

Controller Deck Documentation



Contents

- General features 2**
 - Downloads 2
 - Connection 2
 - IO Ports 2
 - On-board Controls 8
 - TouchDeck 13
 - MIDI Protocol 13
 - Icons Protocol 24
 - Pure Audio Device 30

The **Controller Deck**, by **Lambdaconcept**, is a hardware MIDI control surface that connects to a computer over a single USB-C cable. It is designed for live performance, video/audio production and any workflow that benefits from a dedicated set of physical controls mapped to software parameters.

In addition to being a MIDI controller, the Controller Deck integrates a gigabit Ethernet port, HDMI/SDI video capture, a USB-C DisplayPort Alt Mode video output and a USB audio interface, turning it into a compact, all-in-one control and A/V hub.



General features

- Connects to the computer over a **USB-C cable** with PD delivery, providing power to the computer and a high-speed data connection.
- Full-size **MIDI controller** with 11 faders, each with its own customisable icon/label screen, 11 knobs, 10 pairs of RGB illuminated A/B buttons, 10 RGB illuminated knobs and 10 RGB backlit pads.
- Dedicated **TouchDeck** grid of 40 individually addressable buttons, each with its own customizable image, organized in a 5 rows x 8 columns grid. The grid has multiple pages that can be navigated using the TouchDeck grid pager.
- Two round **touchscreens** for specialized interactions such as a crossfader, color picker, or TouchDeck grid pager.
- Built-in **video capture** (HDMI or SDI) and **video output** (USB-C DisplayPort Alt Mode), allowing the device to sit in the video signal path.
- Integrated **USB audio interface** (768 kbit/s stereo soundcard) for routing audio to and from the computer, without the need for an additional USB audio device.
- Integrated **Gigabit Ethernet** port providing a reliable wired network connection to the connected computer.
- Every control surface reports its state as standard **USB MIDI** events, and the on-board screens (TouchDeck grid screen and fader labels) are updated through a **custom USB protocol**.

Downloads

Download this User Manual in PDF format:

[controller.pdf](#)

Connection

1. **Power 24V**
XXX
2. **Connect USB-C**
XXX

IO Ports

The Controller Deck exposes the following physical connectors, all accessible on the rear or left side panel of the device. Each port is described below with its role and, when relevant, its electrical or protocol characteristics.

Power (24V)



The Controller Deck is powered externally through a dedicated 24V DC Jack barrel connector. This input powers the whole unit, and can also provide power or charge the computer connected over the USB-C connection [if the computer supports USB Power Delivery (PD)].

Voltage	24V DC
Purpose	Main power supply for the whole device
Purpose	Can provide power to the connected computer over USB-C

Note

The device cannot be powered from USB-C alone and requires a 24V power supply to operate. The USB-C connection is used for data transfer and can provide power to the connected computer, but it does not power the Controller Deck itself.

USB-C (Host PC connection)



The USB-C port is used to connect the Controller Deck to the computer. It carries:

USB Power Delivery (PD)	Optionally provide power to the connected computer.
USB MIDI interface	Control surface events, see MIDI Protocol .
USB Bulk interface	Update the TouchDeck grid screen and the fader label screens, see Icons Protocol .
USB Audio interface	Described below in Audio Input/Output (USB sound card) .
USB Video Class (UVC) video capture interface	Described below in Video Input (HDMI / SDI) .
DisplayPort Alt Mode video output interface	Described below in Video Output (USB-C DisplayPort Alt Mode) .

Ethernet (Gigabit)



A standard RJ45 Gigabit Ethernet port. It can be used to provide a reliable wired network connection to the connected computer, without the need for an additional USB-to-Ethernet adapter.

Connector	RJ45
Speed	2.5 Gbps (Gigabit)

Audio Input/Output (USB sound card)



The Controller Deck integrates a stereo USB sound card, exposed to the host as a standard USB Audio Class device, for both playback and recording.

Class	USB Audio Class (UAC)
Bitrate	768 kbit/s stereo
Product name	PureAudio
Audio input	2x 3.5mm TS jack (L + R)
Audio output	2x 3.5mm TS jack (L + R)

See full details about the audio input/output capabilities in [Pure Audio Device](#).

Video Input (HDMI / SDI)



The Controller Deck can capture an external video source over either **HDMI** or **SDI**. The captured video is exposed to the host computer as a standard USB video capture device (UVC), so it can be used directly in any video conferencing, streaming or production software as a webcam source.

Connectors	HDMI, SDI
Exposed as	USB Video Class (UVC) capture device

Video Output (USB-C DisplayPort Alt Mode)



Using the DisplayPort Alt Mode capability of USB-C, the Controller Deck can also output video to a monitor or projector.

Connector	USB-C (DisplayPort Alt Mode)
-----------	------------------------------

On-board Controls

This section describes every physical control built into the Controller Deck. Each group of controls reports its state to the host as USB MIDI events, detailed in [MIDI Protocol](#).

Faders row



11 linear faders, typically used as 10 channel faders plus 1 master channel fader. Each fader has its own small screen below it, which can be used to display a custom icon or label for the fader.

Quantity	11
Typical usage	10 channels + 1 master channel
MIDI mapping	see MIDI Protocol
Screen label update	see Icons Protocol

Knobs row



11 rotary knobs, one per fader/channel. When turned, the knob level is reported on the small screen below the fader.

