Date: 2016-08-25
Version 1.0

Features

The emitted data is delivered at the sampling rate the measurement is running at regular intervals. The chosen delivery rate (i.e. how often UDP datagrams are emitted) depends on measurement configuration. User can request delivery rates of 100 Hz, 250 Hz, 500 Hz, 1 kHz, 2 kHz, 3 kHz, 4 kHz and 5 kHz. Data packets cannot be delivered at 1ms intervals if measurement sampling rate is below 1 kHz.

NeurOne digital out Ethernet port will be assigned an IP address relative to the control Ethernet port IP address. The control Ethernet port address is in turn determined by connection to SyncBox:

Main unit	Control IP address	Digital Out IP address
Stand-alone	192.168.200.200	192.168.200.220
SyncBox master	192.168.200.101	192.168.200.121
SyncBox slave1	192.168.200.102	192.168.200.122
SyncBox slave2	192.168.200.103	192.168.200.123
...		
SyncBox slave 9	192.168.200.110	192.168.200.130

Triggering information can be transmitted in packet or sample mode (or both). If packet mode transmission is enabled digital out will emit Triggers packets. If sample mode transmission is enabled digital out will emit an extra sample channel as the last channel in Samples packet.

There is no ACK mechanism for digital output. NeurOne handles only a single type of packet sent to its digital output interface (the Join packet). All other packets received by NeurOne are discarded.

UDP datagram length is limited by Ethernet MTU (IP layer fragmentation isn't supported). Each packet has at most 1472 bytes

Packet Types

There are many kinds of packets that NeurOne can emit to digital out. The emission of most of these can be enabled / disabled, depending on the actual needs of the user and the capabilities of the system receiving digital out data.

Packet type specifics are described in the following subchapters. Every field is encoded in big-endian byte order (i.e. most significant byte first). A packets type is identified by its first byte. This way the receiving system may trigger conditional parsing based on packet type – or even decide to discard certain packets.

The data types referenced in packet type descriptions are:

Type	Description
int8	8-bit signed integer

Type	Description
`uint8`	8-bit unsigned integer
`int16`	16-bit signed integer
`uint16`	16-bit unsigned integer
`int32`	32-bit signed integer
`uint32`	32-bit unsigned integer
`int64`	64-bit signed integer
`uint64`	64-bit unsigned integer
`t[N]`	Array of values, each of type *t*. If *N* is specified it denotes array length. If N isn't given (i.e. "`[]`") the array is variable-length.

MeasurementStartPacket

This packet is sent by NeurOne when it starts measuring, before any measurement data (samples / triggers) is emitted and also upon receival of Join packet. The emission of this packet is optional and may be enabled / disabled per-measurement. By default this packet isn't emitted.

The MeasurementStart packet structure is shown below:

<table>
<tr><th>Type</th><th>Name</th><th>Description</th></tr>
<tr><td><code>uint8</code></td><td><code>FrameType</code></td><td>1 = MeasurementStart</td></tr>
<tr><td><code>uint8</code></td><td><code>MainUnitNum</code></td><td>Indicates which main unit the packet is coming from (see MainUnitNum field in Samples packet).</td></tr>
<tr><td><code>uint8[2]</code></td><td><code>Reserved</code></td><td>Reserved for future use</td></tr>
<tr><td><code>uint32</code></td><td><code>SamplingRateHz</code></td><td>Sampling frequency (Hz)</td></tr>
<tr><td><code>uint32</code></td><td><code>SampleFormat</code></td><td>Currently this field has a fixed value (<code>0x80000018</code>),</td></tr>
<tr><td><code>uint32</code></td><td><code>TriggerDefs</code></td><td>Indicates which trigger type is assigned to each source port.
The type of each port is expressed as a value encoded in 3 bits:
<table>
<tr><th>Bits</th><th>Trigger port</th></tr>
<tr><td>0-2</td><td>Isolated A</td></tr>
<tr><td>3-5</td><td>Isolated B</td></tr>
<tr><td>6-8</td><td>Parallel trigger</td></tr>
<tr><td>9-11</td><td>SyncBox button</td></tr>
<tr><td>12-14</td><td>SyncBox external</td></tr>
</table>
Value meaning:
<table>
<tr><th>Value</th><th>Trigger type</th></tr>
<tr><td>0</td><td>Disabled</td></tr>
<tr><td>1</td><td>Stimulus</td></tr>
<tr><td>2</td><td>Video</td></tr>
<tr><td>3</td><td>Mute</td></tr>
<tr><td>4</td><td>Parallel</td></tr>
<tr><td>5-7</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td><code>uint16</code></td><td><code>NumChannels</code></td><td>Number of active digital out channels.</td></tr>
</table>

