

TUTORIALS (/EN/TUTORIAL/HOMEPAGE) > Built-In Examples (/en/Tutorial/BuiltInExamples) > 04.Communication > Midi

---

# MIDI Note Player

This tutorial shows how to send MIDI notes from an Arduino or Genuino board to a MIDI instrument connected through the standard 5 poles DIN cable.

MIDI (<http://en.wikipedia.org/wiki/MIDI>), the Musical Instrument Digital Interface, is a useful protocol for controlling synthesizers, sequencers, and other musical devices. MIDI devices are generally grouped in to two broad classes: controllers (i.e. devices that generate MIDI signals based on human actions) and synthesizers (including samplers, sequencers, and so forth). The latter take MIDI data in and make sound, light, or some other effect.

MIDI is a serial protocol that operates at 31,250 bits per second. The board built-in serial port (all of them on the Mega as well) can send data at that rate.

MIDI bytes are divided into two types: **command bytes** and **data bytes**. Command bytes are always 128 or greater, or 0x80 to 0xFF in hexadecimal. Data bytes are always less than 127, or 0x00 to 0x7F in hex. Commands include things such as note on, note off, pitch bend, and so forth. Data bytes include things like the pitch of the note to play, the velocity, or loudness of the note, amount of pitch bend and so forth. For more details, see the MIDI specification or one of the many MIDI Protocol Guides (<http://hinton-instruments.co.uk/reference/midi/protocol/index.htm>) on the Web.

MIDI data is usually notated in hexadecimal because MIDI banks and instruments are grouped in groups of 16.

For more see this introduction to MIDI (<http://www.tigoe.net/pcomp/code/communication/midi>) or this example (<http://itp.nyu.edu/physcomp/Labs/MIDIOutput>).

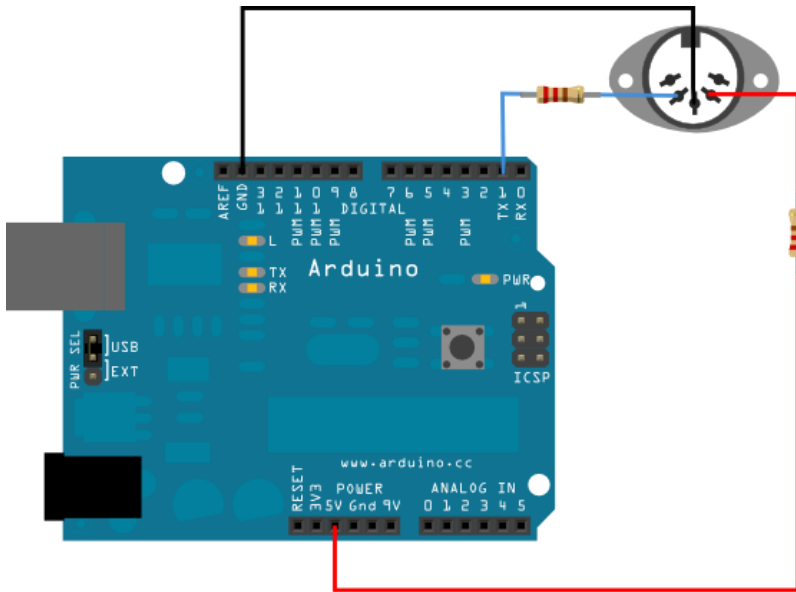
## Hardware Required

- Arduino or Genuino Uno
- Female MIDI jack
- 2 220 ohm resistors
- hook-up wires
- MIDI enabled device (*optional, for testing*)

All MIDI connectors are female, by definition of the MIDI spec. Here's how to wire the connector to the board:

- MIDI jack pin 5 connected to Digital pin 1 through a 220 ohm resistor
- MIDI jack pin 2 connected to ground
- MIDI jack pin 4 connected to +5V through a 220 ohm resistor

click the image to enlarge

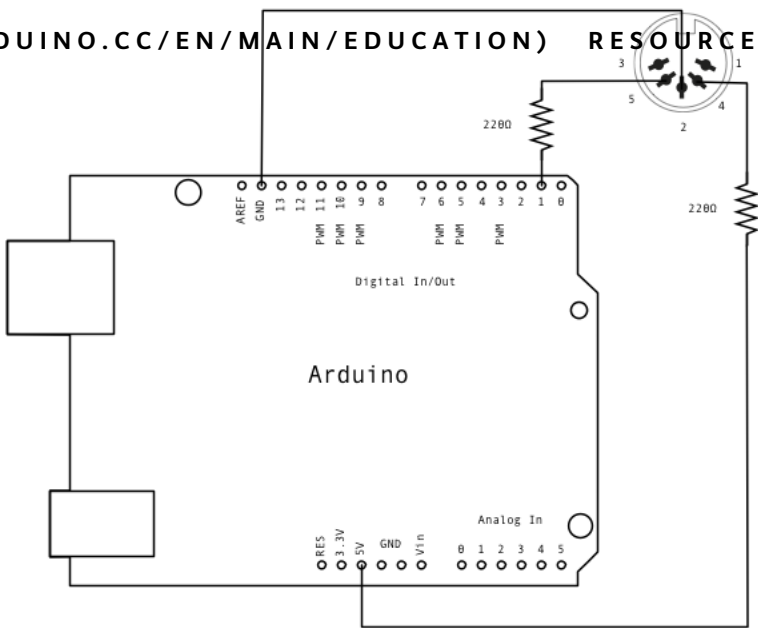


(//www.arduino.cc/en/uploads/Tutorial/MIDI\_bb.png)

image developed using Fritzing (<http://www.fritzing.org>). For more circuit examples, see the Fritzing project page (<http://fritzing.org/projects/>)

## Schematic

click the image to enlarge



(//www.arduino.cc/en/uploads/Tutorial/MIDI\_schem.png)

