Updates the trigger status for synchronization over USB	None
29	ClearTrigger	D↔H	Updates the trigger status for synchronization over USB	None

[a] Direction of packet transfer:
- **D→H:** Device to host
- **H→D:** Host to device
- **D↔H:** Both directions used

[b] Packet type that will be sent in response to this packet
[c] The response will be sent multiple times

An Ack is transmitted by the device for every received command after it has been handled successfully. If additional responses are triggered by the command, they are transmitted after this Ack.

Received packets from the device are not acknowledged by the host; the host never sends an Ack packet.

3.1 SweepSettings

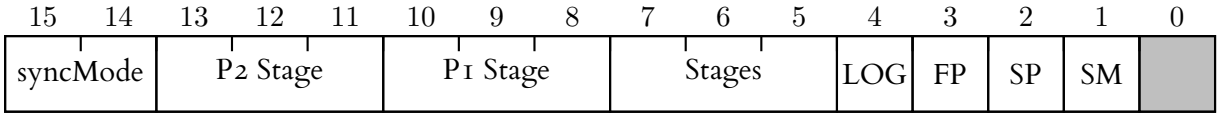
Transmitting this packet will switch the LibreVNA into VNA mode and start the sweep. During the sweep, VNADatapoint packets are generated for each completed point in the sweep.

The sampling for each frequency (or power) point in the sweep is done in stages. In each stage, the stimulus can be active at another port. A typical full two-port sweep would therefore use two stages, with the stimulus being active on port 1 during stage 1 and on port 2 during port 2. For faster measurements, this could be reduced to a single stage if only a subset of the S-parameters is required. Similarly, more than two stages can be used (with the stimulus inactive during some) when multiple devices are synchronized. Another device in the setup will have to generate the stimulus during the inactive stages.

The packet contains the following fields:

Offset	Length	Type	Name	Description
0	8	UINT64	f_start	Start frequency in Hz
8	8	UINT64	f_stop	Stop frequency in Hz
16	2	UINT16	points	Stop frequency in Hz
18	4	UINT32	IF_bandwidth	Bandwidth of the IF sampling in Hz
22	2	UIN164	cdbm_excitation_start	Stimulus power at the first point in $\frac{1}{100}$dBm
24	2	UINT16	Configuration	Bitmap for configuration, see below
26	2	UINT16	cdbm_excitation_stop	Stimulus power at the last point in $\frac{1}{100}$dBm

Configuration:



- **syncMode:** Synchronization mode when multiple devices are used together

Setting	Synchronization
00	Disabled
01	USB
10	External reference
11	External trigger

- **P1 Stage:** Sets the stage number when the stimulus is active at port 1. Stage number indizes start at 0.
- **P2 Stage:** Sets the stage number when the stimulus is active at port 2. Stage number indizes start at 0.
- **Stages:** Sets the number of used stages. The number of stages is one more than this value. E.g. set to 1 for 2 stages
- **LOG:** Set for a logarithmic sweep (only for frequency, power adjustment during the sweep is always linear)
- **FP:** Fixed power setting. This must be enabled for power sweeps (when cdbm_excitation_start ≠ cdbm_excitation_stop)

Setting	Behavior
0	Attenuator setting is fixed during the sweep. This will result in inaccurate stimulus level but prevent discrete jumps in output power.
1	Attenuator setting is changed during the sweep. This will result in more accurate stimulus level but also create discrete jumps in output power.

- **SP:** Suppress peaks. Recommended setting: always enabled.

Setting	Behavior
0	2.LO is adjusted to compensate for limited frequency resolution in 1.LO. Slight decrease in maximum sweep speed.
1	2.LO is kept at its nominal value. Slightly faster sweep but this will result in peaks at frequencies where the 1.LO it too far off the ideal frequency.

- **SM:** Sync Master. Must be set to 1 at exactly one device when multiple devices are synchronized. Set to 0 when synchronization is disabled.