Quantity	11
Type	Endless rotary potentiometer
MIDI mapping	see MIDI Protocol

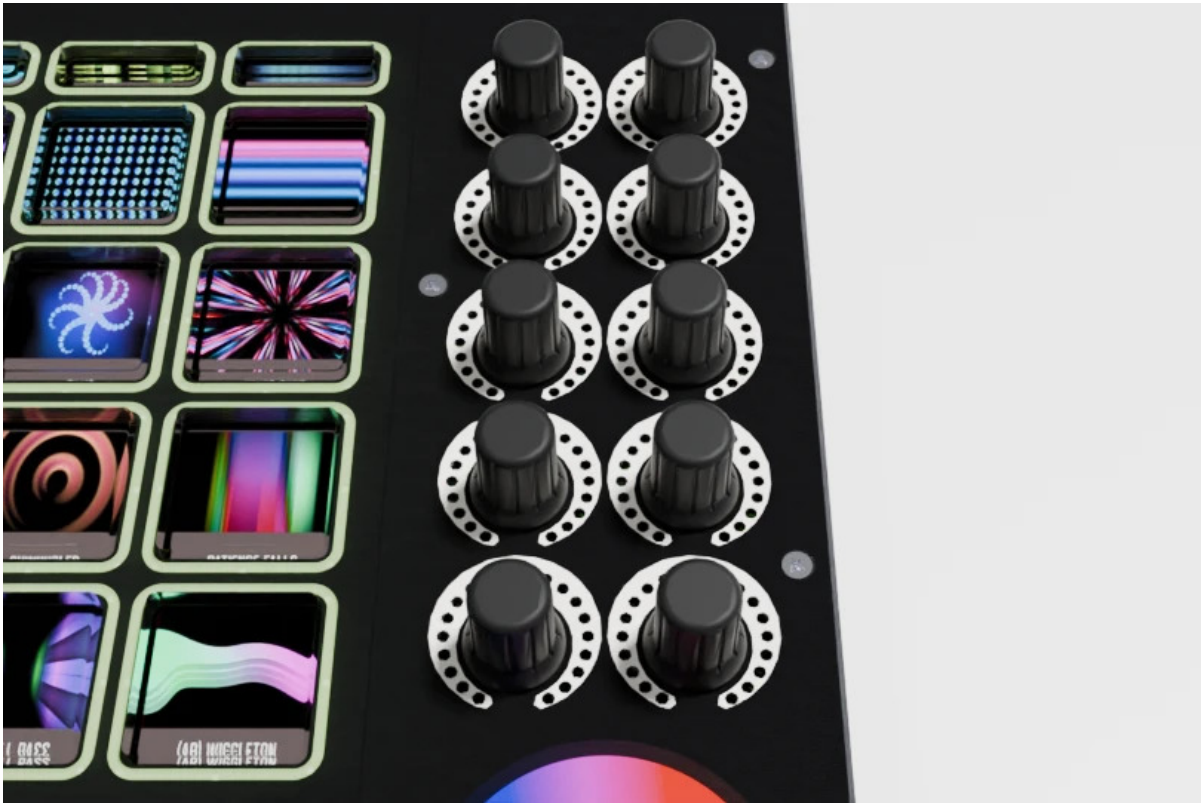
A/B - B/S Buttons row



10 pairs of buttons, configurable as A/B or B/S (Bank/Solo) function buttons, each with an individually addressable RGB backlight.

Quantity	10 pairs (20 buttons)
Backlight	RGB, individually addressable
MIDI mapping	see MIDI Protocol
LED color reference	see LED color control

RGB Knobs array



10 rotary knobs, organised in a 5 rows x 2 columns array, each surrounded by an RGB LED ring used as a visual position/state indicator.

Quantity	10 (5 rows x 2 columns)
Indicator	RGB LED ring
MIDI mapping	see MIDI Protocol
LED color reference	see LED color control

Pad Buttons array



10 backlit pad buttons, organised in a 5 rows x 2 columns array, with individually addressable RGB backlight, typically used to trigger clips, samples or scenes.

Quantity	10 (5 rows x 2 columns)
Backlight	RGB, individually addressable
MIDI mapping	see MIDI Protocol
LED color reference	see LED color control

Control Buttons



4 dedicated utility buttons with RGB backlight:

- **x2** – double the current tempo/rate.
- **/2** – halve the current tempo/rate.
- **TAP** – tap tempo.
- **RESYNC** – force a resynchronization.

Quantity	4
Backlight	RGB, individually addressable
MIDI mapping	see MIDI Protocol
LED color reference	see LED color control

