

Distance Laboratory

Protocol Specification

Version 4.1

Protocol overview	2
Packer header	2
Packet types	2
Request header	2
Response header	3
Error packets	3
Instrument Interface	4
Instrument addresses	4
Instrument request and response	4
Function Generator.....	5
Function Generator request	5
Function Generator response.....	6
DC Power Supply.....	7
DC Power Supply Request.....	7
DC Power Supply Response	7
Oscilloscope	9
Oscilloscope Request	9
Oscilloscope response	13
Digital multimeter.....	15
Digital multimeter request.....	15
Digital multimeter response	16
Extended peripherals.....	17
Extended peripherals Request	17
Extended peripherals Response.....	17
Circuit builder.....	19
Circuit builder Request.....	19
Circuit builder Response	20
Digital IO	21
Digital IO Request.....	21
Digital IO Response	21
Circuit Switch	22
Circuit Switch Request.....	22
Circuit Switch Response	22
Data type explanation.....	23
Suggested command order	23

PROTOCOL OVERVIEW

The protocol is based on requests and responses. The client builds a experiment string, a request, and sends it to the server. The server then handles the request and produces a response that is sent back to the client.

Each request are sent in a separate TCP session, that means you have to connect and disconnect for every request made to the server.

Packer header

Packet types

There are four different packet types:

	Indicator	Reque st	Response
Data	“data”	x	x
Information	“info”	x	x
Error	“error”		x

x = Valid

Generally the packets look like:

<Length><\n><Type><\n><Content>

<Length> : String (6 characters long)

The length of the data following. Not including length.

<Type> : String

Type of request, is encoded as character. See the table above.

<Content> : Data

Packet payload

Separator is the new line character.

Request header

<Length> : String (6 characters long)

See above.

<Type>

As seen in the table above, only data and information packets can be sent as requests.

Data packets: Are used for instrument setup and measurement execution.

Information: Packets are used for requesting information about the server.

<Content>

The content or payload contains the instructions for the server, what we want to do.

Response header

<Length> : String (6 characters long)

See above.

<Type>

The type of this packet depends on what kind of request was made. But mainly it indicated if the request was successful or not (Error).

Data packets: Are used to return measurement data.

Information packets: Are used when requesting information from the server.

Error packets: Are used when an error occurs.

<Content>

Measurement data, information or error message.

Error packets

When something goes wrong or if the request is invalid, the server will return a error packet.

<Error length><\n><Error><\n><Error message>

<Error length> : String (6 characters long)

String encoded length of the error message following.

<Error>

Packet type for error (“error”).

<Error message>

Description of the error. Short text describing what went wrong.

INSTRUMENT INTERFACE

To execute a experiment, all instrument states must be sent to the server in chronological order. That is, the order that they should be executed on the server. You should follow the standard rules of experimentation when deciding in which order command is to be done. (See command order documentation)

Notice: It may be possible to construct request that can be harmful to the hardware, or at least produce unexpected results. Therefore the requester should be careful to validate the request before it is sent to the server. More documentation about validating requests are available.

Instrument addresses

<Instrument ID>	Enum
Function Generator (5401)	11
DC Power Supply	12
Oscilloscope	21
Digital Multimeter	22
Extended Peripherals	31
Circuit Builder	41
Digital I/O	42
Circuit Switch	44

Instrument request and response

Requests and responses are sent as data packets.

There can be multiple instrument requests/responses in one packet, separated by newline ('**\n**').

All instrument requests and responses are encoded in the style of:

```
<InstrumentID><\t><Instrument Data>  
<InstrumentID>          : See above
```

<Instrument Data>

One or more instrument request/responses, separated by '**\n**'

FUNCTION GENERATOR

Function Generator request

Each element should be separated by space.