<table>
<tr><th>Type</th><th>Name</th><th>Description</th></tr>
<tr><td>uint16[N]</td><td>SourceChannels</td><td>Variable-length array of input numbers, indicating which analog input is mapped to each active digital out channel. (N = NumChannels).
E.g.
SourceChannels[0] = 2 => digitalOut1 = input2
SourceChannels[1] = 5 => digitalOut2 = input5
SourceChannels[2] = 4 => digitalOut3 = input4
SourceChannels[3] = 65535 => digitalOut4 = triggers

Main units may transmit input numbers as follows
<table>
<tr><th>Main unit</th><th>Input range</th><th>Trigger Ch#</th></tr>
<tr><td>Stand-alone</td><td>1..160</td><td>65535</td></tr>
<tr><td>SyncBox master</td><td>1..120</td><td>65534</td></tr>
<tr><td>SyncBox slave 1</td><td>121..240</td><td>65533</td></tr>
<tr><td>SyncBox slave 2</td><td>241..360</td><td>65532</td></tr>
<tr><td colspan="3">…</td></tr>
<tr><td>SyncBox slave 9</td><td>1080..1200</td><td>65524</td></tr>
</table></td></tr>
<tr><td>uint8[N]</td><td>ChannelTypes</td><td>Variable-length array of input channel metadata (N = NumChannels). Each 8-bit entry is as follows:
<table>
<tr><th>Bits</th><th>Meaning</th></tr>
<tr><td>0-2</td><td>Channel type (AC / DC)
000: AC
001: DC
Rest of the values are reserved</td></tr>
<tr><td>3-4</td><td>Amplifier type (EXG / Tesla)
00: EXG
01: Tesla
Rest of the values are reserved</td></tr>
<tr><td>5-7</td><td>Reserved</td></tr>
</table>
Trigger channel is an exception to the channel type presented above; its channel type is 0x80 ($1000\ 0000_2$).</td></tr>
</table>

Samples Packet

This packet contains measured digital samples. The samples are delivered in bundles. A bundle contains a sample for every channel for the same time instant, and cannot be broken into multiple sample packets. In other words, the samples are delivered as channel-interleaved and the samples of every channel from a single moment in time are delivered within the same sample packet.

This packet also describes the sample index and time stamp for the first sample contained. The Samples packet structure is shown below:

Type	Name	Description
uint8	FrameType	2 = Samples

Type	Name	Description
`uint8`	`MainUnitNum`	Indicates which main unit the samples packet is coming from: 0: Stand-alone 1: SyncBox master 2: SyncBox slave 1 … 10: SyncBox slave 9
`uint8[2]`	`Reserved`	Reserved for future use
`uint32`	`PacketSeqNo`	Packet sequence number (incremented by 1 for each sample packet sent)
`uint16`	`NumChannels`	Number of channels (constant throughout the measurement)
`uint16`	`NumSampleBundles`	Number of sample bundles in this packet
`uint64`	`FirstSampleIndex`	Sample index of first bundle in this packet
`uint64`	`FirstSampleTime`	Timestamp of first bundle in this packet
`int24[N][C]`	`Samples`	Sample bundles; each bundle contains 1 sample from each channel. (N = `NumSampleBundles`, C = `NumChannels`).