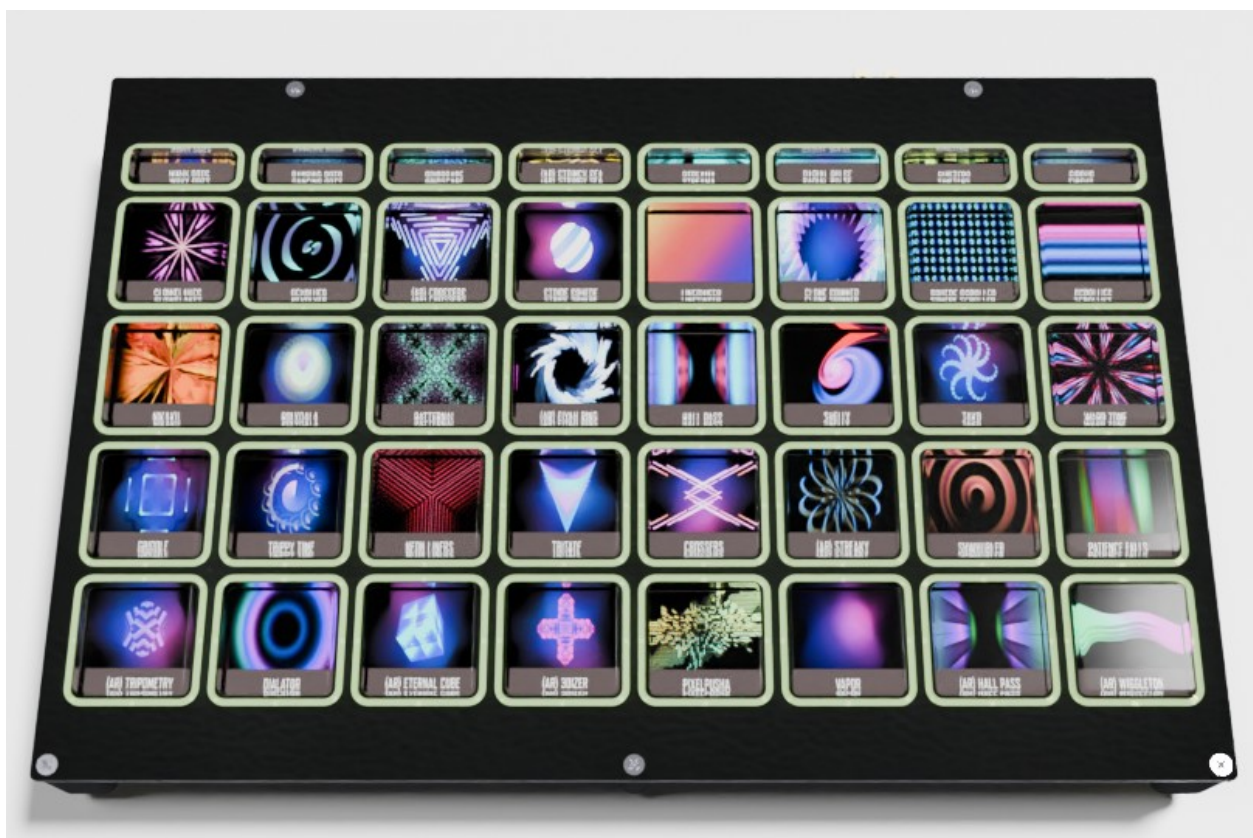
Round Touchscreens



2 round touchscreens that can be assigned to special interactive widgets, such as a crossfader or a color picker, configurable through MIDI SysEx messages.

Quantity	2
Shape	Round
Example widgets	Crossfader, color picker, TouchDeck grid pager, X/Y axis controller.
Configuration	MIDI SysEx, see MIDI Protocol

TouchDeck



The TouchDeck is a dedicated grid of 5 rows x 8 columns (40 buttons in total) touch-sensitive screen buttons. Each button displays its own customizable image, updated through the icons protocol described in [Icons Protocol](#).

In order to support a larger number of buttons, the TouchDeck has a built-in pager feature that can be used to scroll vertically and horizontally. The pager current line and column position can be controlled through MIDI messages, or interactively by using the Pager widget of the round touchscreens, described in [On-board Controls](#).

Grid size	5 rows x 8 columns (40 buttons)
Per-button image	Yes, individually customizable
Horizontal paginated columns	128 (16 grids of 8 columns each)
Vertical paginated lines	16 (the first row is always fixed)
MIDI mapping	see MIDI Protocol
Icons update	See Icons Protocol

MIDI Protocol

All the physical controls of the Controller Deck (faders, knobs, buttons, pads and the TouchDeck grid) are reported to the host as standard **USB MIDI** events. The device also accepts MIDI messages sent by the host to control the buttons/knobs levels and their RGB backlight, or to configure the touchscreens widgets.

USB device identification

The MIDI interface is exposed on the main `TouchDeck` USB device:

Field	Value
bcdUSB	2.00 High Speed
Vendor ID	0xFFFF
Product ID	0x1234
iManufacturer	LambdaConcept
iProduct	TouchDeck

USB Descriptor details ...

Bus 001 Device 092: ID ffff:1234 LambdaConcept TouchDeck

Negotiated speed: High Speed (480Mbps)

Device Descriptor:

```

bLength          18
bDescriptorType   1
bcdUSB           2.00
bDeviceClass      0 [unknown]
bDeviceSubClass   0 [unknown]
bDeviceProtocol   0
bMaxPacketSize0   64
idVendor          0xffff LambdaConcept
idProduct         0x1234 TouchDeck
bcdDevice         1.05
iManufacturer     1 LambdaConcept
iProduct          2 TouchDeck
iSerial           3 00000000
bNumConfigurations 1

```

Configuration Descriptor:

```

bLength          9
bDescriptorType   2
wTotalLength      0x006c
bNumInterfaces    2
bConfigurationValue 1
iConfiguration    0
bmAttributes      0x80
    (Bus Powered)
MaxPower          500mA

```

Interface Descriptor:

```

bLength          9
bDescriptorType   4
bInterfaceNumber  0
bAlternateSetting 0
bNumEndpoints    2
bInterfaceClass   1 Audio
bInterfaceSubClass 3 MIDI Streaming
bInterfaceProtocol 0
iInterface        0

```

(continues on next page)

(continued from previous page)

MIDIStreaming Interface Descriptor:

```

bLength          7
bDescriptorType   36
bDescriptorSubtype 1 (HEADER)
bcdADC           1.00
wTotalLength     0x0043

```

MIDIStreaming Interface Descriptor:

```

bLength          6
bDescriptorType   36
bDescriptorSubtype 2 (MIDI_IN_JACK)
bJackType        1 Embedded
bJackID          1
iJack            4 Grid

```

MIDIStreaming Interface Descriptor:

```

bLength          6
bDescriptorType   36
bDescriptorSubtype 2 (MIDI_IN_JACK)
bJackType        1 Embedded
bJackID          3
iJack            5 Controller

```

MIDIStreaming Interface Descriptor:

```

bLength          9
bDescriptorType   36
bDescriptorSubtype 3 (MIDI_OUT_JACK)
bJackType        1 Embedded
bJackID          2
bNrInputPins     1
baSourceID( 0)   1
BaSourcePin( 0)  1
iJack            4 Grid

```

MIDIStreaming Interface Descriptor:

```

bLength          9
bDescriptorType   36
bDescriptorSubtype 3 (MIDI_OUT_JACK)
bJackType        1 Embedded
bJackID          4
bNrInputPins     1
baSourceID( 0)   3
BaSourcePin( 0)  1
iJack            5 Controller

```

Endpoint Descriptor:

```

bLength          9
bDescriptorType   5
bEndpointAddress 0x01 EP 1 OUT
bmAttributes     2
  Transfer Type   Bulk
  Synch Type      None
  Usage Type      Data
wMaxPacketSize   0x0200 1x 512 bytes
bInterval        0
bRefresh         0

```

(continues on next page)

(continued from previous page)

```

bSynchAddress          0
MIDIStreaming Endpoint Descriptor:
  bLength               6
  bDescriptorType       37
  bDescriptorSubtype    1 (Invalid)
  bNumEmbMIDIJack       2
  baAssocJackID( 0)     1
  baAssocJackID( 1)     3
Endpoint Descriptor:
  bLength               9
  bDescriptorType       5
  bEndpointAddress      0x81  EP 1 IN
  bmAttributes          2
    Transfer Type       Bulk
    Synch Type          None
    Usage Type          Data
  wMaxPacketSize        0x0200  1x 512 bytes
  bInterval             0
  bRefresh              0
  bSynchAddress         0
MIDIStreaming Endpoint Descriptor:
  bLength               6
  bDescriptorType       37
  bDescriptorSubtype    1 (Invalid)
  bNumEmbMIDIJack       2
  baAssocJackID( 0)     2
  baAssocJackID( 1)     4
Interface Descriptor:
  [... Custom bulk interface ...]