<Function><...>

<Function>	Enum
Setup	0
Fetch	1

Fetch: <Nil>

Setup: <Waveform><Amplitude><Frequency><DC Offset><Start phase>
<Trigger mode><Trigger source><Burst count><Duty Cycle High><User defined waveform>

<Waveform>	Enum
Sine	0
Square	1
Triangle	2
Ramp up	3
Ramp down	4
DC	5
Noise	6
User defined	7

<Amplitude> : Double

Limits: 0 – 10.0 V peak to peak

<Frequency> : Double

Limits: Sine, Square = 20 MHz, All others = 1 MHz

<DC Offset> : Double

Limits: -5 V to +5 V, |(DC Offset + Amplitude)| < 10 V

<Start phase> : Double

Limits: -180.0 to +180.0 degrees

<Trigger mode>	Enum
Single**	0
Continuous	1
Stepped**	2
Burst*	3

<Trigger source>	Enum
Immediate	0
External	1

<Burst count>* : Integer

Is 0 when disabled.

Notice: Should always be set to 0 on the 5401 function generator

<Duty Cycle High> : Double

This property controls the duty cycle of the square wave the function generator is producing. You specify this property as a percentage of time the square wave is high in a cycle. Default Value: 50%

<User defined waveform> : Array of 512 Double or '0'

Should only be sent as a single 0, when no user waveform is used.

** Is only supported by 5411 Functions generator*

***Only continuous triggering mode is available in standard function output mode*

Function Generator response

When Fetch request are sent to the server, a resulting string is returned.

<Setup>: <Instrument ID><Function>

<Instrument ID> : Integer

<Function> : Integer

<Fetch>: <Instrument ID><Function><Waveform><Amplitude><Frequency><DC Offset><Start phase><Trigger mode><Trigger source><Duty Cycle High>

<Instrument ID> : Integer

<Function> : Integer

<Waveform> : Integer

<Amplitude> : Double

<Frequency> : Double

<DC Offset> : Double

<Start phase> : Double

<Trigger mode> : Integer

<Trigger source> : Integer

<Duty Cycle High> : Double

DC POWER SUPPLY

DC Power Supply Request

Each element should be separated by space.

Each channel has separate characteristics:

+6 (0.0 - 6.0) V, (0.0 – 1.0) A

+20 (0.0 - 20.0) V, (0.0 – 100m) A*

-20 (-20.0 - 0.0) V, (0.0 – 100m) A*

*the combined outputs must not exceed 3 W total.

With an auxiliary power supply connected:

+6 (0.0 - 6.0) V, (0.0 – 1.0) A

+20 (0.0 - 20.0) V, (0.0 – 1.0) A

-20 (-20.0 - 0.0) V, (0.0 – 1.0) A

<Function><...>

<Function>	Enum
Setup	0
Fetch	1

Fetch: <Nil>

Setup: <Enable><Voltage +6><Current Limit +6><Voltage +20>
<Current Limit +20><Voltage -20><Current Limit -20>

Setup does not return any data. To do that you have to send fetch data (as function). The fetch request doesn't have any other parameters. Setup must be done before fetch.

<Enable> : Integer

0: Disable

1: Enable (**default**)

<Voltage> : Double

Valid value is described under the parameter Channel Name above.

<CurrentLimit> : Double

Valid value is described under the parameter Channel Name above.

DC Power Supply Response

<Setup>: <Instrument ID><Function>

<Instrument ID> : Integer

<Function> : Integer

<Fetch>:

<Power Channel * 3>

Order +6V, +20V, -20V

<Power Channel>: **<Output Voltage><Output Current>**

<Output Voltage> **: Double**
Delivered voltage

<Output Current> **: Double**
Delivered current

Notice: all channels are always included in the response.

OSCILLOSCOPE

Oscilloscope Request

Each element should be separated by space.

<Function><...>

<Function>	Enum
Setup	0
Fetch data	1
Fetch	2

Fetch: <Nil>
Setup: <Autoscale><Horizontal Conf><Channel * 2>
 <Trigger><Measurement * 3>

Setup does not return any data. To do that you have to send fetch data (as function). The fetch request doesn't have any other parameters. Setup must be done before fetch.

<Autoscale> : Boolean

If autoscale is used the oscilloscope makes all settings according to the signal. (This is a function created by NI and we don't have any opportunities to change anything).