3.2 ManualStatusV1

This packet is generated by the LibreVNA 1.0 when in manual control mode. It is transmitted in regular intervals on its own.

The packet contains the following fields:

Offset	Length	Type	Name	Description
0	2	UINT16	port1min	Minimum value of the ADC at port 1
2	2	UINT16	port1max	Maximum value of the ADC at port 1
4	2	UINT16	port2min	Minimum value of the ADC at port 2
6	2	UINT16	port2max	Maximum value of the ADC at port 2
8	2	UINT16	refmin	Minimum value of the ADC at the reference receiver
10	2	UINT16	refmax	Maximum value of the ADC at the reference receiver
12	4	FLOAT	port1real	Real part of the complex signal at port 1
16	4	FLOAT	port1imag	Imaginary part of the complex signal at port 1
20	4	FLOAT	port2real	Real part of the complex signal at port 2
24	4	FLOAT	port2imag	Imaginary part of the complex signal at port 2
28	4	FLOAT	refreal	Real part of the complex signal at the reference receiver
32	4	FLOAT	refimag	Imaginary part of the complex signal at the reference receiver
36	1	UINT8	temp_source	Temperature of the source PLL in °C
37	1	UINT8	temp_LO	Temperature of the LO PLL in °C
38	1	UINT8	Lock status	Bit 0: lock status of source PLL. Bit 1: lock status of LO PLL

3.3 ManualControlV1

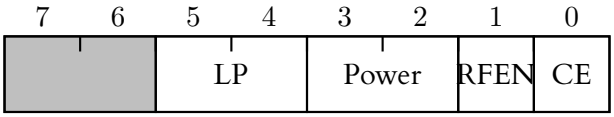
This packet switches the LibreVNA to manual control mode. As long as the manual control mode is active, the LibreVNA will generate ManualStatusV1 packets and send them to the host.

The packet contains the following fields:

Offset	Length	Type	Name	Description
0	1	UINT8	Source High Config	Configuration of the highband source
1	8	UINT64	Source High Frequency	Frequency of the highband source in Hz
9	1	UINT8	Source Low Config	Configuration of the lowband source
10	4	UINT32	Source High Frequency	Frequency of the highband source in Hz
14	2	UINT16	Source Path Config	Configuration of the source signal from the PLLs to the ports

Offset	Length	Type	Name	Description
16	1	UINT8	1.LO Config	Configuration of the 1.LO
17	8	UINT64	1.LO Frequency	Frequency of the 1.LO in Hz
25	1	UINT8	2.LO Enable	Set to 1 to enable the 2.LO. Set to 0 to disable the 2.LO
26	4	UINT32	2.LO Frequency	Frequency of the 2.LO in Hz
30	1	UINT8	Receiver enable	Bit 0: Enable port 1 receiver Bit 1: Enable port 1 receiver Bit 2: Enable reference receiver
31	4	UINT32	Samples	Number of ADC samples for each complex wave calculation
32	1	UINT8	WindowType	Window selection for the complex wave calculation

Source High Config:



- **LP:** Lowpass setting

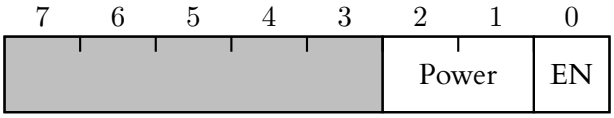
Setting	Cut-off frequency
00	947 MHz
01	1.88 GHz
10	3.5 GHz
11	No filter

- **Power:** Power output of the highband source PLL

Setting	Power
00	−4 dBm
01	−1 dBm
10	2 dBm
11	5 dBm

- **RFEN:** RF output enable
- **CE:** Chip enable

Source Low Config:

