

Arduino to MIDI guide

Keep in mind that the Arduino Uno can not connect as MIDI, you need the Pro Micro / Leonardo model.

Setup

1. Install MIDIUSB library <https://docs.arduino.cc/libraries/midiusb/>
2. Add the MIDIUSB.h Library: Go to Tools < Manage Libraries...
3. Search for "MIDIUSB" by Gary Grewal
4. Install the latest version
5. Select your board: Tools > Board > Arduino Leonardo (or Pro Micro)



Potentiometers / Sliders

What you need:

- 1 potentiometer or slider
- 3 wires

How to connect:

```
Potentiometer:  
├ Left pin / GND → GND  
├ Middle pin / Signal → A0 (or A1, A2, A3...)  
└ Right pin / VCC / Power → 5V
```

Code

```

#include <MIDIUSB.h>

int lastValue = 0;

void setup() {
  pinMode(A0, INPUT); // ← Change A0 to your pin
}

void loop() {
  int value = analogRead(A0) / 8; // ← Change A0 to match setup

  if (value != lastValue) {
    midiEventPacket_t event = {0x0B, 0xB0, 1, value};
    //                               ↑ ← CC number
    MidiUSB.sendMIDI(event);
    MidiUSB.flush();
    lastValue = value;
  }

  delay(10);
}

```

To customize:

- Change A0 to A1, A2, A3 etc. for different pins
- Change the CC number 1 a different one for every input

Note button

What you need

- 1 push button
- 2 wires

How to connect

Button:

```

├ One side → Pin 2 (or 3, 4, 5...)
└ Other side → GND

```

Code

```

#include <MIDIUSB.h>

int lastState = HIGH;

void setup() {
  pinMode(2, INPUT_PULLUP); // ← Change 2 to your pin
}

void loop() {
  int state = digitalRead(2); // ← Change 2 to match setup

  if (state != lastState) {
    if (state == LOW) {
      // Button pressed
      midiEventPacket_t noteOn = {0x09, 0x90, 60, 127};
      //                                     ↑   ↑
      //                                     note velocity
      MidiUSB.sendMIDI(noteOn);
    } else {
      // Button released
      midiEventPacket_t noteOff = {0x08, 0x80, 60, 0};
      //                                     ↑
      //                                     note
      MidiUSB.sendMIDI(noteOff);
    }
    MidiUSB.flush();
    lastState = state;
  }

  delay(10);
}

```

To customize:

- Change pin 2 to 3, 4, 5 etc.
- Change note number 60 :
 - 60 = C (middle C)
 - 62 = D
 - 64 = E
 - 65 = F
 - 67 = G
 - 69 = A
 - 71 = B

- 72 = C (one octave higher)
- Change velocity 100 (0-127, higher = louder)
-

Pull sensor / distance sensor

What you need:

- Any sensor with analog output
- Wires as needed for your sensor

How to connect:

Sensor:

- └ VCC/Power → 5V
- └ GND → GND
- └ Signal → A0 (or A1, A2, A3...)

Code to copy:

```

#include <MIDIUSB.h>

int lastValue = 0;

void setup() {
  pinMode(A0, INPUT); // ← Change A0 to your pin
}

void loop() {
  int sensorValue = analogRead(A0); // ← Change A0 to match setup
  int midiValue = map(sensorValue, 0, 1023, 0, 127);
  //                               ↑   ↑
  //                               adjust these if needed

  if (abs(midiValue - lastValue) > 1) { // Ignore tiny changes
    midiEventPacket_t event = {0x0B, 0xB0, 74, midiValue};
    //                               ↑ ← CC number
    MidiUSB.sendMIDI(event);
    MidiUSB.flush();
    lastValue = midiValue;
  }

  delay(10);
}

```

To customize:

- Change `A0` to your sensor pin
- Change `0, 1023` in the `map()` function if your sensor gives different values
- Change CC number `74` to control different parameters



2 potmeters 2 buttons code

```

#include <MIDIUSB.h>

int lastValueA0 = 0;
int lastValueA1 = 0;
int lastState2 = HIGH;
int lastState3 = HIGH;

void setup() {
  pinMode(A0, INPUT);          // Pot 1
  pinMode(A1, INPUT);          // Pot 2
  pinMode(2, INPUT_PULLUP);    // Button 1
  pinMode(3, INPUT_PULLUP);    // Button 2
}

void loop() {
  // =====
  // POTENTIOMETER 1 (A0) → CC 1
  // =====
  int valueA0 = analogRead(A0) / 8;
  if (valueA0 != lastValueA0) {
    midiEventPacket_t event = {0x0B, 0xB0, 60, valueA0};
    MidiUSB.sendMIDI(event);
    MidiUSB.flush();
    lastValueA0 = valueA0;
  }

  // =====
  // POTENTIOMETER 2 (A1) → CC 2
  // =====
  int valueA1 = analogRead(A1) / 8;
  if (valueA1 != lastValueA1) {
    midiEventPacket_t event = {0x0B, 0xB0, 61, valueA1};
    MidiUSB.sendMIDI(event);
    MidiUSB.flush();
    lastValueA1 = valueA1;
  }

  // =====
  // BUTTON 1 (Pin 2) → Note 60 (Middle C)
  // =====
  int state2 = digitalRead(2);
  if (state2 != lastState2) {
    if (state2 == LOW) {
      midiEventPacket_t noteOn = {0x09, 0x90, 60, 127};
      MidiUSB.sendMIDI(noteOn);
    } else {
      midiEventPacket_t noteOff = {0x08, 0x80, 60, 0};
    }
  }
}

```

```

    MidiUSB.sendMIDI(noteOff);
}
MidiUSB.flush();
lastState2 = state2;
}

// =====
// BUTTON 2 (Pin 3) → Note 61 (C#)
// =====
int state3 = digitalRead(3);
if (state3 != lastState3) {
    if (state3 == LOW) {
        midiEventPacket_t noteOn = {0x09, 0x90, 61, 127};
        MidiUSB.sendMIDI(noteOn);
    } else {
        midiEventPacket_t noteOff = {0x08, 0x80, 61, 0};
        MidiUSB.sendMIDI(noteOff);
    }
    MidiUSB.flush();
    lastState3 = state3;
}

delay(10);
}

```



Common Pin Reference

Analog Pins (for sensors & potentiometers):

A0 , A1 , A2 , A3 , A4 , A5

Digital Pins (for buttons):

2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10 , 11 , 12 , 13

Don't use:

- Pin 0 and 1 (used for USB communication)

Sending notes

C4 = 60 (Middle C)
C#4 = 61
D4 = 62
D#4 = 63
E4 = 64
F4 = 65
F#4 = 66
G4 = 67
G#4 = 68
A4 = 69
A#4 = 70
B4 = 71
C5 = 72

Each octave adds or subtracts 12:

- C3 = 48 (one octave lower)
- C5 = 72 (one octave higher)

CC

You can use any CC from 0-119, but these are the most common!