<Horizontal Conf> : <Min. Sample Rate><Reference Position>
 : <Record Length>

<Horizontal Conf.Min. Sample Rate> : Double

Specifies the sampling rate for the acquisition.

Units: Samples per second

Valid Values: The combination of sampling rate and minimum record length must allow the digitizer to sample at a valid sampling rate for the acquisition type specified in niScope Configure Acquisition and not require more memory than the onboard memory module allows.

NI Default Value: 20 MS/s

<Horizontal Conf.Reference Position> : Double

Specifies the position of the Reference Event in the waveform record as a percentage of the record. When the digitizer detects a trigger, it waits the length of time the Trigger Delay property specifies. The event that occurs when the delay time elapses is the Reference Event. The Reference Event is relative to the start of the record and is a percentage of the record length. For example, the value 50.0 corresponds to the center of the waveform record and 0.0 corresponds to the first element in the waveform record.

NI Default Value 50.0%

<Horizontal Conf.Record Length> : Integer

Passes the minimum number of points you need in the record for each channel; call niScope Actual Record Length to obtain the actual record length used. Refer to Coercions of Horizontal Parameters for more information about why the record length may be different than what was specified.

Valid Values: Greater than 1; limited by available memory

NI Default Value 1000

<Channel> : <Enable>

<Channel.Enable> : Boolean

False: <Nil>

True: <Vertical coupling><Vertical range><Vertical offset>
<Probe attenuation>

If false no more channel parameters should be sent.

<Channel.Vertical coupling>	Enum
AC	0
DC	1
GND	2

Specifies how the digitizer couples the input signal for the channel. When changing input coupling, the input stage takes a finite amount of time to settle.

<Channel.Vertical range> : Double

Specifies the absolute value of the input range for a channel. The units are volts. For example, to acquire a sine wave that spans between -5 and $+5$ V, set the Vertical Range property to 10.0 V. If the signal is outside vertical range, the response will be clamped to the measurement window, eg. \pm vertical range.

<Channel.Vertical offset> : Double

Specifies the location of the center of the range. The value is with respect to ground and is in volts. For example, to acquire a sine wave that spans between 0.0 and 10.0 V, set this property to 5.0 V.

<Channel.Probe attenuation> : Double

Specifies the probe attenuation for the input channel. For example, for a 10:1 probe, you would set this property to 10.0.**NI Default Value 1.0**

Limits: Any positive real number. Typical values are 1, 10 and 100.

<Trigger> : <Source><Slope><Coupling><Level><Holdoff><Delay>
: <Trigger mode><Timeout>

<Trigger.Source>	Enum
Channel 1	0
Channel 2	1
Immediate	2
External trigger	3

<Trigger.Slope>	Enum
Positive	0
Negative	1

<Trigger.Coupling>	Enum
AC	0
DC	1

<Trigger.Level> : Double

Specifies the voltage threshold for the trigger. The units are volts. The value you pass for this parameter must meet the following conditions. Trigger Level \leq Vertical Range/2 + Vertical Offset and Trigger Level \geq (-Vertical Range/2) + Vertical Offset

<Trigger.Holdoff> : Double

Specifies the length of time the digitizer waits after detecting a trigger before enabling the trigger subsystem to detect another trigger. The units are seconds. This property affects instrument operation only when the digitizer requires multiple acquisitions to build a complete waveform.

NI Default Value: 0.0 s

<Trigger.Delay> : Double

Specifies the trigger delay time in seconds. The trigger delay time is the length of time the digitizer waits after it receives the trigger. The event that occurs when the trigger delay elapses is the Reference Event.

NI Default Value: 0.0 s

<Trigger mode>	Enum
Normal	0
Auto	1
Auto level	2

Normal: If no trigger is received an error message is returned.

Auto: If no trigger is received an immediate trigger is set.

Auto level : If no trigger is received a volt max measurement is performed and the trigger level is adjusted according to this. If no volt max level is found an immediate trigger is set.

<Trigger.Timeout> : Double

Defines how long the fetch function shall wait for a trigger before it timeouts.