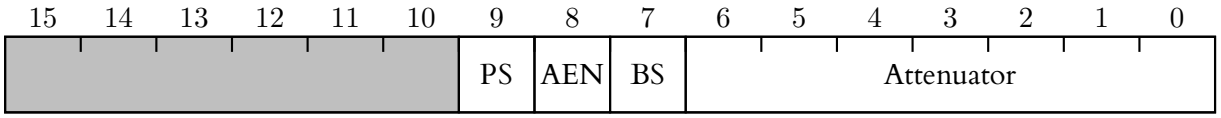


- **Power:** Power output of the lowband source PLL

Setting	Drive Strength
00	2 mA
01	4 mA
10	6 mA
11	8 mA

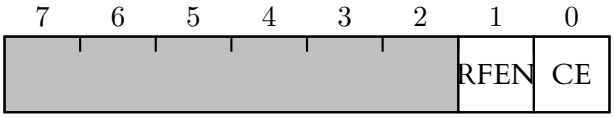
- **EN:** Lowband source enable

Source Path Config:



- **PS:** Port switch. Set to 1 to route the source signal to port 2, set to 0 to route the source signal to port 1.
- **AEN:** Amplifier enable.
- **PS:** Band select. Set to 1 to use the highband source, set to 0 to use the lowband source.
- **Attenuator:** Attenuation of the source signal in 0.25 dBm.

1.LO Config:



- **RFEN:** RF output enable
- **CE:** Chip enable

WindowType:

Setting	Window
0	None
1	Kaiser
2	Hann
3	Flattop

3.4 DeviceInfo

This packet contains information about the connected device. It can be requested by sending a RequestDeviceInfo packet. This request is the first thing that should happen after the device has been enumerated to make sure the right protocol version is used.

The packet contains the following fields:

Offset	Length	Type	Name	Description
0	2	UINT16	ProtocolVersion	Set to 12. If another value is reported, refer to the corresponding protocol description.
2	1	UINT8	FW_major	Major firmware version
3	1	UINT8	FW_minor	Minor firmware version
4	1	UINT8	FW_patch	Patch of the firmware version
5	1	UINT8	hardware_version	Version of the hardware, currently only '1'
6	1	CHAR	HW_revision	Revision of the hardware, currently only 'B' is used
7	8	UINT64	MinFreq	Minimum supported frequency in Hz
15	8	UINT64	MaxFreq	Maximum supported frequency in Hz
23	4	UINT64	MinIFBW	Minimum supported IF bandwidth in Hz
27	4	UINT64	MaxIFBW	Maximum supported IF bandwidth in Hz
31	2	UINT16	MaxPoints	Maximum number of points per sweep

Offset	Length	Type	Name	Description
33	2	UINT16	MincdBm	Minimum stimulus power in $\frac{1}{100}$dBm
35	2	UINT16	MaxcdBm	Maximum stimulus power in $\frac{1}{100}$dBm
37	4	UINT32	MinRBW	Minimum supported resolution bandwidth in Hz
41	4	UINT32	MaxRBW	Maximum supported resolution bandwidth in Hz
45	1	UINT8	MaxAmplitudePoints	Maximum supported number of amplitude calibration points
46	8	UINT64	MaxHarmonicFrequency	Maximum supported frequency when using harmonic mixing

3.5 FirmwarePacket

This packet contains a part of the firmware. When updating the firmware, this packet must be transmitted multiple times until the whole firmware has been transferred.

The packet contains the following fields:

Offset	Length	Type	Name	Description
0	4	UINT32	Address	Address at which the firmware data starts
4	256	UINT8	Data	Binary firmware data

3.6 Ack

This packet is sent by the device whenever a valid packet has been received. It has no payload.

3.7 ClearFlash

This packet must be sent before transferring the first piece of firmware data. It has no payload.

3.8 PerformFirmwareUpdate

This packet must be sent after the complete firmware data has been transmitted. It triggers the actual update process. The device will reboot during the update process. It has no payload.

3.9 Nack

This packet is sent by the device whenever an error occured while processing a received packet. It has no payload.

3.10 Reference

This packet is used to configure the external reference input and output.