It should be noted that the sample values may need to be scaled by the receiver. The sample scaling factor depends on the amplifier hardware (amplifier type, input type (AC / DC)). This information is supplied in the MeasurementStart packet. The scaling factors are shown below:

Amplifier type	Channel type	Scaling factor
EXG	AC	1
EXG	DC	100
Tesla	AC	20
Tesla	DC	100

Send Triggers as a Channel Mode

When Send Triggers as a Channel mode is enabled the samples packet will contain an additional sample channel (always the last channel in the packet). The triggering information contains only the source port the trigger occurred on. Trigger types are defined in MeasurementStart packet. When a trigger occurs its corresponding bit will be high for the duration of 1 sample in the sample channel.

As triggers are relayed as samples, the temporal accuracy of triggering information is dictated by the measurement sampling rate. If the same trigger occurs multiple times during a single sample triggering information will be lost. If more accurate timing information is needed, it's advisable to use the Triggers packet which enables microsecond accuracy.

If trigger must occur on the sample of the rising edge the system must stall digital out sample delivery in order to deliver the trigger on the correct sample (t_{rise}). Alternatively, the trigger can be

delivered on the sample it is accepted on, i.e. the sample when trigger pulse length is satisfied on (t_{accept}) (see Figure 1).

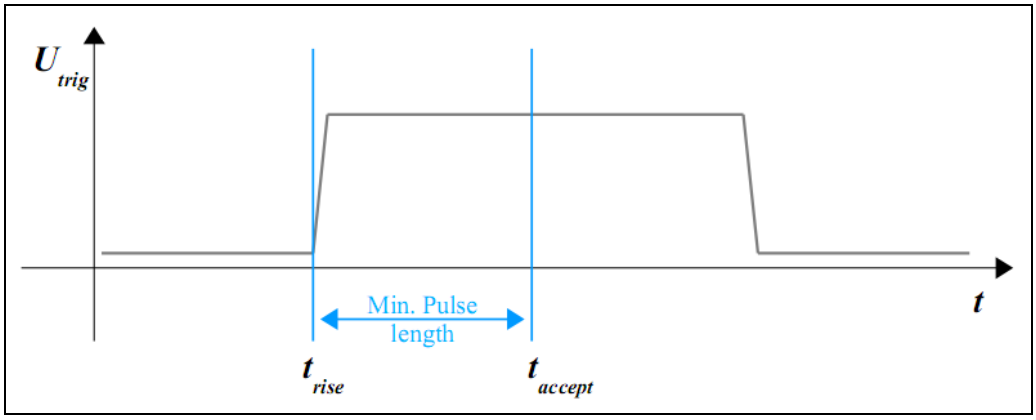

Figure 1. Trigger signal rising edge and accept timing.

Each trigger sample has the following information:

Bits	Name	Description
0	Res	Reserved for future use
1-2	Ai/Ao	Isolated A in (bit 1) / out (bit 2)
3-4	Bi/Bo	Isolated B in (bit 3) / out (bit 4)
5	SBB	SyncBox button
6	SBEi	SyncBox external trigger in
7	Res2	Reserved for SyncBox external trigger out
8-15	Par8	8-bit trigger code
16-23	Res3	Reserved for future use

Trigger Packets

This packet contains information about one or more trigger events registered by the measurement hardware. The emission of this packet is optional and may be enabled / disabled. By default this packet isn't emitted.

