**What are Midi Channels?**

Midi channels are communication pathways between Midi devices. There are 16 Midi channels and each can be assigned to a different Midi device. Channel assignments allow Midi devices to be arranged in combinations so that each device responds only to the information intended for it. Midi is a versatile way of recording notes, chords, and drum beats—to be played by Midi instruments—in the form of Midi tracks. Several Midi tracks can come together to form a Midi song.

**What information do Midi channels carry?**

Midi channels are *independent paths* over which communication can occur between Midi devices. By assigning channels to particular Midi devices or Midi-enabled sound generators within devices, those individual devices or generators can be independently controlled via Midi data.

**Midi messages**

Midi devices communicate through *Midi messages*, and these messages travel between devices on Midi channels.

Midi messages carry the specific instructions and data intended for the device that’s at the receiving end of a Midi channel. They typically contain two to three bytes of information, including channel-specific (with the channel number) and system-specific information.

Channel-specific information includes both *voice* and *mode* information. Midi voice information carries *performance data*, such as which notes to play, how long to hold them, and how hard to press the keys. Midi mode information carries instructions on how the receiving device should use the performance data that’s being received by it. Mode information includes instructions on whether to respond to data on *any* channel or only a *single* channel, and whether to play notes *polyphonically* (several notes at a time) or *monophonically* (only one note at a time).

Midi system-specific information includes general system, synchronization, and other system-related data.

**Midi commands**

Most traffic in a Midi data stream is Midi voice information. This information carries performance data, also referred to as *Midi commands*. A Midi command is a series of numbers which when received by device will cause it to do something. Midi commands convey *action* to receiving device, such as *play* note, *change*  sound (Midi program change), or *turn off* note. There are 8 groups of commands which are sent/received by Midi device.

Midi activity can take place on all 16 Midi channels simultaneously. Midi channels identified by individual Midi commands through Range of their Status numbers. For instance, look at range of numbers in Note On command. Range of possible Status number is 128 - 255. Here is a list of the 8 Midi command Status groups along with the range of numbers for the 16 Midi channels in each group:

|  |  |  |  |
| --- | --- | --- | --- |
| **Status (128 - 255)** | **Command** | **DATA 1 (0 - 127)** | **DATA 2 (0 - 127)** |
| Note OFF | 128-143 | Key # | Off Velocity |
| Note ON | 144-159 | Key # | On Velocity |
| Polyphonic Key Pressure | 160-175 | Key # | Pressure value |
| Control Change | 176-191 | Control # | Control Value |
| Program Change | 192-207 | Program # | -- Not Used -- |
| Monophonic Key (Channel) Pressure | 208-223 | Pressure Value (0-127) | -- Not Used -- |
| Pitch Bend | 224-239 | Range (LSB) | Range (MSB) |
| System exclusive | 240-255 | Manufacturer's ID | Model ID |

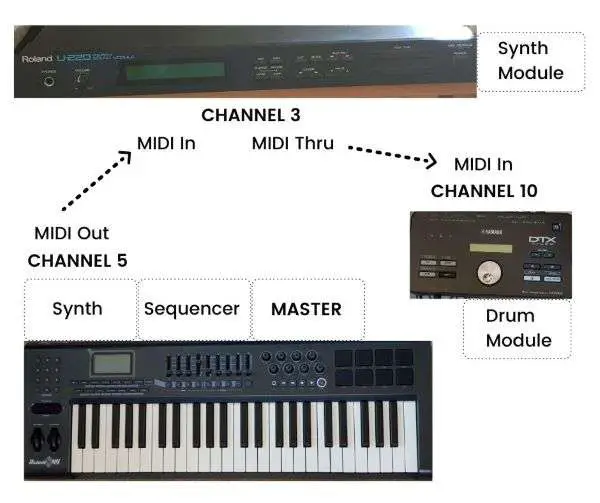
System exclusive are usually global commands affecting the entire device, and any quantity of individual numbers from as few as 4 to as many as 1000's), all other Midi commands have either 2 or 3 numbers. First number in any Midi command is Status, range 128 - 255. Numbers following Status are Data, range 0 - 127. Some Midi commands have only 1 Data number and some have 2. For example:

* **144** 60 127 - turn ON note #60 on Midi channel 1 with a velocity of 127
* **144** 60 0 - turn OFF note #60 on Midi channel 1
* **192** 15 - change the program (sound) on Midi channel 1 to program #15
* **193** 21 - change the program (sound) on Midi channel 2 to program #21

**Midi channel assignments**

A Midi device can control the way it interacts with other Midi devices using *channel assignments*. With Midi channel assignments you can, for instance, use one device to send Midi performance data to a variety of other devices, and specify which device receives what type of data. Let’s look at an example to see how Midi channel assignments work.