The packet contains the following fields:

Offset	Length	Type	Name	Description
0	4	UINT32	OutputFrequency	Frequency of the external reference output. Not every frequency can be reached by the PLL. Set to 0 to disable the reference output.

Offset	Length	Type	Name	Description
4	1	UINT8	ExternalInputConfig	Bit 0: Switch to external when signal detected Bit 1: Force usage of the external reference

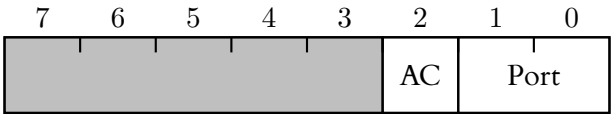
3.11 Generator

This packet switches the LibreVNA into signal generator mode and configures the output signal.

The packet contains the following fields:

Offset	Length	Type	Name	Description
0	8	UINT64	OutputFrequency	Output frequency of the generator in Hz
8	2	UINT16	cdBmLevel	Output level in $\frac{1}{100}$dBm
10	1	UINT8	Configuration	Configuration bitmap, see below

Configuration:



- **AC:** Amplitude correction enable. If set to 1, the source amplitude calibration is used to reach better amplitude accuracy.
- **Port:** Port selection:

Setting	Window
0	Disabled
1	Output on port 1
2	Output on port 2

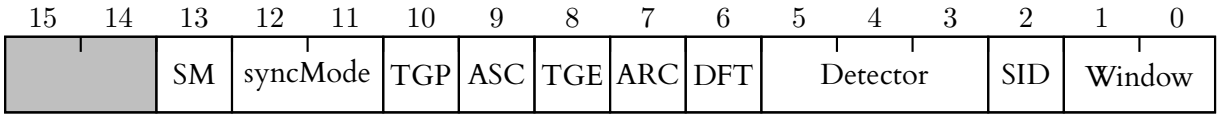
3.12 SpectrumAnalyzerSettings

Transmitting this packet will switch the LibreVNA into spectrum analyzer mode and start the sweep. During the sweep, SpectrumAnalyzerResult packets are generated for each completed point in the sweep.

The packet contains the following fields:

Offset	Length	Type	Name	Description
0	8	UINT64	f_start	Start frequency in Hz
8	8	UINT64	f_stop	Stop frequency in Hz
16	4	UINT32	RBW	Resolution bandwidth in Hz
20	2	UINT16	pointNum	Number of reported points in the sweep. The internally used number of points can be higher (depending on the RBW)
22	2	UINT16	Configuration	Bitmap for configuration, see below
24	8	INT64	TrackingOffset	Offset of the tracking generator in Hz
32	2	INT16	TrackingPower	Power of the tracking generator in $\frac{1}{100}$dBm

Configuration:



- **SM:** Sync Master. Must be set to 1 at exactly one device when multiple devices are synchronized. Set to 0 when synchronization is disabled.
- **syncMode:** Synchronization mode when multiple devices are used together

Setting	Synchronization
00	Disabled
01	USB
10	External reference
11	External trigger

- **TGP:** Tracking generator port. Set to 1 for port 2. Set to 0 for port 1.
- **ASC:** Apply source amplitude corrections. If enabled, the amplitude calibration is used to reach better accuracy of the tracking generator output.
- **TGE:** Tracking generator enable.
- **ARC:** Apply receiver amplitude corrections. If enabled, the amplitude calibration is used to reach better measurement accuracy.
- **DFT:** Use DFT to speed up the acquisition. Can not be used when the tracking generator is enabled. Only useful for low resolution bandwidths.
- **Detector:**

Setting	Detector type
0	Positive peak
1	Negative peak
2	Sample
3	Normal
4	Average

- **SID:** Signal ID enable.
- **Window:**

Setting	Window
0	None
1	Kaiser
2	Hann
3	Flattop

3.13 SpectrumAnalyzerResult

This packet is transmitted by the LibreVNA for every point in the sweep when in spectrum analyzer mode.

