

NiftyDrum

Official Documentation

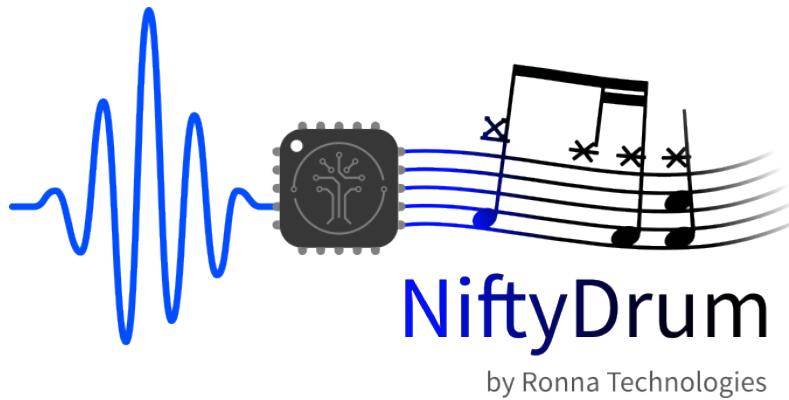
Ronna Technologies

Copyright © 2025 Ronna Technologies

Table of contents

1. About NiftyDrum	3
1.1 Description	3
1.2 How It Works	3
1.3 Specifications	4
2. The Board	5
2.1 Description	5
2.2 Connecting Sensors	6
2.3 MIDI Outputs	6
2.4 USB-C Port	7
2.5 Raspberry-Pi hat form factor	7
3. The App	8
3.1 Configure NiftyDrum	8
3.2 How to install the app	8
4. Serial Protocol	10
5. Arduino	11

1. About NiftyDrum



1.1 Description

NiftyDrum is a trigger-to-MIDI conversion module that transforms piezo and FSR sensor inputs into MIDI messages. Connect up to 9 piezo sensors and 1 FSR (Force Sensing Resistor) to the dedicated terminal blocks, then receive MIDI data via USB-C connection.

1.2 How It Works

NiftyDrum delivers high-level MIDI performance in 4 easy steps:

- Connect sensors: Attach up to 9 piezo sensors and 1 FSR to the terminal blocks
- Plug in: Connect to your DAW, Raspberry Pi, or drum module via USB
- Configure: Use the web-based GUI to adjust trigger parameters, MIDI mapping, and velocity curves
- Play: Notes are transmitted instantly with imperceptible latency

1.3 Specifications

1.3.1 Hardware

- Piezo inputs: 9 channels
- FSR input: 1 channel (hi-hat controller)
- Connector type: Terminal blocks
- USB interface: Type-C
- Dimensions: 65 × 56.5 mm

1.3.2 Performance

- Latency: <2.5 ms
- Sample rate: >10 kHz
- Velocity resolution: 127 levels (full MIDI range)