**How to use Midi channels**



**Example of Midi channel assignments**

The above setup shows a *master* Midi synth/sequencer connected to a synth module, and the synth module is connected to a drum module. Midi ports on each of the devices are connected as follows:

* Midi (master) synth/sequencer to synth module—Midi *out* to Midi *in*
* Synth module to drum module—Midi *thru* to Midi *in*

This setup describes a system where one Midi device (the master) *controls*, ie. sends Midi information, to two *receiving* Midi devices (the synth module and the drum module).

But in this example, the master device is actually made up of two components—a *sequencer* and a *synth*.

Having an in-built sequencer is not unusual for many Midi devices, though many people prefer to use sequencers available in computer software (ie. *digital audio workstations* or *DAW*s) as they offer more versatility and flexibility.

The synth module is connected directly to the master device. The drum module, however, is connected only to the synth module and not directly to the master device.

Master device can control drum module by using *Midi thru* port on synth module. This port passes all information received at Midi in port. So Midi information sent by master devices passes through synth module and reaches drum module.

By assigning different Midi *channel numbers* to the synth module and drum module, although they are both receiving the same information (due to the Midi thru connection), they will only act on the information intended for them.

**Midi channel numbers**

For each of the receiving devices in the example set up, you can assign Midi channel numbers according to how you want each device to operate. Assume the following channel assignments apply:

* Channel 5—Master synth
* Channel 3—Synth module
* Channel 10—Sampler module

Say you record bass track on Midi sequencer using Channel 5, melody track on sequencer using Channel 3, and drum track on Channel 10. Incidentally, many synths are pre-assigned to output drums and percussion on Midi Channel 10.

If you record on (master) sequencer using above channels, then when you play back sequencer each module will only respond with the instruments that match the channel assignments. So, for instance, the synth component of the master device will respond with a bass track (assigned to Channel 5), the synth module will respond with a melody track (assigned to Channel 3), and the drum module will respond with a drum track (assigned to Chanel 10).

This arrangement will only work properly if you select the correct Midi mode—you would need to ensure that an *Omni Off* mode is selected, otherwise, the channel assignments would be ignored. ([*see how Midi modes operate*](https://audiointerfacing.com/what-are-midi-modes/#the-midi-modes))

This example shows how Midi channel assignments can be used to match channel numbers between devices that are intended to pair.

**Midi channels vs Midi tracks**

Midi channels are the *pathways of communication* between Midi devices. Midi channels are used to convey Midi messages between devices based on their channel assignments. Midi tracks, on the other hand, refer to *Midi sequencing data* that contains information about the notes, chords, melodies, and drum beats (in Midi format) that are to be played by designated Midi devices.

In our earlier example, the sequencer module in the master device stored Midi tracks. The bass, melody, and drum sequences that you recorded each represented a Midi track.

A Midi song would essentially be stored as a series of Midi tracks in a Midi sequencer.

Midi tracks work in a similar manner to audio tracks, but the information contained in a Midi track is simply a set of Midi instructions about which notes, chords, and drum beats to play (that can trigger designated Midi devices)—it is not actual audio data. ([*see difference between Midi and digital audio data*](https://audiointerfacing.com/midi-and-digital-audio-difference/))

**Conclusion**

Midi channels are *pathways for communication* between Midi devices. Midi channels convey information in the form of *Midi messages*. Midi messages contain Midi performance data—in form of *Midi commands*—and Midi mode data, both of which help Midi devices to know what to play and how to respond to the Midi information that they receive.

By using channel assignments, you can control how Midi devices respond to the Midi information that they receive based on the channels they’re assigned to. Midi channel assignments, therefore, allow arrangements of several Midi instruments to work together, each producing their own sound.

Midi tracks contain sequences of Midi notes, chords, and drum beats that come together to form a Midi song. Midi data of this type represents performance information that triggers *which notes to play and when to play them* on a Midi device—it does not represent actual audio information.