<Measurement> : <Channel><Selection>

<Measurement.Channel>	Enum
Channel 1	0
Channel 2	1
*Channel 3	3
*Channel 4	4

**currently not in use*

<Measurement.Selection>	Enum
AC Estimate	1012
Area	1003
Average Frequency	1016
Average Period	1015
Cycle Area	1004
DC Estimate	1013
Fall Time	1
Falling Slew Rate	1011
FFT Amplitude	1009
FFT Frequency	1008
Frequency	2
Integral	1005
Negative Duty Cycle	13
Negative Width	11
None	4000
Overshoot	18
Period	3
Phase Delay	1018
Positive Duty Cycle	14
Positive Width	12
Preshoot	19
Rise Time	0
Rising Slew Rate	1010
Time Delay	1014
Voltage Amplitude	15
Voltage Average	10
Voltage Base	1006
Voltage Base to Top	1017
Voltage Cycle Average	17
Voltage Cycle RMS	16
Voltage High	8
Voltage Low	9
Voltage Max	6
Voltage Min	7
Voltage Peak to Peak	5
Voltage RMS	4
Voltage Top	1007

Oscilloscope response

When an oscilloscope request is sent with function set to fetch data, the oscilloscope will return (on success) a waveform for each channel enabled.

Values that has not been requested are returned as 0.000000

Notice: all channels are always included in the response.

<Setup>: <Instrument ID><Function>

```
<Instrument ID>           : Integer
```

```
<Function> : Integer
```

<Fetch data>

<Horizontal><Channel *2><Measure *3><Trigger>

```
<Horizontal.ActualSampleRate>      : Double
```

Returns the effective sample rate of the acquired waveform the digitizer acquires for each channel.

<Horizontal.ActualRecordLength> : Integer

The actual number of samples in the acquired waveform; this number may be less than **numSamples** if the number you request is not available.

```
<Channel.ProbeAttenuation>      : Double
```

Specifies the probe attenuation for the input channel.

```
<Channel.VerticalRange>      : Double
```

Specifies the absolute value of the input range for a channel.

```
<Channel.VerticalOffset>           : Double
```

The offset factor of the given channel; useful for scaling binary data with the following formula

$$voltage = binary\ data * gain\ factor + offset$$

```
<Channel.Gain> : Double
```

The gain factor of the given channel; useful for scaling binary data with the following formula

$$voltage = binary\ data * gain\ factor + offset$$

```
<Channel.Waveform>           : Array of [ActualSamples] Bytes
```

If **ActualSamples** is 0 then there will be nothing here.

```
<Measure>           : Double
```

<Trigger.Received> : Boolean

Return false if no trigger is received.

```
<Trigger.Level> : Double
```

Returns the trigger level used.

When an oscilloscope request is sent with function set to fetch, the oscilloscope will return (on success) the current settings for the oscilloscope:

```
<Fetch>:  <Min Sample rate><Ref Position><Min number of Points><Channel *  
          2><Trigger source><Trigger Slope><Trigger Coupling><Trigger  
          Level><Trigger holdoff><Trigger delay><Trigger Type>
```

```
<Min Sample rate>          : Double  
  
<Ref Position>            : Double  
  
<Min number of Points>    : Integer  
  
<Channel * 2>  
    <Vertical Coupling>    : Integer  
  
    <Vertical Range>      : Double  
  
    <Vertical Offset>     : Double  
  
    <Probe Attenuation>    : Double  
  
<Trigger source>          : String  
  
<Trigger Slope>           : Integer  
  
<Trigger Coupling>        : Integer  
  
<Trigger Level>           : Double  
  
<Trigger Holdoff>         : Double  
  
<Trigger Delay>           : Double  
  
<Trigger Type>            : Integer
```

DIGITAL MULTIMETER

Digital multimeter request

Each element should be separated by space.