```

USB MIDI interface

The Controller Deck enumerates as a standard USB Audio Class 1.0 MIDI streaming interface (Interface 0), exposing **two independent MIDI jacks**:

bInterfaceNumber	bNumEndpoints	bInterfaceClass	bInterfaceSubClass	bInterfaceProtocol
0	2	Audio	MIDI Streaming	0

This interface offers one bidirectional endpoint for MIDI communication with the host.

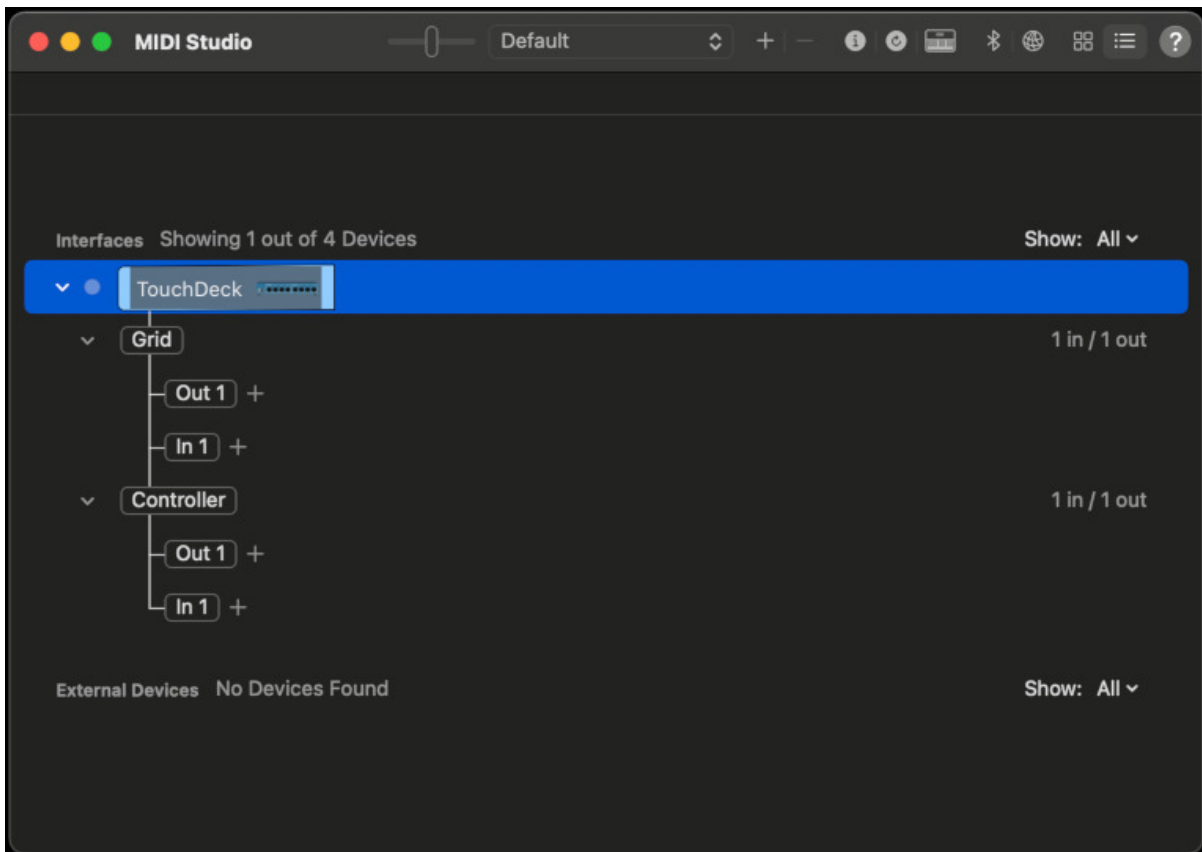
Endpoint	Direction	Purpose
1	IN/OUT	MIDI Streaming

Jack index	Name	Content
0	TouchDeck Grid	Events and LED control for the 40-button TouchDeck grid.
1	TouchDeck Controller	Events and LED control for every other control: faders, knobs, A/B - B/S buttons, pad buttons, control buttons and the wid-gets touchscreens (SysEx).

Manufacturer name reported by the device: `LambdaConcept`.



View MIDI Devices



USB-MIDI packet format

Every event, in either direction, is carried using the standard 4-byte USB-MIDI packet format:

Byte number	00	01	02	03
Description	Cable Number Code Index Number (CIN)	MIDI status byte	MIDI data byte 1	MIDI data byte 2
	Cable Number: (0 = Grid, 1 = Controller).	Message type Channel	Note	Velocity / Value

CIN	Message
0x8	Note Off
0x9	Note On
0xB	Control Change
0x4	SysEx starts or continues (3 data bytes follow)
0x5 / 0x6 / 0x7	SysEx ends with 1 / 2 / 3 data bytes

Control mapping

Every physical control group is mapped to a range of MIDI note numbers or controller (CC) numbers, all on **channel 0**. Groups that have a backlight also report a `led_channel`, which is the MIDI channel to use when sending LED color/mode messages back to the device (see [LED color control](#) below).

Control	Jack	Message	Range	LED channel
TouchDeck grid (40 buttons)	Grid (0)	Note On / Note Off	0-39	0
RGB knobs	Controller (1)	Control Change	0-9	1
Knobs	Controller (1)	Control Change	10-20	-
Faders	Controller (1)	Control Change	21-31	-
A/B - B/S buttons	Controller (1)	Control Change	32-51	1
Pad button + Control buttons (x2, /2, TAP, RESYNC)	Controller (1)	Control Change	72-85	1
Touchscreens widgets	Controller (1)	Control Change	86-127	-

Event examples

The following examples show the raw 4-byte USB-MIDI packet produced by the device when a control is used.

