Developer Mode MIDI specification

This document describes the MIDI API for the **Exquis Developer Mode**, which allows direct communication with the device via MIDI *SysEx* and *channel* messages to control its settings, receive data, and extend or replace its functionality. Once **Developer Mode** is enabled, it allows you to receive input events from Exquis and to control what is displayed on its LEDs. It is intended for tinkerers who want to customize the way Exquis works. This is the API used by the official Exquis MIDI Remote Script for Ableton Live. For technical issues, contact support at dualo.com/en/support.

MIDI SysEx messages

Exquis will only respond to Developer Mode commands sent to its USB-MIDI port. All follow the same basic format:

```
F0 00 21 7E 7F id [...] F7
```

- `id` (1 byte): Specifies the command.

Identifier (hex)	Command
[00](#cmd00)	Setup Developer Mode
[01](#cmd01)	Use custom scale list
[02](#cmd02)	Color palette
[03](#cmd03)	Refresh
[04](#cmd04)	Set LED color
[05](#cmd05)	Tempo
[06](#cmd06)	Root note
[07](#cmd07)	Scale number
[08](#cmd08)	Custom scale
[09](#cmd09)	Snapshot

The rest of the message is specific to each command.

Important: All commands except for `00` (*setup*) are only available when Developer Mode is active.

Setup Developer Mode (00h) {#cmd00}

Set up Exquis to select which physical zones to take over in Developer Mode while the rest of the device functions as usual:

Bit	Mask (hex)	Zone
0	01	Pads
1	02	Encoders
2	04	Slider
3	08	Up/Down buttons
4	10	Settings/Sound buttons
5	20	All other buttons

As the zones are specified as a bit mask, you can combine them by adding the values together. For example, to take over everything except the Settings/Sound buttons, you would use `2F` as a mask.

Once a zone is taken over by the Developer Mode, Exquis will stop processing its inputs normally (except during native menu interaction in the case of the Settings/Sound buttons). Instead, it will directly send input events through USB-MIDI, and it will accept LED control messages related to it (see the MIDI channel messages section below).

Thus, **to enter Developer Mode**, just send a *setup* command using the mask of the zones you want to take over. **To exit Developer Mode**, send a *setup* command with a mask of `00`.

Request: `F0 00 21 7E 7F 00 mask F7`
- `mask` (1 byte): The bit mask of the zones taken over by Developer Mode (see the table above).
Response (Exquis): None

Use custom scale list (01h) {#cmd01}

In the Settings menu, Exquis usually allows you to select a scale from its internal list. To facilitate synchronization with DAWs and other programs, you can instead make Exquis use a custom list of scales using this message and specifying the number of scales that can be selected in the Settings menu. Upon any scale change, Exquis will send a MIDI SysEx *Scale number* message to notify the host of the index of the newly selected scale in the list. The developer can then respond by sending a *Custom scale* message to send the currently used scale to Exquis.

Request: `F0 00 21 7E 7F 01 [count] F7`
- `[count]` (1 byte, optional): The number of scales in the custom list.
If `count` is omitted, Exquis will revert to using its internal list.
Response (Exquis): None

Color palette (02h) {#cmd02}

All the LEDs on Exquis can be controlled either through SysEx messages (see the *Set LED Color* message) or through MIDI CC messages (see the corresponding section). In the latter case, a palette is used to specify the colors. This message allows you to get and set the color palette used by Exquis for MIDI CC messages.

Get the entire color palette

Request: `F0 00 21 7E 7F 02 F7`
Response (Exquis): `F0 00 21 7E 7F 02 color(0) ... color(127) F7`
- `color(n)` (3 bytes): The color at the index `n` in the palette, in the RGB format:
  - `red` (1 byte): The red component of the color (0…127).
  - `green` (1 byte): The green component of the color (0…127).
  - `blue` (1 byte): The blue component of the color (0...127).

Get a single color from the palette

Request: `F0 00 21 7E 7F 02 index F7`
- `index` (1 byte): The index of the color in the palette (0...127).
Response (Exquis): `F0 00 21 7E 7F 02 index red green blue F7`
- `index` (1 byte): The index of the color in the palette (0…127).
- `red` (1 byte): The red component of the color (0…127).
- `green` (1 byte): The green component of the color (0...127).
- `blue` (1 byte): The blue component of the color (0…127).

