

# Serialization Protocol Quick Start

The Touch Sensor Module communicates with messages that are serialized according to the zForce serialization protocol. Here is a quick introduction to the messages you need to send to get started and also how you interpret received messages.

## Encoding Integers

ASN.1 encoded integers, for example values representing scanning frequency or touch active area size, are represented by one or more bytes:

- If the integer is between 0 and 127, it is represented by one byte (00 to 7F).
- If the integer is between 128 and 32767, it is represented by two bytes (00 80 to 7F FF).

The length of the message therefore varies depending on parameter values.

## Enabling Touch Sensor Modules

Do the following to enable a Touch Sensor Module, that is, tell the module to start sending touch notifications.

1. Enable the sensor module by sending a request with an Enable command:

```
EE 09 40 02 02 00 65 03 81 01 00
```

This enables the sensor module to send touch notifications.

2. Read the response. The response should be:

```
EF 09 40 02 02 00 65 03 81 01 00
```

3. The Touch Sensor Module is now Enabled. After this, wait for the sensor module to indicate it has something to send, which means that the device will send a Touch Notification or a BootComplete. A BootComplete indicates that the device has restarted for some reason, so rerun the initialization and enable the sensor module to start receiving touch notifications again.

## Disabling Touch Sensor Modules

Do the following to disable a Touch Sensor Module, that is, tell the module to stop sending touch notifications.

1. Disable the sensor module by sending a request with the following command:

```
EE 08 40 02 02 00 65 02 80 00
```

This disables the sensor module to send touch notifications.

2. Read the response. The response should be:

```
EF 08 40 02 02 00 65 02 80 00
```

3. The Touch Sensor Module is now Disabled.

## Device Configuration

Device configuration is a command that includes different settings for the sensor module, for example the Touch Active Area. When sending a device configuration message, all of the settings specified are not required to be sent, however the response from the sensor module will include the full device configuration message.

This is an example message that changes the touch active area using the Device Configuration command:

```
EE 1A 40 02 02 00 73 14 A2 12 80 02 00 B5 81 01 43 82 02 06 98 83 02 04 34 85 01 FF
```

The message is explained in the table below:

Part	Description
------	-------------

EE 1A	ID for request followed by length of total payload (0x1A = 26 bytes)
40 02 02 00	Device address (always the same for Neonode Touch Sensor Modules)
73 14	ID for Device Configuration followed by the length of the total Device Configuration payload (20 bytes)
A2 12	ID for Sub Touch Active Area followed by the length of Sub Touch Active Area payload
80 02 00 B5	ID for xMin followed by payload length and an integer value (0x00B5 = 181)
81 01 43	ID for yMin followed by payload length and an integer value (0x43 = 67)
82 02 06 98	ID for xMax followed by payload length and an integer value (0x0698 = 1688)
83 02 04 34	ID for yMax followed by payload length and an integer value (0x0434 = 1076)
85 01 FF	ID for "Invert y axis" followed by length of payload and a Boolean (0xFF= True)

The response from the sensor module to the above message will contain the full device configuration message, also the parts not set in the request.

Setting the Touch Active Area should be done before enabling the sensor module with the ENABLE request.

## Reflective Edge Filter

In the Device Configuration command there is something called Reflective Edge filter (from firmware version 1.45 and later). This setting is useful if there is a highly reflective material, right outside of the Touch Active Area. If such conditions are present, enabling this feature could increase the touch performance.

### Command to Enable Reflective Edge Filter

```
EE 09 40 02 02 00 73 03 85 01 80
```

### Command to Disable Reflective Edge Filter

```
EE 09 40 02 02 00 73 03 85 01 00
```

## Setting Frequency

To set the finger frequency to 200 Hz and idle frequency to 63 Hz use a request with the following command:

```
EE 0D 40 02 00 00 68 07 80 02 00 C8 82 01 3F
```



Neonode Touch Sensor Module does not support Stylus mode, and setting the stylus frequency does not do anything.

## Decoding Touch Notifications

A packet can contain from one up to 10 touches, and optionally a timestamp. On packets where the timestamp is not included, the 58 02 TT TT bytes are missing from the end and the length bytes are adjusted accordingly. For One touch (below), F0 15 in the beginning will be F0 11 and A0 0F in the middle will be A0 0B. The same bytes are decreased by 4 for Two and Three touches.

### One Touch

A packet that contains one touch will look like:

```
F0 15 40 02 02 00 A0 0F 42 09 II VV XX XX YY YY GG HH JJ 58 02 TT TT
```

where the data is defined as follows:

Syntax	Meaning
--------	---------

II	ID of this specific touch object. More than one can be tracked simultaneously.
VV	Event: 00 = DOWN (new object). 01 = MOVE (existing object has moved). 02 = UP (existing object is no longer being tracked).
XX XX	X Coordinate of the object. This is in Network Byte Order / Big Endian / Motorola Endian.
YY YY	Y Coordinate of the object. See above.
GG	Size of object on the X Axis.
HH	Size of object on the Y Axis.
JJ	This value must be ignored.
TT TT	Timestamp of the touch.

## Two Touches

A packet that contains two Touches will look like:

```
F0 20 40 02 02 00 A0 1A 42 09 II VV XX XX YY YY GG HH JJ 42 09 II VV XX XX YY YY GG HH JJ 58 02 TT TT
```

where the first "II" and the following bytes up to and including "JJ" are from the first touch, and the second "II" and the following bytes up to and including "JJ" are from the second touch.

## Three Touches

A packet that contains three Touches will look like:

```
F0 2B 40 02 02 00 A0 25 42 09 II VV XX XX YY YY GG HH JJ 42 09 II VV XX XX YY YY GG HH JJ 42 09 II VV XX XX YY YY GG HH JJ 58 02 TT TT
```

For a more thorough explanation of the serialization protocol, refer to [Understanding the zForce ASN.1 Serialization Protocol](#).

## Read More about the Communication Protocol

- [Serialization Protocol Quick Start](#)
- [USB HID Transport](#)
- [I2C Transport](#)
- [zForce Message Specification](#)
- [Understanding the zForce ASN.1 Serialization Protocol](#)

## Read More

- [Introduction](#)
- [Getting started with Touch Sensor Module Evaluation](#)
- [Getting Started with Software Integration](#)
- [Mechanical Integration](#)
- [Electrical Integration](#)
- [Software Integration](#)
- [Implementation Examples](#)
- [Specifications](#)
- [Legal Notice](#)