## Code

**Attention** If you're using a board with ATmega32U4 like DUE or Leonardo, please replace Serial with Serial1 in the sketch below.

```

/*
  MIDI note player

  This sketch shows how to use the serial transmit pin (pin 1) to send MIDI note data.
  If this circuit is connected to a MIDI synth, it will play the notes
  F#-0 (0x1E) to F#-5 (0x5A) in sequence.

  The circuit:
  - digital in 1 connected to MIDI jack pin 5
  - MIDI jack pin 2 connected to ground
  - MIDI jack pin 4 connected to +5V through 220 ohm resistor
  - Attach a MIDI cable to the jack, then to a MIDI synth, and play music.

  created 13 Jun 2006
  modified 13 Aug 2012
  by Tom Igoe

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Midi
  */

void setup() {
  // Set MIDI baud rate:
  Serial.begin(31250);
}

void loop() {

```

ARDUINO (COURTESY OF ARDUINO) 5 ARDUINO RESOURCES { COMMUNITY HELP SIGN IN ()

```

// play notes from F#-0 (0x1E) to F#-5 (0x5A):
noteOn(0x90, note, 0x45);
delay(100);
noteOn(0x90, note, 0x00);
delay(100);
}
}

```

*// plays a MIDI note. Doesn't check to see that cmd is greater than 127, or that data values are less than 127:*

```

void noteOn(int cmd, int pitch, int velocity) {
  Serial.write(cmd);
  Serial.write(pitch);
  Serial.write(velocity);
}

```

[Get Code] (<http://www.arduino.cc/en/Tutorial/Midi?action=sourceblock&num=1>)

## See Also

- [serial.begin](http://www.arduino.cc/en/Serial/Begin) (<http://www.arduino.cc/en/Serial/Begin>())
- [for](http://www.arduino.cc/en/Reference/For) (<http://www.arduino.cc/en/Reference/For>()) loop
- [ASCIITable](http://www.arduino.cc/en/Tutorial/ASCIITable) (<http://www.arduino.cc/en/Tutorial/ASCIITable>) - Demonstrates Arduino's advanced serial output functions.
- [Dimmer](http://www.arduino.cc/en/Tutorial/Dimmer) (<http://www.arduino.cc/en/Tutorial/Dimmer>) - Move the mouse to change the brightness of an LED.
- [Graph](http://www.arduino.cc/en/Tutorial/Graph) (<http://www.arduino.cc/en/Tutorial/Graph>) - Send data to the computer and graph it in Processing.
- [MultiSerialMega](http://www.arduino.cc/en/Tutorial/MultiSerialMega) (<http://www.arduino.cc/en/Tutorial/MultiSerialMega>) - Use two of the serial ports available on the Arduino and Genuino Mega.
- [PhysicalPixel](http://www.arduino.cc/en/Tutorial/PhysicalPixel) (<http://www.arduino.cc/en/Tutorial/PhysicalPixel>) - Turn a LED on and off by sending data to your board from Processing or Max/MSP.
- [ReadASCIIString](http://www.arduino.cc/en/Tutorial/ReadASCIIString) (<http://www.arduino.cc/en/Tutorial/ReadASCIIString>) - Parse a comma-separated string of integers to fade an LED.
- [SerialCallResponse](http://www.arduino.cc/en/Tutorial/SerialCallResponse) (<http://www.arduino.cc/en/Tutorial/SerialCallResponse>) - Send multiple variables using a call-and-response (handshaking) method.
- [SerialCallResponseASCII](http://www.arduino.cc/en/Tutorial/SerialCallResponseASCII) (<http://www.arduino.cc/en/Tutorial/SerialCallResponseASCII>) - Send multiple variables using a call-and-response (handshaking) method, and ASCII-encode the values before sending.
- [SerialEvent](http://www.arduino.cc/en/Tutorial/SerialEvent) (<http://www.arduino.cc/en/Tutorial/SerialEvent>) - Demonstrates the use of `SerialEvent()`.
- [VirtualColorMixer](http://www.arduino.cc/en/Tutorial/VirtualColorMixer) (<http://www.arduino.cc/en/Tutorial/VirtualColorMixer>) - Send multiple variables from Arduino to your computer and read them in Processing or Max/MSP.

## NEWSLETTER

ENTER YOUR EMAIL TO SIGN UP

SUBSCRIBE

[Terms Of Service \(//www.arduino.cc/en/Main/TermsOfService\)](http://www.arduino.cc/en/Main/TermsOfService)

[Privacy Policy \(//www.arduino.cc/en/Main/PrivacyPolicy\)](http://www.arduino.cc/en/Main/PrivacyPolicy)

[Contact Us \(//www.arduino.cc/en/Main/ContactUs\)](http://www.arduino.cc/en/Main/ContactUs)

[About Us \(//www.arduino.cc/en/Main/AboutUs\)](http://www.arduino.cc/en/Main/AboutUs)

[Distributors \(//store.arduino.cc/distributors\)](http://store.arduino.cc/distributors)

[Careers \(//careers.arduino.cc\)](http://careers.arduino.cc)

[Security \(//www.arduino.cc/en/Main/Security\)](http://www.arduino.cc/en/Main/Security)

© 2019 Arduino ([//www.arduino.cc/en/Main/CopyrightNotice](http://www.arduino.cc/en/Main/CopyrightNotice))

(<https://www.facebook.com/arduino>) (<https://www.instagram.com/arduino>) (<https://www.linkedin.com/company/arduino>) (<https://www.pinterest.com/arduino>) (<https://www.youtube.com/channel/UC8butISFwT-97uouP7ZjFdBg>) ([arduino@arduino.cc](mailto:arduino@arduino.cc)) (<http://www.arduino.cc>)