<Function><...>

<Function>	Enum
Measure	0
Fetch	1

Fetch: <Nil>

Measure: <Function><Resolution><Range><Auto zero>

<Function>	Enum
DC Volts	0
AC Volts	1
DC Current	2
AC Current	3
Resistance (2 wire)	4
Resistance (4 wire)*	5
Frequency*	6
Period*	7
Diode*	8

** This mode is not supported.*

<Resolution> **: Integer**

What precision to make the measurement. (The higher the resolution the longer the measure time will be).

<Resolution>	Enum
3.5 Digit precision	0
4.5 Digit precision	1
5.5 Digit precision	2
6.5 Digit precision	3

<Range> **: Double**

Maximum range of the measurement. If signal is outside range the result will probably be clamped to range.

Autorange = -1.0

Example 10m = 0.01

<Autozero>	Enum (Integer)
Auto	-1
Off (default)	0
On	1
Once	2

Digital multimeter response

When a digital multimeter request is sent to the server, a result string (when valid) is returned. This string contains the result.

```
<Measure>:           <Instrument ID><Function><Double>

<Instrument ID>      : Integer
<Function>           : Integer

<Fetch>:              <Instrument ID><Function><Functionality><Resolution>
                      <Range><Auto zero>

<Instrument ID>      : Integer
<Function>           : Integer
<Functionality>      : Integer
<Resolution>         : Double
<Range>              : Double
<Auto zero>          : Double
```

EXTENDED PERIPHERALS

Each element should be separated by space.

This is not really an instrument but is addressed in that way. Used for general control over the hardware.

Notice: Maybe some of the function are misplaced and should be in the information interface instead

Extended peripherals Request

<Function><...>

Function	Enum
Delay	0
Temp	1
Self test	2
Reset	3
Calibration	4

Delay: <Delay time>
<Delay time> : Integer
Time to wait in milliseconds

Temp: <Nil>
This will return temperature for DC power supply and function generator.

Self test: <Nil>
This will perform a self test on DC power supply, oscilloscope, digital multimeter and function generator.

Reset: <Nil>
This will reset all instruments and disconnect all cards. This function will run automatically after the server has been idle according to the settings in measuereserver.ini.

Calibration: <Nil>
This will calibrate DC power supply, oscilloscope, digital multimeter and function generator. It is **IMPORTANT** to have all of the connected equipment removed before calibration.

Extended peripherals Response

<Delay>: <Instrument ID><Function>

<Instrument ID> : Integer
<Function> : Integer

<Temp>: <Instrument ID><Function><Temp DC power supply><Temp Function generator>

<Temp DC power supply> : Double
Deliver the core temp in °C
<Temp function generator> : Double

Deliver the core temp in °C

```
<Self test>:          <Instrument ID><Function><Status DC power supply>
                      <Status Oscilloscope><Status Digital
                      multimeter><Status Function generator>

<Status DC power supply>      : String
Contains Passed or Failed
<Status Oscilloscope>        : String
Contains Passed or Failed
<Status Digital multimeter>   : String
Contains Passed or Failed
<Status Function generator>   : String
Contains Passed or Failed

<Reset>:                <Instrument ID><Function>

<Calibration>:           <Instrument ID><Function><Cal DC power supply><Cal
                      Oscilloscope><Cal Digital multimeter><Cal Function
                      generator>

<Cal DC power supply>        : String
Contains Passed or Failed
<Cal Oscilloscope>          : String
Contains Passed or Failed
<Cal Digital multimeter>    : String
Contains Passed or Failed
<Cal Function generator>    : String
Contains Passed or Failed
```

CIRCUIT BUILDER

Each element should be separated by space.

This is not an instrument, it is a function that connects or disconnects either an instrument or a component between two or more nodes.

Circuit builder Request

<Function><...>

<Function>	Enum
Circuit Builder	0
Circuit Switch	1
Clear all cards	2
Circuit Builder Beta	3
Components list	4
Circuit Switch Beta	5
Instruments	6
	7
MD5 comparison	8

<u>Circuit builder:</u>	<Card><\t><Component>
<Card>	: Integer

For which card to build circuit.

```
<Component>           : Integer
```

<u>Circuit Switch:</u>	<Card><Component>
<Card>	: Integer

The card holding the component.

```
<Component> : Integer
```

Describes which component on card to toggle (1-20).

```
Clear all cards:          <Nil>
```

Clear all relays on all cards.

```
Circuit Builder Beta:    <Identifiers>
<Identifiers>           : Integer
```

Unique identifiers corresponding to the component.list “See the Components list”

```
Components list:      <Nil>
```

Returns a list of unique identifiers for all of the components that are present in Components.list.