Event	Packet (hex)	Description
Grid button 5 pressed	09 90 05 7F	Cable 0 (Grid), Note On, channel 0, note 5, velocity 127
Grid button 5 released	08 80 05 00	Cable 0 (Grid), Note Off, channel 0, note 5, velocity 0
RGB knob #0 (<code>k1a</code>) turned	1B B0 00 40	Cable 1 (Controller), CC, channel 0, controller 0, value 64
Knob #0 (<code>knr</code> , CC 10) turned	1B B0 0A 64	Cable 1 (Controller), CC, channel 0, controller 10, value 100
Fader #0 (<code>fdr</code> , CC 21) moved	1B B0 15 7F	Cable 1 (Controller), CC, channel 0, controller 21, value 127
A/B button (<code>bsr</code> , CC 32) pressed	1B B0 20 7F	Cable 1 (Controller), CC, channel 0, controller 32, value 127
Pad button (<code>bpa</code> , CC 72) pressed	1B B0 48 7F	Cable 1 (Controller), CC, channel 0, controller 72, value 127
TAP control button (<code>drt</code> , CC 86) pressed	1B B0 56 7F	Cable 1 (Controller), CC, channel 0, controller 86, value 127

LED color control

The RGB backlight of the TouchDeck grid buttons (`grid`), RGB knobs (`k1a`), A/B - B/S buttons (`bsr`) and pad buttons (`bpa`) is controlled by sending a **Note On** message to the device, on the `led_channel` given in the mapping table above.

Field	Meaning
Message	Note On
Note	Button/knob index (0-39)
Velocity bits [0:4]	Palette color index (0-31)
Velocity bits [5:6]	Mode

Mode value	Mode
0	SOLID
1	BLINKING
2	FADING
3	TURNING

The velocity byte is therefore computed as `(mode << 5) | color_index`.

32 colors palette

RingLight Color Palette

0 0x000000	1 0x000000	2 0x000000	3 0x000000	4 0x505050	5 0xFFFFFFFF	6 0x7F2626	7 0xFF0000
8 0x803713	9 0xFF2A00	10 0x7F5226	11 0xFF7F00	12 0x7F7F26	13 0xFFFF00	14 0x527F26	15 0x7FFF00
16 0x267F26	17 0x00FF00	18 0x267F44	19 0x00FF55	20 0x267F7F	21 0x00FFFF	22 0x26437F	23 0x0054FF
24 0x26267F	25 0x0000FF	26 0x874CFF	27 0x5400FF	28 0x7F2661	29 0xFF00AA	30 0x7F2635	31 0xFF002A

Idx	RRGGBB	Idx	RRGGBB	Idx	RRGGBB
0	0x000000	1	0x000000	2	0x000000
3	0x000000	4	0x505050	5	0xFFFFFF
6	0x7F2626	7	0xFF0000	8	0x803713
9	0xFF2A00	10	0x7F5226	11	0xFF7F00
12	0x7F7F26	13	0xFFFF00	14	0x527F26
15	0x7FFF00	16	0x267F26	17	0x00FF00
18	0x267F44	19	0x00FF55	20	0x267F7F
21	0x00FFFF	22	0x26437F	23	0x0054FF
24	0x26267F	25	0x0000FF	26	0x874CFF
27	0x5400FF	28	0x7F2661	29	0xFF00AA
30	0x7F2635	31	0xFF002A		

LED example

Set grid button 5 to solid red (palette index 7, mode `SOLID` = 0):

```
velocity = (SOLID << 5) | 7 = (0 << 5) | 7 = 0x07
```

Packet: 09 90 05 07

```

| | | |
| | | --- velocity: color index 7 (red), mode SOLID
| | +----- note: button index 5
| +----- Note On, channel 0
+----- Cable 0 (Grid), CIN 0x9 (Note On)
```

Set A/B button, relative index 0 within the `bsr` range, to fading cyan (palette index 20, mode `FADING` = 2), on the `bsr` LED channel (1):

```
velocity = (FADING << 5) | 20 = (2 << 5) | 20 = 0x54
```

Packet: 19 91 00 54

Pager control

This is the matrix representation with MIDI note indexes:

Horizontal page scrolling:

Channel: 0 0 0 0 0 0 0 0 1 1 1 ..

Matrix: +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

position | 0  1  2  3  4  5  6  7 |           0  1  ..
          | 8  9 10 11 12 13 14 15 |           8  9  ..
          | 16 17 18 19 20 21 22 23 |  --> 16 17  ..
          | 24 25 26 27 28 29 30 31 |           24 25  ..
```

(continues on next page)

(continued from previous page)

```

      | 32 33 34 35 36 37 38 39 |      32 33 ..
      +-----+
Vertical          |
line scrolling:    v

      40 41 42 43 44 45 46 47
      ..
      .. .. .. .. .. .. ..
      ..
      120 121 122 123 124 125 126 127

```

To set the current line of the TouchDeck grid:

Set the current line:

```

mtype:    PROG_CHANGE (0xC)
channel:   0
note:     0-LINES_MAX_SCROLL    (new line index)
value:     N/A

```

Where LINES_MAX_SCROLL = 11 (12 possible values)

Note

12 scroll positions + 4 lines = 16 lines in total

Set the current column:

```

mtype:    PROG_CHANGE (0xC)
channel:   1
note:     0-COLUMNS_MAX_SCROLL (new column index)
value:     N/A

```

COLUMNS_MAX_SCROLL = 119 (120 possible values)

Note

120 scroll positions + 8 columns = 128 columns in total

Sending LED / MIDI messages via the control endpoint

If the standard MIDI interface is already claimed by other software, the same 4-byte USB-MIDI packet can also be delivered to the device through a **vendor control transfer**, “Set MIDI” (request 5), on the control endpoint (see [Icons Protocol](#) for the full list of vendor requests):

Field	Value	Computed from packet [b0, b1, b2, b3]
bRequest	5	–
wIndex	(b0 << 8) b1	Cable/CIN byte and status byte
wValue	(b2 << 8) b3	Data byte 1 and data byte 2

Example, using the same “grid button 5 solid red” packet (09 90 05 07) as above:

