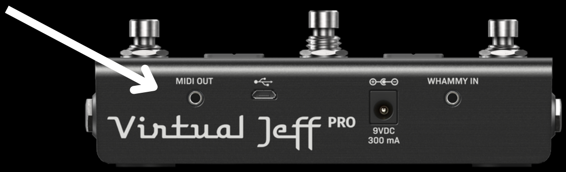
Virtual Jeff(R) PRO

DEEP DIVE

MIDI OUT: USER GUIDE

MIDI OUT: USER GUIDE

Virtual Jeff (R) PRO puts out MIDI data which always indicates the arm position



NOTE: MIDI OUT is completely independent from pitch control. It is not affected by any VJP features (e.g. V-Capo, HOLD, BLEND). MIDI data indicating the actual arm position is output at all times, even when the stompbox is in BYPASS mode !!

The MIDI OUT format is standard MIDI 14bit pitch bend data. This is a high-resolution MIDI format which provides much finer control by using two bytes for the pitch bend value (instead of one). Note than some MIDI devices can't handle 14bit format - but a lot do.

You'll need a 3.5mm to 5pin converter lead (if your MIDI device has the original 5pin DIN socket).

There are two types of converter leads: VJP uses 'Type A' (not Type B). See later for the Type A pinout - you can make your own converter!

FYI: There are two types because various manufacturers decided to wire them the opposite way around (thanks guys!). Type A is probably the most common so that's what we chose.

VJP MIDI details:

The MIDI data changes as the arm is moved up or down and is a static value when the arm is in the center. In hex, the values range from:

0x0000	(max pitch down) to...
0x2000	(center, no pitch change) to...
0x3FFF	(max pitch up).

In decimal, these values are: 0000 (max down), 8192 (center), 16383 (max up).

Note that some programs (like Band-in-a-Box) display this as -8192 thru to +8191

MIDI MESSAGE

The MIDI message is in this format: Status byte, Data byte 1, Data byte 2

- Status byte : 1110 CCCC (MIDI command 'E', Channel no from 0-15)
- Data byte 1 : 0LLL LLLL (7 bit pitch data LSB - fine val from 0-127)
- Data byte 2 : 0MMM MMMM (7 bit pitch data MSB - coarse val from 0-127)

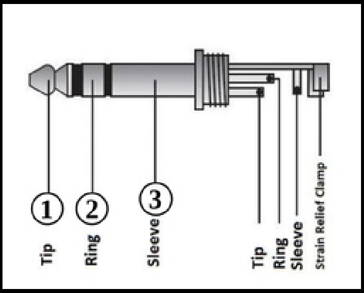
Note: VJP always sends on Channel 1, so the Status byte is always:
E0 hex (224 decimal)...i.e: pitch bend command (='E'), on Channel 1 (='0')

Example messages:

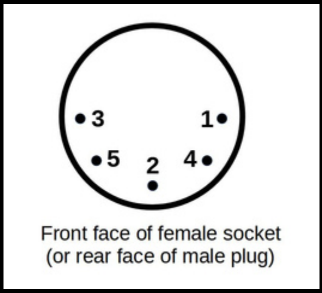
* Arm centered:	0xE0	0x40	0x00	(decimal 224 64 0)
* Arm fully up:	0xE0	0x7F	0x7F	(decimal 224 127 127)
* Arm half down:	0xE0	0x20	0x00	(decimal 224 32 00)

FYI: Some devices only accept 7bits for data by reading Databyte 2 and discarding Databyte1. This will work, but isn't as smooth in operation.

Type A PINOUT



3.5mm TRS plug



5pin 180degree DIN

TRS plug Pin	Type A converter wiring	DIN plug Pin
(1)	--	(5)
(2)	--	(4)
(3)	--	(2)