Set colors in the palette

Request: `F0 00 21 7E 7F 02 start_index color(0) [... color(N)] F7`
- `start_index` (1 byte): The index of the first color to set (0…127).
- `color(n)` (3 bytes): The color at the index `start_index + n` in the palette, in the RGB format:
  - `red` (1 byte): The red component of the color (0...127).
  - `green` (1 byte): The green component of the color (0...127).
  - `blue` (1 byte): The blue component of the color (0…127).
Response (Exquis): None

Refresh (03h) {#cmd03}

This command is sent either by the host or Exquis to request a refresh of the LED display from the other part. Notably, Exquis sends a Refresh command when entering and returning from settings. You may want to process this command in your application, as well as send it when needed, to ensure that Exquis is always displaying the correct information.

Request: `F0 00 21 7E 7F 03 [settings_page] F7`
- `settings_page` (1 byte, optional): When sent by Exquis, specifies the page number of the settings menu that has just been left (`7F` when entering the settings menu).
Response: None

Set LED color (04h) {#cmd04}

This command allows you to directly set the color of any contiguous set of LEDs on Exquis, without passing through the palette.

Request: `F0 00 21 7E 7F 04 start_id color(0) [... color(N)] F7`
- `start_id` (1 byte): The ID of the first LED to set (see [table](#led-id)).
- `color(n)` (4 bytes): The complete color of the LED identified by the number `start_id + n` in the palette, in the RGB format, with an *effects* component:
  - `red` (1 byte): The red component of the color (0...127).
  - `green` (1 byte): The green component of the color (0…127).
  - `blue` (1 byte): The blue component of the color (0…127).

- `fx` (1 byte): The effect applied to the color (see [LED effects](#led-fx)).
Response (Exquis): None

Tempo (05h) {#cmd05}

Allows you to synchronize Exquis with the tempo of your DAW or other software. This command is sent by Exquis each time the tempo is changed in the settings while Developer Mode is active.

Tempo value format

The tempo is expressed in BPM (beats per minute) as a positive integer over 2 bytes:
- `tempo[0]`: The most significant bit of the tempo value.
- `tempo[1]`: The seven least significant bits of the tempo value.

Examples:
- *120 BPM* is encoded as `00 78`.
- *200 BPM* is encoded as `01 48`.

Get the tempo of Exquis

Request: `F0 00 21 7E 7F 05 F7`
Response (Exquis): `F0 00 21 7E 7F 05 tempo F7`
- `tempo` (2 bytes): The tempo in BPM.

Set the tempo of Exquis

Request: `F0 00 21 7E 7F 05 tempo F7`
- `tempo` (2 bytes): The tempo in BPM (20…240).
Response (Exquis): None

Root note (06h) {#cmd06}

Allows you to synchronize Exquis with the root note of your DAW or other software. This command is sent by Exquis each time the root note is changed in the settings while Developer Mode is active.

Get the root note of Exquis

Request: `F0 00 21 7E 7F 06 F7`
Response (Exquis): `F0 00 21 7E 7F 06 note F7`
- `note` (1 byte): The root note (0…11, with 0 == *C*, 1 == *C#*, ..., 11 == *B*).

Set the root note of Exquis

Request: `F0 00 21 7E 7F 06 note F7`
- `note` (1 byte): The root note (0…11, with 0 == *C*, 1 == *C#*, ..., 11 == *B*).
Response (Exquis): None

Scale number (07h) {#cmd07}

This command is sent by Exquis each time the scale is changed in the settings while Developer Mode is active. It allows you to synchronize Exquis with the scale of your DAW or other software.

Get the scale number of Exquis

Request: `F0 00 21 7E 7F 07 F7`
Response (Exquis): `F0 00 21 7E 7F 07 scale F7`
- `scale` (1 byte): The scale number (0...127).

Set the scale number of Exquis

Request: `F0 00 21 7E 7F 07 scale F7`
- `scale` (1 byte): The scale number (0…127).
Response (Exquis): None

Custom scale (08h) {#cmd08}

With this command, Exquis can use a custom scale instead of the predefined scales, for the duration of the Developer Mode.

Get the current scale degrees of Exquis

Request: `F0 00 21 7E 7F 08 F7`
Response (Exquis): `F0 00 21 7E 7F 08 degree(0) ... degree(11) F7`
- `degree(n)` (1 byte): Whether the scale has the note at the degree of index `n` (0…1).

Set the current scale degrees of Exquis

Request: `F0 00 21 7E 7F 08 degree(0) ... degree(11) F7`
- `degree(n)` (1 byte): Whether the scale has the note at the degree of index `n` (0...1).
Response (Exquis): None

Snapshot (09h) {#cmd09}

This command allows you to save and retrieve the current layout and the MIDI settings of Exquis. For instance, it is used by the Ableton Live MIDI Remote Script to save and restore the state of Exquis depending on the selected track.