```
wIndex = (0x09 << 8) | 0x90 = 0x0990
wValue = (0x05 << 8) | 0x07 = 0x0507
```

```
async setCustomMIDI(device, midi32) {
  const value = (midi32[2] << 8) | midi32[3];
  const index = (midi32[0] << 8) | midi32[1];
  await device.controlTransferOut({
    requestType: "vendor",
    recipient: "device",
    request: 5,
    value: value,
    index: index,
  }, new Uint8Array(0));
}
```

Touchscreens widgets (SysEx)

The two round touchscreens are configured with SysEx messages sent on the Controller jack (jack index 1). SysEx payloads longer than 3 bytes are split across several USB-MIDI packets, using CIN 0x4 for the bytes that start/continue the message and CIN 0x5 / 0x6 / 0x7 for the packet that ends it with 1, 2 or 3 bytes respectively.

Command	SysEx (hex)	Purpose
touchscreen_widget	F0 7D 00 01 02 F7	Assign a widget (e.g. crossfader, color picker) to a touchscreen.
touchscreen_brightness	F0 7D 01 7F F7	Set the touchscreen backlight brightness.
touchscreen_mirror	F0 7D 7E 01 01 F7	Mirror the touchscreen content.
touchscreen_factory	F0 7D 7F 01 F7	Restore factory settings for the touchscreen.

Example – USB-MIDI packet breakdown for `touchscreen_brightness` (F0 7D 01 7F F7, 5 bytes, sent on Cable 1):

```
Packet 1: 14 F0 7D 01 (CIN 0x4: SysEx starts/continues, 3 bytes: F0 7D 01)
Packet 2: 16 7F F7 00 (CIN 0x6: SysEx ends with 2 bytes: 7F F7, byte 3 unused)
```

Example – USB-MIDI packet breakdown for `touchscreen_widget` (F0 7D 00 01 02 F7, 6 bytes, sent on Cable 1):

```
Packet 1: 14 F0 7D 00 (CIN 0x4: SysEx starts/continues, 3 bytes: F0 7D 00)
Packet 2: 17 01 02 F7 (CIN 0x7: SysEx ends with 3 bytes: 01 02 F7)
```

Icons Protocol

The Controller Deck exposes a **custom USB interface** used to update the content of the customizable buttons icons and fader labels:

- The **TouchDeck grid screen**: exposed as one big screen behind the 8x5 grid of buttons.



- The **Fader label screens**: (one per fader channel, 11 in total).



USB device identification

The bulk interface is exposed on the main `TouchDeck` USB device:

Field	Value
bcdUSB	2.00 High Speed
Vendor ID	0xFFFF
Product ID	0x1234
iManufacturer	LambdaConcept
iProduct	TouchDeck

Note

The Controller Deck's USB audio interface enumerates separately, as `PureAudio`, PID `0x1235` (see [Pure Audio Device](#)).

USB Descriptor details ...

Bus 001 Device 092: ID ffff:1234 LambdaConcept TouchDeck

Negotiated speed: High Speed (480Mbps)

Device Descriptor:

```

bLength          18
bDescriptorType   1
bcdUSB           2.00
bDeviceClass      0 [unknown]
bDeviceSubClass   0 [unknown]
bDeviceProtocol   0
bMaxPacketSize0   64
idVendor          0xffff LambdaConcept
idProduct         0x1234 TouchDeck
bcdDevice         1.05
iManufacturer     1 LambdaConcept
iProduct          2 TouchDeck
iSerial           3 00000000
bNumConfigurations 1

```

Configuration Descriptor:

```

bLength          9
bDescriptorType   2

```

(continues on next page)

(continued from previous page)

```

wTotalLength      0x006c
bNumInterfaces     2
bConfigurationValue 1
iConfiguration     0
bmAttributes       0x80
    (Bus Powered)
MaxPower           500mA
Interface Descriptor:
    [... MIDI Interface ...]
Interface Descriptor:
    bLength          9
    bDescriptorType   4
    bInterfaceNumber 1
    bAlternateSetting 0
    bNumEndpoints    2
    bInterfaceClass   255 Vendor Specific Class
    bInterfaceSubClass 255 Vendor Specific Subclass
    bInterfaceProtocol 255 Vendor Specific Protocol
    iInterface        0
Endpoint Descriptor:
    bLength          7
    bDescriptorType   5
    bEndpointAddress 0x82 EP 2 IN
    bmAttributes      3
        Transfer Type      Interrupt
        Synch Type         None
        Usage Type         Data
    wMaxPacketSize    0x0002 1x 2 bytes
    bInterval         7
Endpoint Descriptor:
    bLength          7
    bDescriptorType   5
    bEndpointAddress 0x03 EP 3 OUT
    bmAttributes      2
        Transfer Type      Bulk
        Synch Type         None
        Usage Type         Data
    wMaxPacketSize    0x0200 1x 512 bytes
    bInterval         255

```

USB interfaces

Interface 0: USB MIDI

Standard USB Audio Class MIDI streaming interface, documented in [MIDI Protocol](#).

Interface 1: Icons update

Custom interface specific to the TouchDeck controller:

bInterfaceNumber	bNumEndpoints	bInterfaceClass	bInterfaceSubClass	bInterfaceProtocol
1	2	Vendor Specific Class (255)	Vendor Specific Subclass (255)	Vendor Specific Protocol (255)

This interface exposes two endpoints:

Endpoint	Direction	Type	Max Packet Size	Purpose
2	IN	Interrupt	2 bytes	Reports the current position of the TouchDeck grid. Useful for companion software that needs to know the grid state but cannot access the MIDI interface because it is already claimed by another application (the same information is otherwise available from the MIDI interface, see MIDI Protocol).
3	OUT	Bulk	512 bytes	Receives screen icon data to update the TouchDeck grid screen or a fader label screen (see Bulk Endpoint 3 (Icons update payload) below).

Control Endpoint 0 (vendor requests)

The following vendor-specific control requests are available on the device's control endpoint (endpoint 0):

bRequest	wValue	wIndex	Description
1	Brightness (0-255)	0	Sets the TouchDeck grid screen brightness. (Defaults to 16).
2	0	0	Realign the TouchDeck grid screen buffer. Do this to make sure the next screen update starts with a clean slate.
3	Shift value (0-63)	0	Sets the vertical shift applied to the TouchDeck grid screen. Adjust this setting as needed to align the screen content correctly with the silicone buttons, depending on the user viewing angle. (Defaults to 32).
4	Brightness (0-255)	0	Sets the brightness of the TouchDeck LED matrix. This concerns only the LEDs surrounding the 8 x 5 silicone button grid. (Default to 5).
5	see MIDI Protocol	see MIDI Protocol	Sends a raw USB-MIDI packet to the device through the control endpoint instead of the MIDI streaming interface. This should be used only for development or debugging purposes.

Example of use (JS) – Setting the TouchDeck screen brightness to `duty`:

