

# MIDI

Musical Instrument Digital Interface (MIDI) (<http://en.wikipedia.org/wiki/MIDI>) is a standard protocol for interconnecting computers with musical instruments, stage lighting, and other time-oriented media.

Strictly speaking, MIDI is unrelated to audio. But because MIDI is commonly used with music, this article is placed in the audio section.

## MIDI for NDK

---

Android 10 makes it easier to port professional audio apps using MIDI to the Android platform.

AMidi (<https://developer.android.com/ndk/reference/group/midi>) is an NDK API (<https://developer.android.com/ndk>) that gives app developers the ability to send and receive MIDI data with C/C++ code.

Android MIDI apps usually use the midi (<https://developer.android.com/reference/android/media/midi/package-summary>) API to communicate with the Android MIDI service. MIDI apps primarily depend on the MidiManager (<https://developer.android.com/reference/android/media/midi/MidiManager>) class to discover, open, and close one or more MidiDevice (<https://developer.android.com/reference/android/media/midi/MidiDevice>) objects, and pass data to and from each device through the device's MIDI input and output ports.

## Transports

---

The physical transport layer ([http://en.wikipedia.org/wiki/Transport\\_layer](http://en.wikipedia.org/wiki/Transport_layer)) specified in original MIDI 1.0 is a current loop with 5-pin DIN ([http://en.wikipedia.org/wiki/DIN\\_connector](http://en.wikipedia.org/wiki/DIN_connector)) connector.

Since MIDI 1.0, additional transports have been defined, including MIDI over USB and MIDI over Bluetooth Low Energy (BLE).

([http://en.wikipedia.org/wiki/Bluetooth\\_low\\_energy](http://en.wikipedia.org/wiki/Bluetooth_low_energy)).

## MIDI for Android

---

Android supports USB On-The-Go ([http://en.wikipedia.org/wiki/USB\\_On-The-Go](http://en.wikipedia.org/wiki/USB_On-The-Go)), which permits an Android device to act as a USB host to drive USB peripherals. The USB host mode APIs permit developers to implement MIDI over USB at the application level, but until recently there have been no built-in platform APIs for MIDI.

Beginning with the Android 6.0 (Marshmallow) release, device makers can enable optional MIDI support in the platform. Android directly supports USB, BLE, and virtual (interapp) transports. Android supports MIDI 1.0 through an external adapter.

For details on application programming with the MIDI APIs, see the

[android.media.midi](https://developer.android.com/reference/android/media/midi)

(<https://developer.android.com/reference/android/media/midi/package-summary.html>) package.

The remainder of this article discusses how an Android device maker can enable MIDI support in the platform.

## Enabling transports

---

MIDI implementation depends on ALSA for USB host mode and USB peripheral mode transports. ALSA isn't used for the BLE and virtual transports.

### USB host mode

To enable MIDI for USB host mode, first support USB host mode in general, and then enable `CONFIG_SND_RAWMIDI` and `CONFIG_SND_USB_MIDI` in your kernel configuration. See [Android Kernel Configuration](#). (/devices/architecture/kernel/config)

The MIDI over USB transport is formally defined by the [Universal Serial Bus Device Class Definition for MIDI Devices Release 1.0 Nov 1, 1999](#)

(<https://www.usb.org/sites/default/files/midi10.pdf>) standard published by the [USB Implementers Forum, Inc](http://www.usb.org/) (<http://www.usb.org/>).

## USB peripheral mode

To enable MIDI for USB peripheral mode, you may need to apply patches to your Linux kernel to integrate `drivers/usb/gadget/f_midi.c` into the USB gadget driver. These patches are available for Linux kernel version 3.10. These patches have not yet been updated for [ConfigFs](http://en.wikipedia.org/wiki/ConfigFs) (<http://en.wikipedia.org/wiki/ConfigFs>) (a new architecture for USB gadget drivers), nor are they merged at upstream [kernel.org](http://kernel.org) (<http://kernel.org>).

The patches are shown in commit order for the kernel tree at project `kernel/common` branch `android-3.10`:

1. <https://android-review.googlesource.com/#/c/127450/>  
(<https://android-review.googlesource.com/#/c/127450/>)
2. <https://android-review.googlesource.com/#/c/127452/>  
(<https://android-review.googlesource.com/#/c/127452/>)
3. <https://android-review.googlesource.com/#/c/143714/>  
(<https://android-review.googlesource.com/#/c/143714/>)

You also need to do either of the following:

- Go to **Settings > Developer options > Networking** and check the box for MIDI in the Select USB Configuration dialog.
- While attached to the USB host, pull down from the top of screen, select the entry **USB for**, and then select **MIDI**.

## BLE

MIDI over BLE is always enabled, provided the device supports BLE.

## Virtual (interapp)

The virtual (interapp) transport is always enabled.

## Claiming the feature

---

Apps can screen for the presence of MIDI support using the `android.software.midi` feature.

To claim MIDI support, add this line to your `device.mk`:

```
PRODUCT_COPY_FILES += \
frameworks/native/data/etc/android.software.midi.xml:system/etc/permissions/midi.xml
```

See the [Android Compatibility Definition Document \(CDD\)](#).

(</compatibility/android-cdd.pdf>) for information on requirements to claim the feature.

## Debugging while in host mode

---

While in USB host mode, Android Debug Bridge (adb) debugging over USB is unavailable. See the [Wireless usage](#)

(<http://developer.android.com/tools/help/adb.html#wireless>) section of Android Debug Bridge for an alternative.

---

Content and code samples on this page are subject to the licenses described in the [Content License \(/license\)](#). Java is a registered trademark of Oracle and/or its affiliates.