The Trigger packet structure is shown below:

Type	Name	Description
uint8	FrameType	3 = Triggers
uint8	MainUnitNum	Indicates which main unit the triggers packet is coming from (see MainUnitNum field in Samples packet).
uint16	NumTriggers	Number of triggers in this packet
uint32	Reserved	Reserved for future use
Trigger[N]	Triggers	Array of structures, each describing a single trigger event: (N = NumTriggers)

Type	Name	Description

Type	Name	Description		
		uint64	MicroTime	Trigger position relative to start time of measurement (microseconds)
		uint64	SampleIndex	Index of sample tied to trigger
		uint8	Type	4/4 bits; trigger source channel id and mode.
		uint8	Code	Parallel trigger code.
		uint8[2]	Reserved	Reserved for future use

The value of trigger type field is divided into upper and lower 4 bits; the upper 4 bits denote the physical channel the trigger came from. The lower 4 bits indicate the mode the trigger hardware was configured for:

Source id (upper 4)	Description
1	Isolated port A
2	Isolated port B
3	Parallel trigger
4	SyncBox button
5	SyncBox external trigger

Mode (lower 4)	Description
1	Stimulation trigger
2	Video trigger
3	Mute trigger
4	Parallel trigger
5	Trigger output

MeasurementEnd Packet

This packet is sent by NeurOne to indicate the end of the measurement. It's the last packet emitted when measurement hardware is stopped. The emission of this packet is optional and may be enabled / disabled per-measurement. By default this packet isn't emitted.

The MeasurementEnd packet structure is shown below:

Type	Name	Description
uint8	FrameType	4 = MeasurementEnd
uint8	MainUnitNum	Indicates which main unit the packet is coming from (see MainUnitNum field in Samples packet).
uint8[2]	Reserved	Reserved for future use
uint64	FinalSampleCount	Total number of sample bundles sent during this measurement

HardwareState Packet

This packet relays information about measurement hardware state, which can be considered metadata in regards to the measurement. It's sent by applicable NeurOne main units at the start of the measurement after MeasurementStart packet before any measurement data (samples /

triggers) is emitted and also upon receival of a Join packet. Note that this packet can contain different payloads which depend on the situation.

The emission of this packet is optional and may be enabled / disabled per-measurement. By default this packet isn't emitted.

The HardwareState packet structure is shown below:

Type	Name	Description
uint8	FrameType	5 = HardwareState
uint8	MainUnitNum	Indicates which main unit the packet is coming from (see MainUnitNum field in Samples packet).
uint8	StateType	Type of the payload: 1 = ClockSourceState
uint8	Reserved	Reserved for future use
uint8[]	StatePayload	Payload depending on StateType

ClockSourceState Packet

This state payload contains information about the current clock source of SyncBox system. It's not emitted on standalone systems. Every main unit sends this payload at the start of the measurement after MeasurementStart packet before any measurement data (samples / triggers) is emitted and also upon receival of a Join packet.

The payload structure is as follows:

Type	Name	Description
uint64	MicroTime	Clock source change time relative to start time of measurement (microseconds), according to sampling clock.
uint32	ClockFreq	Actual input clock frequency in Hz.
uint32	TargetClockFreq	Target (ideal) clock frequency in Hz.
uint16	ClockSrc	The clock source used by SyncBox: 1 = SyncBox internal 2 = BNC clock port of SyncBox 3= Fiber clock port of SyncBox

Join Packet

The Join packet is sent by the system receiving NeurOne digital out data. Main purpose of this packet is to enable MeasurementStart packet re-emission by NeurOne in cases where the receiver system becomes online after NeurOne measurement has already been started.

The Join packet structure is shown below:

Type	Name	Description
uint8	FrameType	128 = Join

Type	Name	Description
uint8[3]	Reserved	Must be filled with zeros, contents are discarded.

Upon receival of Join packet NeurOne sends a MeasurementStart in the following circumstances:

a) if digital out is sending to a unicast address: NeurOne responds to Join only if it originates from that same address.

b) If digital out is sending to a broadcast address: NeurOne responds to Join only if it originates from the same subnet (255.255.255.0). The MeasurementStart packet is sent as unicast to the IP address the Join was received from. This is for situations where a single client needs to acquire MeasurementStart.

It should be noted that the action of submitting a Join packet to NeurOne and NeurOne responding with a MeasurementStart packet may cause momentary jitter to digital out frame delivery timing.