```
async setScreenBacklightPWM(device, duty) {
  await device.controlTransferOut({
    requestType: "vendor",
    recipient: "device",
    request: 1,
    value: duty,
    index: 0,
```

(continues on next page)

(continued from previous page)

```
    }, new Uint8Array(0));
}
```

Example of use (JS) – Realigning the TouchDeck screen buffer:

```
async resetScreenBuffer(device) {
    await device.controlTransferOut({
        requestType: "vendor",
        recipient: "device",
        request: 2,
        value: 0,
        index: 0,
    }, new Uint8Array(0));
}
```

Example of use (JS) – Setting the TouchDeck screen vertical shift:

```
async setScreenVerticalShift(device, shift) {
    await device.controlTransferOut({
        requestType: "vendor",
        recipient: "device",
        request: 3,
        value: shift,
        index: 0,
    }, new Uint8Array(0));
}
```

Example of use (JS) – Setting the TouchDeck LED matrix brightness:

```
async setLEDMatrixBrightness(device, brightness) {
    await device.controlTransferOut({
        requestType: "vendor",
        recipient: "device",
        request: 4,
        value: brightness,
        index: 0,
    }, new Uint8Array(0));
}
```

Interrupt Endpoint 2

This interrupts endpoint reports the current position of the TouchDeck grid pager, whenever it changes. Note that this is useful only in the case of a companion software that needs to know the grid state but cannot access the MIDI interface because it is already claimed by another application (the same information is otherwise available from the MIDI interface, see [MIDI Protocol](#)).

2 bytes in each interrupt packet:

Byte	Content
0	TouchDeck grid pager column
1	TouchDeck grid pager line

Bulk Endpoint 3 (Icons update payload)

Screen icon data is sent to the bulk endpoint as a single buffer:

Length (Bytes)	Content
1	Header byte: destination screen. 0 = TouchDeck grid screen, 1-11 = fader icons screens.
N	Raw image data (format depends on the destination screen, see below).

Screen number	Screen Resolution	Pixel Format	Full payload size
TouchDeck Grid screen (id 0)	1504 x 812	RGB 8:8:8, padded with 128 bytes at the end	3663873: 1 (header) + 1504 * 812 * 3 + 128 (padding)
Fader screens (id 1-11)	64 x 32	1-bit (black & white)	257: 1 (header) + 64 * 32 / 8

TouchDeck grid screen

The grid is organized into 40 buttons, arranged in an 8 columns x 5 rows layout. All buttons have the same square shape, except for the first button row which is smaller in height than the others.



Button type	Icon size (pixels)	Pixel Format
First row buttons	188 x 60	RGB 8:8:8
Square buttons	188 x 188	RGB 8:8:8

Concatenate the icon data for all buttons to build the full screen buffer.

The full screen buffer have a size of:

- 188 x 8 columns = 1504 pixels width.
- 60 x 1 row + 188 x 4 rows = 812 pixels height.

Example (JS) – Building the screen buffer:

```
xxx
```

The resulting `buffer` is then sent to the device with a bulk `transferOut` call on the endpoint described above.

Pure Audio Device



The Controller Deck integrates a stereo USB sound card, exposed to the host as a standard USB Audio Class device, for both playback and recording.

USB device identification

Field	Value
bcdUSB	2.00 High Speed
bDeviceClass	Miscellaneous Device (239)
Vendor ID	0xFFFF
Product ID	0x1235
iManufacturer	LambdaConcept
iProduct	PureAudio

Class	USB Audio Class (UAC)
Bitrate	Up to 768 kbit/s stereo, selectable by the host
Audio input	2x 3.5mm TS jack (L + R)
Audio output	2x 3.5mm TS jack (L + R)

USB Descriptor details ...

Bus 001 Device 091: ID ffff:1235 LambdaConcept PureAudio

Negotiated speed: High Speed (480Mbps)

Device Descriptor:

```

bLength          18
bDescriptorType   1
bcdUSB           2.00
bDeviceClass      239 Miscellaneous Device
bDeviceSubClass   2 [unknown]
bDeviceProtocol   1 Interface Association
bMaxPacketSize0   64
idVendor          0xffff LambdaConcept
idProduct         0x1235 PureAudio
bcdDevice         1.05
iManufacturer     1 LambdaConcept
iProduct          2 PureAudio
iSerial           3
bNumConfigurations 1

```

Configuration Descriptor:

```

bLength          9
bDescriptorType   2
wTotalLength      0x00da
bNumInterfaces    3
bConfigurationValue 1
iConfiguration    0
bmAttributes      0x80
  (Bus Powered)
MaxPower          500mA

```

Interface Association:

```

bLength          8
bDescriptorType  11

```

(continues on next page)

(continued from previous page)