The packet contains the following fields:

Offset	Length	Type	Name	Description
0	4	FLOAT	Port 1	Signal level in mW at port 1
4	4	FLOAT	Port 2	Signal level in mW at port 2

Offset	Length	Type	Name	Description
8	8	UINT64	Frequency/Time	Frequency of the point (or time since beginning of SA mode if in zerospan)
16	2	UINT16	PointNum	Number of the point in the sweep

3.14 Nack

This packet is used to make the device send the DeviceInfo packet. It has no payload.

3.15 RequestSourceCal

This packet is used to make the device send the source amplitude calibration. It has no payload. For each source amplitude calibration point one SourceCalPoint packet will be returned.

3.16 RequestSourceCal

This packet is used to make the device send the receiver amplitude calibration. It has no payload. For each receiver amplitude calibration point one ReceiverCalPoint packet will be returned.

3.17 SourceCalPoint

This packet contains one source calibration point. It can be transmitted in both directions. When reading the source calibration, it is transmitted from the device to the host. When writing the source calibration multiple of these packets are transferred from the host to the device. In both cases the packet for the point with the highest point number must be transmitted last.

The packet contains the following fields:

Offset	Length	Type	Name	Description
0	1	UINT8	TotalPoints	Amount of total points in the amplitude calibration
1	1	UINT8	PointNum	Number of the calibration point contained in this packet
2	4	UINT32	Frequency	Frequency of the calibration point in 10 Hz
6	2	INT16	Port 1	Correction value for port 1 in $\frac{1}{100}$dB
8	2	INT16	Port 2	Correction value for port 2 in $\frac{1}{100}$dB

3.18 ReceiverCalPoint

This packet contains one receiver calibration point. It can be transmitted in both directions. When reading the receiver calibration, it is transmitted from the device to the host. When writing the receiver calibration multiple of these packets are transferred from the host to the device. In both cases the packet for the point with the highest point number must be transmitted last.

The packet payload is identical to the SourceCalPoint packet.

3.19 SetIdle

This packet is used to stop any data acquisition from the LibreVNA. It has no payload.

3.20 RequestFrequencyCorrection

This packet is used to make the device send the FrequencyCorrection packet. It has no payload.

3.21 FrequencyCorrection

This packet contains the frequency correction factor for the internal reference. It can be transmitted in both directions. When reading the frequency correction, it is transmitted from the device to the host. When writing the frequency correction, it is transmitted from the host to the device.

The packet contains the following fields:

Offset	Length	Type	Name	Description
0	4	FLOAT	PPM	Error of the internal TCXO in ppm

3.22 RequestAcquisitionFrequencySettings

This packet is used to make the device send the AcquisitionFrequencySettings packet. It has no payload.

3.23 AcquisitionFrequencySettings

This packet contains the configuration of the acquisition hardware. These settings are at default values after the device has booted. It is normally not required to send this packet but changing these settings might be useful in special use cases. It can be transmitted in both directions. When reading the acquisition settings, it is transmitted from the device to the host. When writing the acquisition settings, it is transmitted from the host to the device.

The packet contains the following fields:

Offset	Length	Type	Name	Description
0	4	UINT32	1.IF frequency	1.IF frequency in Hz
4	1	UINT8	ADC prescaler	Prescaler used for the ADC sampling (refer to the FPGA protocol)
5	2	UINT16	DFT phase increment	Phase increment of the DFT between ADC samples (refer to the FPGA protocol). Together with the ADC prescaler it also sets the 2.IF frequency.

3.24 DeviceStatusV1

This packet contains the status of the device. It can be requested by sending a RequestDeviceStatus packet. The device also sends this packet on its own. The interval in which this packet is sent depends on the currently active mode.