Note: the receiving system does not need to send this packet unless it explicitly wants a new MeasurementStart packet.

Note: NeurOne will respond to Join packet only if emission of MeasurementStart packet is enabled in NeurOne configuration.

Note: The Join packet must be sent to UDP port 5050.

Examples

This chapter contains a number of packet flow examples for different digital out configurations. The time column indicates wall-clock time elapsed from the start of the measurement.

Case 1: MeasurementStart / MeasurementEnd / Trigger packet emission isn't enabled. Sampling rate 5 kHz, delivery rate 1 kHz.

Time (ms)	Packet	Description
Measurement is started		
1.0	Samples	Samples 0..4 for each output channel PacketSeqNo = 0 NumSampleBundles = 5 FirstSampleIndex = 0 MicroTime = 0
2.0	Samples	Samples 5..9 for each output channel PacketSeqNo = 1 NumSampleBundles = 5 FirstSampleIndex = 5 MicroTime = 1 000
3.0	Samples	Samples 10..14 for each output channel

Time (ms)	Packet	Description
		PacketSeqNo = 2 NumSampleBundles = 5 FirstSampleIndex = 10 MicroTime = 2 000
…		
Measurement is stopped		

Case 2: MeasurementStart / MeasurementEnd packet emission isn't enabled. Sampling rate 5 kHz, delivery rate 500 Hz.

Time (ms)	Packet	Description
Measurement is started		
2.0	Samples	Samples 0..9 for each output channel PacketSeqNo = 0 NumSampleBundles = 10 FirstSampleIndex = 0 MicroTime = 0
4.0	Samples	Samples 10..19 for each output channel PacketSeqNo = 1 NumSampleBundles = 10 FirstSampleIndex = 10 MicroTime = 2 000
4.2	Triggers	One or more triggers relating to samples sent in previous Samples packet.
6.0	Samples	Samples 20..29 for each output channel PacketSeqNo = 2 NumSampleBundles = 10 FirstSampleIndex = 20 MicroTime = 4 000
…		
Measurement is stopped		

Case 3: Emission of all packet types is enabled. Sampling rate 10 kHz, delivery rate 1 kHz.

Time (ms)	Packet	Description
Measurement is started		
0.0	MeasurementStart	Indicates start of measurement
0.0	HardwareState (ClockSourceState)	Relays hardware state metadata
1.0	Samples	Samples 0..9 for each output channel PacketSeqNo = 0 NumSampleBundles = 10 FirstSampleIndex = 0 MicroTime = 0
2.0	Samples	Samples 10..19 for each output channel PacketSeqNo = 1 NumSampleBundles = 10 FirstSampleIndex = 10 MicroTime = 1 000

Time (ms)	Packet	Description
2.2	`Triggers`	One or more triggers relating to samples sent in previous Samples packet

3.0	`Samples`	Samples 20..29 for each output channel
		`PacketSeqNo = 2`
		`NumSampleBundles = 10`
		`FirstSampleIndex = 20`
		`MicroTime = 2 000`

…		

1501.0	`Samples`	Samples 15000..15009 for each output channel
		`PacketSeqNo = 1500`
		`NumSampleBundles = 10`
		`FirstSampleIndex = 15000`
		`MicroTime = 150 000`

Measurement is stopped		
1501.7	`MeasurementEnd`	Indicates end of measurement

GNU Octave script for receiving UDP sample packets

```
pkg load sockets %loads the needed sockets packet

buff_size=50; %buffer size must be greater than the packet size

rcv_port=50000; %target port number

rcv_sck=socket(AF_INET, SOCK_DGRAM, 0);

bind(rcv_sck,rcv_port);

[str,len_s]=recv(rcv_sck,buff_size); %the first packet is empty packet


for n=1:10

  [pckt(n,:),len_s]=recv(rcv_sck,60); % buffer must be greater that the packet size

end
```

Note: This script is intended only for test purposes. It is not suitable for real time measurements.