Get a snapshot of Exquis

Request: `F0 00 21 7E 7F 09 F7`
Response (Exquis): `F0 00 21 7E 7F 09 snapshot F7`
- `snapshot` (255 bytes): The snapshot data of Exquis.

Restore a snapshot of Exquis

Request: `F0 00 21 7E 7F 09 snapshot F7`
- `snapshot` (255 bytes): The snapshot data of Exquis to restore.
Response (Exquis): None

MIDI channel messages

In addition to SysEx messages, in Developer Mode, Exquis sends and receives MIDI channel messages to change the LEDs and transmit input events. MIDI channel 16 is reserved for this purpose.

Set LED color from palette

You can set the color of any control on Exquis by sending a MIDI CC message on channel 16 to its identifier.

Request: `BF id palette_index` (CC ch.16)
- `id` (1 byte): The identifier of the control (see [table](#led-id)).
- `palette_index` (1 byte): The index of the color in the palette (0…127).

For convenience, a MIDI Note On/Off message can also be used to set the color of a pad.

Set LED color effect

You cann set the [LED effect](#led-fx) of any control on Exquis by sending a MIDI Polyphonic Key Pressure (a.k.a. poly aftertouch) message on channel 16 to its identifier.

Request: `AF id fx` (Poly AT ch.16)
- `id` (1 byte): The identifier of the control (see [table](#led-id)).
- `fx` (1 byte): The effect applied to the color (see [LED effects](#led-fx)).

MIDI input events

In Developer Mode, Exquis sends MIDI channel messages to channel 16 to transmit its input events.

Pad pressed: `9F pad 7F` (Note On ch.16)
- `pad` (1 byte): The identifier of the pad (0…60, see [table](#led-id)).

Pad released: `8F pad 00` (Note Off ch.16)
- `pad` (1 byte): The identifier of the pad (0...60, see [table](#led-id)).

Button pressed/released: `BF button state` (CC ch.16)
- `button` (1 byte): The identifier of the button (100…109, see [table](#led-id)).
- `state` (1 byte): `7F` when pressed, `00` when released.

The same message is used for encoder push buttons and slider portions.

Encoder turned: `BF encoder delta` (CC ch.16)
- `encoder` (1 byte): The identifier of the encoder (110…113, see [table](#led-id)).
- `delta` (1 byte): The relative amount of displacement since the last message.

To interpret the `delta` value, remove 64 from it to get a relative number of steps turned clockwise (positive) or counterclockwise (negative).

Slider touched: `BF 90 portion` (CC ch.16)
- `portion` (1 byte): The portion of the slider currently touched (0..5 or 127 if untouched).

Highlighting MIDI notes on Exquis

Independently of the Developer Mode, Exquis can highlight MIDI notes on its pads. This is done by sending MIDI Note On or Off messages to Exquis on channel 1 through the USB-MIDI port.

Reference

Identifying individual elements {#led-id}

Almost each individual control on Exquis has a LED associated to it and can be referenced by an identifier:

Identifiers (decimal)	Description	Count
0..60	Pads (0=bottom left, 60=top right)	61
80..85	Individual slider portions	6
90	Slider position (127 = untouched)	1
100	Settings button	1
101	Sound button	1
102	Record button	1
103	Loop button	1
104	Clips button	1
105	Play/Stop button	1
106	Down button	1
107	Up button	1
108	Undo button	1
109	Redo button	1
110..113	Encoders	4
114..118	Encoder buttons	4

This identifier will be used in the SysEx and channel message to specify which LED to control, or which individual control the message pertains to.

LED effects {#led-fx}

Aside from a RGB color or a palette index, you can assign an effect to any LED in Exquis, allowing you to free up the MIDI bandwidth while providing dynamic feedback. An effect is specified within a single-byte 7-bit value, and only one of the following effects can be used for a single LED (values in hexadecimal):
- *No effect*: `00`
- *Pulsate to black*: `3F`
- *Pulsate to white*: `7F`
- *Pulsate to red*: `3E`
- *Pulsate to green*: `7E`
- *Alpha channel*: `00` (100% opaque) to `3D` (100% transparent)
- *Blend to white*: `40` (0% white) to `7D` (100% white)

The *pulsate* effects are synchronized with the current tempo.

Changelog

- 2025-03-10: Initial release.