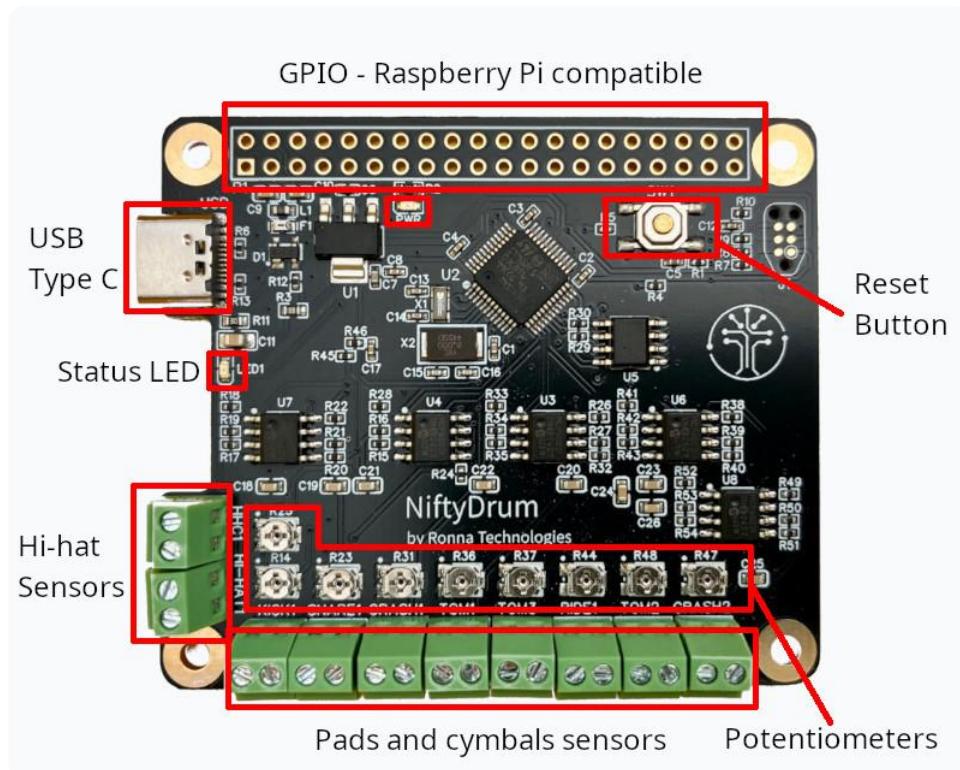
1.3.3 Software

- Platform support: Windows, macOS, Linux
- User interface: Web-based application
- Firmware updates: Via USB
- MIDI output: Note messages and Control Changes (CC)

2. The Board

2.1 Description

The NiftyDrum board is shown in the following image.



This board features the following interfaces:

- Terminal blocks for sensor inputs
- USB Type-C port for laptop or PC connectivity
- 9 potentiometers for sensitivity adjustment
- 2 LEDs
- Reset button
- Raspberry Pi-compatible GPIO header
- 4 mounting holes

2.2 Connecting Sensors

The board provides 9 piezo inputs, supporting up to 9 single-zone pads, as well as 1 FSR input.

2.2.1 Hi-Hat Sensors

On the left side of the board, two dedicated terminal blocks are reserved for hi-hat sensors:

- Top terminal block: Connects to an FSR (Force Sensing Resistor) sensor for hi-hat controller input
- Bottom terminal block: Connects to a piezo sensor for hi-hat cymbal trigger

2.2.2 Standard Pads and Cymbals

The remaining eight terminal blocks, located at the bottom of the board, accommodate regular pads and cymbals.

2.2.3 Important Notes

- For all terminal blocks, the ground pin is positioned on the left-hand side
- Nine onboard potentiometers enable hardware-level sensitivity adjustments for maximum flexibility
- If unsure about sensitivity settings, leave potentiometers at their midpoint for balanced performance

2.3 MIDI Outputs

The board offers two methods for transmitting MIDI notes and Control Changes:

- USB-C port: Outputs USB MIDI messages
- GPIO UART pins: Raspberry Pi GPIO-compatible interface

2.4 USB-C Port

Using NiftyDrum as a USB device is the recommended method for receiving MIDI messages. This configuration enables:

- Integration with DAW software for high-quality sound output from your laptop
- Control and configuration via the [official app](#)

2.5 Raspberry-Pi hat form factor

The board is designed with a Raspberry Pi 4 HAT form factor, ensuring seamless integration.

2.5.1 App Features

The official app provides comprehensive control over your NiftyDrum board:

- Customize MIDI note assignments for each trigger
- Design custom velocity curves per trigger
- Adjust advanced parameters including gain, threshold, scan time, mask time, and decay
- Update board firmware to the latest version

3. The App

3.1 Configure NiftyDrum

NiftyDrum is highly configurable.

3.2 How to install the app

NiftyDrum.com

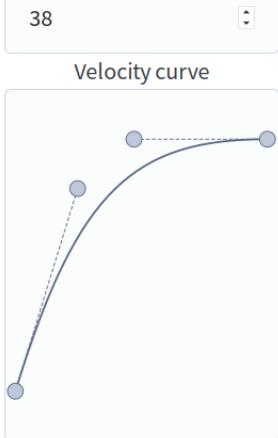
NiftyDrum
by Ronna Technologies

(i) Snare 

Note: 38

Velocity curve:



Gain: 5
Threshold: 100

Scan [ms]: 2.5
Mask [ms]: 10.0
Decay [ms]: 70.0

NiftyDrum
by Ronna Technologies

(i) Hi-hat Controller 

Interval (μ s): 25000
Noise threshold: 100
Offset: 127
Trig: 2000

9000
5000
30
10
105
1
400
200

4. Serial Protocol

5. Arduino