```

bFirstInterface      0
bInterfaceCount      3
bFunctionClass       1 Audio
bFunctionSubClass    0 [unknown]
bFunctionProtocol    32
iFunction            0
Interface Descriptor:
  bLength            9
  bDescriptorType    4
  bInterfaceNumber   0
  bAlternateSetting   0
  bNumEndpoints      0
  bInterfaceClass    1 Audio
  bInterfaceSubClass  1 Control Device
  bInterfaceProtocol  32
  iInterface         0
AudioControl Interface Descriptor:
  bLength            9
  bDescriptorType    36
  bDescriptorSubtype  1 (HEADER)
  bcdADC             2.00
  bCategory          8
  wTotalLength       0x004b
  bmControls         0x00
AudioControl Interface Descriptor:
  bLength            8
  bDescriptorType    36
  bDescriptorSubtype 10 (CLOCK_SOURCE)
  bClockID           40
  bmAttributes       3 Internal programmable clock
  bmControls         0x03
    Clock Frequency Control (read/write)
  bAssocTerminal     0
  iClockSource       4 Any Frequency CMOS Clock Generator
AudioControl Interface Descriptor:
  bLength            17
  bDescriptorType    36
  bDescriptorSubtype  2 (INPUT_TERMINAL)
  bTerminalID        2
  wTerminalType      0x0101 USB Streaming
  bAssocTerminal     0
  bCSourceID         40
  bNrChannels        2
  bmChannelConfig    0x00000000
  iChannelNames      0
  bmControls         0x0000
  iTerminal          5 Stereo high-performance audio DAC
AudioControl Interface Descriptor:
  bLength            12
  bDescriptorType    36
  bDescriptorSubtype  3 (OUTPUT_TERMINAL)

```

(continues on next page)

(continued from previous page)

```

    bTerminalID          20
    wTerminalType        0x0301 Speaker
    bAssocTerminal       0
    bSourceID            2
    bCSourceID           40
    bmControls           0x0000
    iTerminal            0
AudioControl Interface Descriptor:
    bLength              17
    bDescriptorType       36
    bDescriptorSubtype    2 (INPUT_TERMINAL)
    bTerminalID          30
    wTerminalType        0x0201 Microphone
    bAssocTerminal       0
    bCSourceID           40
    bNrChannels           2
    bmChannelConfig       0x00000000
    iChannelNames        0
    bmControls           0x0000
    iTerminal            6 Stereo high-performance audio ADC
AudioControl Interface Descriptor:
    bLength              12
    bDescriptorType       36
    bDescriptorSubtype    3 (OUTPUT_TERMINAL)
    bTerminalID          3
    wTerminalType        0x0101 USB Streaming
    bAssocTerminal       0
    bSourceID            30
    bCSourceID           40
    bmControls           0x0000
    iTerminal            0
Interface Descriptor:
    bLength              9
    bDescriptorType       4
    bInterfaceNumber     1
    bAlternateSetting     0
    bNumEndpoints        0
    bInterfaceClass       1 Audio
    bInterfaceSubClass    2 Streaming
    bInterfaceProtocol    32
    iInterface           0
Interface Descriptor:
    bLength              9
    bDescriptorType       4
    bInterfaceNumber     1
    bAlternateSetting     1
    bNumEndpoints        2
    bInterfaceClass       1 Audio
    bInterfaceSubClass    2 Streaming
    bInterfaceProtocol    32
    iInterface           0

```

(continues on next page)

(continued from previous page)

AudioStreaming Interface Descriptor:

```

bLength          16
bDescriptorType   36
bDescriptorSubtype 1 (AS_GENERAL)
bTerminalLink     2
bmControls        0x00
bFormatType       1
bmFormats         0x00000001
    PCM
bNrChannels       2
bmChannelConfig   0x00000000
iChannelNames     0

```

AudioStreaming Interface Descriptor:

```

bLength          6
bDescriptorType   36
bDescriptorSubtype 2 (FORMAT_TYPE)
bFormatType       1 (FORMAT_TYPE_I)
bSubslotSize      4
bBitResolution    32

```

Endpoint Descriptor:

```

bLength          7
bDescriptorType   5
bEndpointAddress  0x01 EP 1 OUT
bmAttributes      5
    Transfer Type      Isochronous
    Synch Type         Asynchronous
    Usage Type         Data
wMaxPacketSize    0x0308 1x 776 bytes
bInterval         1

```

AudioStreaming Endpoint Descriptor:

```

bLength          8
bDescriptorType   37
bDescriptorSubtype 1 (EP_GENERAL)
bmAttributes      0x00
bmControls        0x00
bLockDelayUnits   0 Undefined
wLockDelay        0x0000

```

Endpoint Descriptor:

```

bLength          7
bDescriptorType   5
bEndpointAddress  0x81 EP 1 IN
bmAttributes      17
    Transfer Type      Isochronous
    Synch Type         None
    Usage Type         Feedback
wMaxPacketSize    0x0004 1x 4 bytes
bInterval         4

```

Interface Descriptor:

```

bLength          9
bDescriptorType   4
bInterfaceNumber  2

```

(continues on next page)

(continued from previous page)

```

bAlternateSetting      0
bNumEndpoints          0
bInterfaceClass        1 Audio
bInterfaceSubClass     2 Streaming
bInterfaceProtocol     32
iInterface             0
Interface Descriptor:
  bLength              9
  bDescriptorType      4
  bInterfaceNumber     2
  bAlternateSetting    1
  bNumEndpoints        1
  bInterfaceClass      1 Audio
  bInterfaceSubClass   2 Streaming
  bInterfaceProtocol   32
  iInterface           0
AudioStreaming Interface Descriptor:
  bLength              16
  bDescriptorType      36
  bDescriptorSubtype   1 (AS_GENERAL)
  bTerminalLink        3
  bmControls           0x00
  bFormatType          1
  bmFormats            0x00000001
  PCM
  bNrChannels          2
  bmChannelConfig      0x00000000
  iChannelNames        0
AudioStreaming Interface Descriptor:
  bLength              6
  bDescriptorType      36
  bDescriptorSubtype   2 (FORMAT_TYPE)
  bFormatType          1 (FORMAT_TYPE_I)
  bSubslotSize         4
  bBitResolution       32
Endpoint Descriptor:
  bLength              7
  bDescriptorType      5
  bEndpointAddress     0x82  EP 2 IN
  bmAttributes         5
    Transfer Type      Isochronous
    Synch Type         Asynchronous
    Usage Type         Data
  wMaxPacketSize       0x0308  1x 776 bytes
  bInterval            1
AudioStreaming Endpoint Descriptor:
  bLength              8
  bDescriptorType      37
  bDescriptorSubtype   1 (EP_GENERAL)
  bmAttributes         0x00
  bmControls           0x00

```

(continues on next page)

(continued from previous page)

bLockDelayUnits

0 Undefined

wLockDelay

0x0000

View Audio Devices