The packet contains the following fields:

Offset	Length	Type	Name	Description
0	1	UINT8	StatusBits	Bitmap of various states. See below.
1	1	UINT8	temp_source	Temperature of the source PLL in °C
2	1	UINT8	temp_LO1	Temperature of the 1.LO PLL in °C
3	1	UINT8	temp_MCU	Temperature of the microcontroller in °C

StatusBits:

7	6	5	4	3	2	1	0
	ULV	OVL	LLO	SLO	FC	ERU	ERA

- **ULV:** Unlevel. The requested output signal amplitude can not be reached. This is not actually measured and based on calculations only.
- **OVL:** ADC overload. The amplitude of at least one of the ADCs reached the non-linear region and the signal level can not be trusted.
- **LLO:** 1.LO locked.
- **SLO:** Source locked.
- **FC:** FPGA successfully configured.
- **ERU:** External reference used. The external reference input is used for all PLLs.
- **ERA:** External reference available. A signal is detected at the external reference input.

3.25 RequestDeviceStatus

This packet is used to make the device send the DeviceStatusV1 packet. It has no payload.

3.26 VNADatapoint

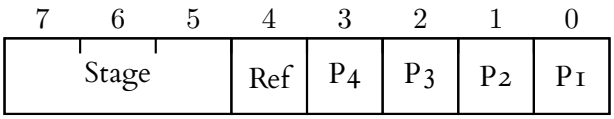
The VNADatapoint packet is generated by the device for every completed sweep point when in VNA mode.

The packet contains the following fields:

Offset	Length	Type	Name	Description
0	8	UINT64	Frequency	Frequency of the sweep point in Hz
8	2	INT16	PowerLevel	Stimulus level of the sweep point in $\frac{1}{100}$dBm
10	2	UINT16	PointNumber	Number of this point in the sweep
12	4*x	Array of FLOAT	Real values	The real parts of a variable amount of receiver data
12+4*x	4*x	Array of FLOAT	Imag values	The imaginary parts of a variable amount of receiver data
12+8*x	1*x	UINT8	Array of UINT8	Variable data description

The sampling data consists of a variable amount of values. The amount of values depend on the amount of configured stages and also on the hardware architecture (might change in the future). The VNADatapoint contains three arrays of equal length. Two of the arrays contain the real and imaginary parts of the acquired data. The third array contains a bitmask for every value, describing the content. The length of all arrays is not explicitly transmitted and must be inferred from the overall packet length.

Data description bitmask:



- **Stage:** The active stage when the value was acquired. The port on which the stimulus was active during this stage is known from the SweepSettings packet that was used to set up the currently active sweep.
- **Ref:** The value is from a reference receiver.
- **P4:** The value is from a port 4 receiver.
- **P3:** The value is from a port 3 receiver.
- **P2:** The value is from a port 2 receiver.
- **P1:** The value is from a port 1 receiver.

As the LibreVNA 1.0 only has two ports, P3 and P4 are never used and reserved for future developments.

In case of a three receiver architecture (as the LibreVNA 1.0 has), multiple port bits can be set for reference receiver values. For a typical full two-port sweep, the LibreVNA 1.0 will generate six values for every sweep point:

#	Bitmask	Content
1	0x01	Port 1 receiver signal during stage 0
2	0x02	Port 2 receiver signal during stage 0
3	0x13	Reference receiver signal during stage 0
4	0x21	Port 1 receiver signal during stage 1
5	0x22	Port 2 receiver signal during stage 1
6	0x33	Reference receiver signal during stage 1

3.27 SetTrigger

This packet is used when multiple devices are synchronized over USB and can be transmitted in both directions. It has no payload. Synchronized devices must be logically organized in a closed loop. When a SetTrigger packet is received from any devices in the loop it must be passed on to the next device in the loop.

3.28 ClearTrigger

This packet is used when multiple devices are synchronized over USB and can be transmitted in both directions. It has no payload. Synchronized devices must be logically organized in a closed loop. When a ClearTrigger packet is received from any devices in the loop it must be passed on to the next device in the loop.