<u>Circuit Switch Beta:</u>	<Identifiers>
<Identifiers>	: Integer

Unique identifiers corresponding to the component.list “See the Components list”

```
Instruments:          <Instrument connection>
<Instrument connection> : String
```

A string containing Instrument type, Instrument ID, Type of measurement and between which nodes to connect the instrument.

```
MD5 comparison:      <MD5 sum>
<MD5 sum>           : String
```

Comparison of the MD5 sum of the component.list and the MD5 sum given to the function.

Circuit builder Response

<Circuit builder>:	<Instrument ID><Function>
<Instrument ID>	: Integer
<Function>	: Integer
<Circuit Switch>:	<Instrument ID><Function>
<Instrument ID>	: Integer
<Function>	: Integer
<Clear all cards>:	<Instrument ID><Function>
<Instrument ID>	: Integer
<Function>	: Integer
<Circuit Builder Beta>	<Instrument ID><Function>
<Instrument ID>	: Integer
<Function>	: Integer
<Components list>	<InstrumentID><Function><uniqueidentifiers>
<Instrument ID>	: Integer
<Function>	: Integer
<uniqueidentifiers>	: String
<Circuit Switch Beta>	<Instrument ID><Function>
<Instrument ID>	: Integer
<Function>	: Integer
<Instruments>	<Instrument ID><Function>
<Instrument ID>	: Integer
<Function>	: Integer
<MD5 comparison>	<Instrument ID><Function>
<Instrument ID>	: Integer
<Function>	: String

DIGITAL IO

Each element should be separated by space.

Digital IO Request

<Function><...>

Function	Enum (Integer)
Clear all cards	1
Clear multiple cards	3
Build a complete card	9

Clear all cards: <Nil>

Clear all relays on all cards.

Clear multiple cards: <Card mask>

Clear all relays multiple cards.

<Card> : Integer

Which cards to clear.

Build a complete card: <Card><Data>

<Card> : Integer

Which card to build a row on.

<Data> : Integer

Describes which relay(s) on a card to enable.

Digital IO Response

<Clear all cards>: <Instrument ID><Function>

<Instrument ID> : Integer

<Function> : Integer

<Clear multiple cards>: <Instrument ID><Function>

<Instrument ID> : Integer

<Function> : Integer

<Build a complete card>: <Instrument ID><Function>

<Instrument ID> : Integer

<Function> : Integer

CIRCUIT SWITCH

Toggle a component ether on or off (connect/disconnect).

Circuit Switch Request

<Card><Component>

<Card> : Integer
Which card.

<Component> : Integer
Describes which component on card to toggle (1-20).

Circuit Switch Response

<Circuit switch>: **<Instrument ID><Function>**

<Instrument ID> : Integer
<Function> : Integer

DATA TYPE EXPLANATION

Datatype	Representation
Integer	String encoded integer
Double	String encoded double (c locale)
String	String
Enum	Integer, with predefined values
Boolean	String, "0" is false, "1" is true
Bitfield string	String representation of bitfield, "01010101"
Char	One Character only

If nothing else is written, the length of a string is undefined.

SUGGESTED COMMAND ORDER

<SourceSetup> <CircuitSetup> <TransientPaus> <MeasurementSetup>
<SwitchDelay> <SwitchChange> <Measurement>

<SourceSetup> : <Function Generator>/<DC Power Supply>
<CircuitSetup> : <Digital I/O>
<TransientPaus> : <Extended Peripherals>
<MeasuerementSetup> : <Oscilloscope>/<Digital Multimeter>
<SwitchDelay> : <Extended Peripherals>
<SwitchChange> : <Circuit Switch>
<Measuerement> : <Oscilloscope>