MIDI

This is not a full-fledged compendium about MIDI as such. If you are looking for details about MIDI in general you may start at https://en.wikipedia.org/wiki/MIDI or at https://www.midi.org/. Here, we'll concentrate on Titan's capabilities to be controlled by MIDI, or to control other devices by MIDI.

Trivia

Some basic trivia:

- MIDI was invented in the early 1980s, and is thus older than DMX
- originally it was just a way to control electronic musical intruments from other instruments/devices
- essentially MIDI is a signal connection. 'Real' MIDI is a 5pin cable, the connector being a round 'DIN' connector. But over the years, many developments have occured. The most important thing is USB-MIDI: transmitting MIDI information over USB interfaces. Then, there is a way to route MIDI over IP networks (rtp-MIDI), MIDI data can be stored in files (not relevant for Titan), and more.
- unlike DMX which strives to semi-permanently send the status of a whole system (i.e. a dmx universe), MIDI only transmits events: a key has been pressed or released, a parameter has changed etc. Hence, the data rate of MIDI is much smaller.
- MIDI messages usually comprise of up to 3 bytes. One bit of every byte is used to determine whether it's a status byte or a data byte. Hence, the 'resolution' of MIDI is 7 bit, or 0...127.
- the specification explains in great detail which message translates to which note or other value. This way you could be sure that when you pressed e.g. the note C on a MIDI keyboard this triggered the C on a sampler or expander.
- Nowadays there are also other uses for MIDI, e.g. MIDI Show Control (triggering cues in a show), MIDI Machine Control (sending transport commands to e.g. audio workstation programs), or MIDI Timecode

Message Types

MIDI devices can be set to 15 channels, and MIDI messages can be sent to a specific channel or all devices. A device set to 'Omni' mode ignores the channel and responds to all messages ignoring their channel.

The specification allows for various types of messages, and e.g. when using a MIDI keyboard there is a good chance that it defaults to an appropriate MIDI mapping without further adjustments. However, not every device responds to all message types.

The message types Titan can use are:

Message Type	Remarks
Note On/Note Off	can be used for triggers. The velocity parameter can be used to define a level. However, depending from the controller, Note Off may have a positive velocity, and Note On with velocity=0 should be treated Note Off.

Message Type	Remarks
Control Change	can be used for triggers. The value parameter can be used to define a level.
Program Change	can be used for triggers. Program change commands do not allow for an additional value parameter
MIDI Timecode	A system-common message, received by all devices in the system. Can be used as timecode source
MIDI Show Control	A system-exclusive message, used only by the device specified by the ID. Titan can receive this and react to it as stated in the manual.
MIDI Machine Control	System-Exclusive messages which can be sent by Titan, to control devices which can understand it.

While making Titan **receive** MIDI is somewhat straight-forward as there are pre-defined reactions provided, making Titan **send out** MIDI is currently done only via macros. This is very versatile as you can literally send all messages you want - but you have to know the message on a rather byte level. See MIDI Machine Control for an example.

MIDI and Titan

In general MIDI has been implemented in Titan consoles right from the beginning. However, there are some limitations. E.g. the Titan One and T1 do not provide for MIDI at all, the T2 can only do USB-MIDI (as there is no hardware MIDI outlet), and the Titan Mobile and Quartz can only receive MIDI but cannot send it (as there is no MIDI output provided). USB-MIDI was only added in Titan v12, output via USB-MIDI in Titan v13.

Here is an overview:

Console Type	MIDI In	MIDI Out	USB-MIDI In	USB-MIDI Out
Titan One	no	no	no	no
T1	no	no	no	no
T2	no	no	yes (v12 and up)	yes (v13 and up)
Titan Mobile	yes	no	yes (v12 and up)	yes (v13 and up)
Quartz	yes	no	yes (v12 and up)	yes (v13 and up)
Tiger Touch (non-pro) Pearl Expert (non-pro)	yes	yes	no	no
Tiger Touch Pro Pearl Expert Pro Tiger Touch II Arena Sapphire Touch	yes	yes	yes (v12 and up)	yes (v13 and up)

Connections and testing MIDI

- the standard MIDI connection is the 5pin DIN connector. Simply connect the sender's MIDI output with the receiver's MIDI input. There is a maximum cable length of 50 ft (~ 15m) as per the MIDI specification. If you need to run the signal over longer distances there are range extenders, see https://www.midiextender.com/
- in case of using USB-MIDI (from Titan v12) it should work out of the box. All connected MIDI controllers will show up in USB Expert (see below) and will work accordingly. output signals

(MIDI sent via macros) will be sent down at all avialable outputs
- if Titan and another MIDI software are running on the same computer (e.g. Titan PC suite and a DAW like reaper) you need to install loopMIDI which is something like a virtual MIDI loopback interface
- if you want to send/receive MIDI over network then you need to install rtpMIDI
- other software like midiOx or midiYoke may or may not help as well