

MIDI (Тройка-модуль)

[Тройка-MIDI](#) (Musical Instrument Digital Interface — цифровой интерфейс музыкальных инструментов) — это модули, которые позволяют подключиться к профессиональным музыкальным инструментам.

Подключение и настройка



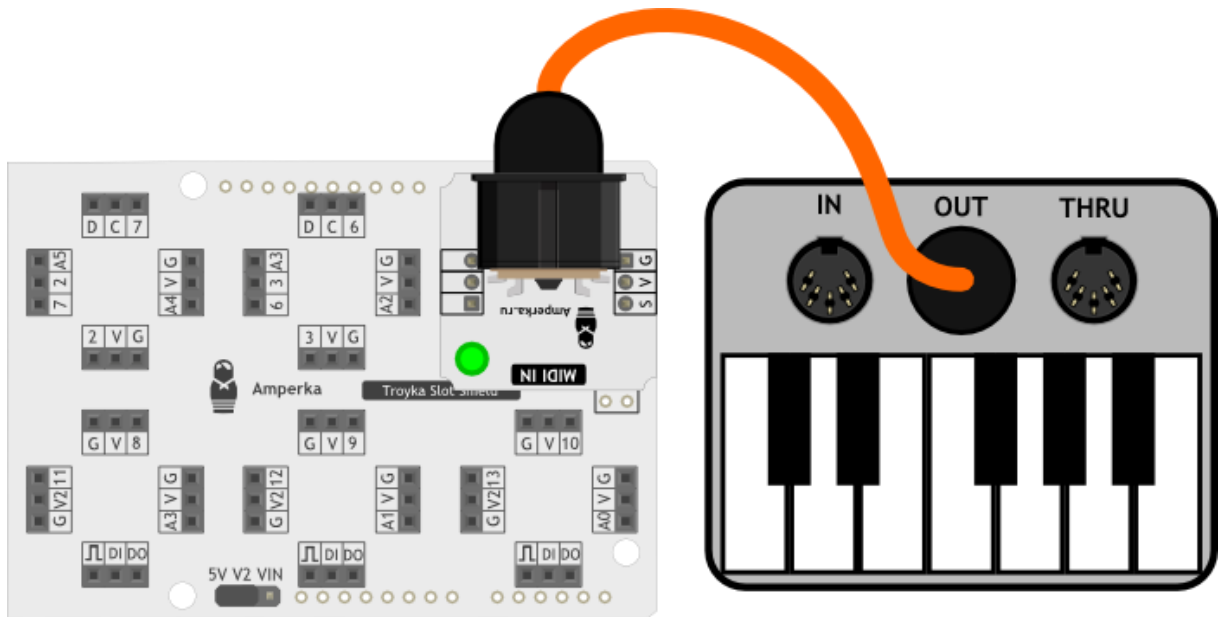
Стандарт MIDI подразумевает общение устройств между собой по асинхронному протоколу, аналогичному [UART](#). Отличие в том, что данные передаются не уровнем напряжения, а током. Поэтому мы будем подключать модули к Serial-порту управляющей платы.

MIDI IN

Простой MIDI-сниффер можно сделать, используя Тройка Slot Shield и Тройка-модуль [MIDI IN](#).

- Подключите Тройка-модуль MIDI IN к пину RX микроконтроллера;
- Загрузите скетч в Iskra Neo, откройте монитор порта;
- Подключите выход MIDI-клавиатуры к Тройка MIDI IN;
- Нажмите любую клавишу на MIDI-клавиатуре.

В мониторе порта вы увидите MIDI-сообщения.



Arduino Leonardo

[midiSniffer.ino](#)

```
//Мы используем плату с двумя Serial-портами. Это
//Arduino Leonardo или Iskra Neo

void setup() {

    //Настройка виртуального Serial-порта для подключения к компьютеру
    Serial.begin(115200);

    // Настройка Serial-порта для подключения к MIDI-клавиатуре
    Serial1.begin(31250);
}

void loop() {
    // Отправляем все сообщения с MIDI IN в монитор порта
    if (Serial1.available()) {
        Serial.println(Serial1.read());
    }
}
```

Iskra JS

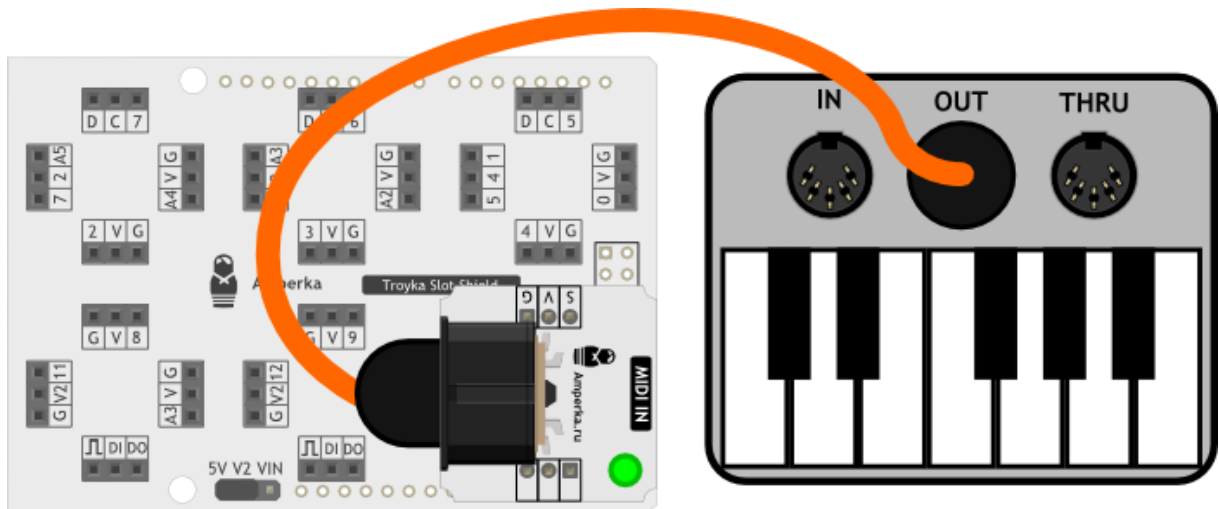
Тот же пример можно выполнить и на Iskra JS. Добавим расшифровку MIDI-сообщений: номер ноты, номер MIDI-канала и сила нажатия клавиши.

[midiSniffer.js](#)

```
var midi = require('Midi').setup(Serial3, 31250);
midi.on('noteOn', function(i) {
    console.log('Note: ' + i.note +
        ' on channel: ' + i.chan +
        ' with velocity: ' + i.velocity);
});
```

Arduino Uno

В Arduino Uno подключение к компьютеру происходит по единственному Serial-соединению. Чтобы иметь возможность наблюдать MIDI-команды в мониторе порта, изменим схему и создадим программный последовательный порт.

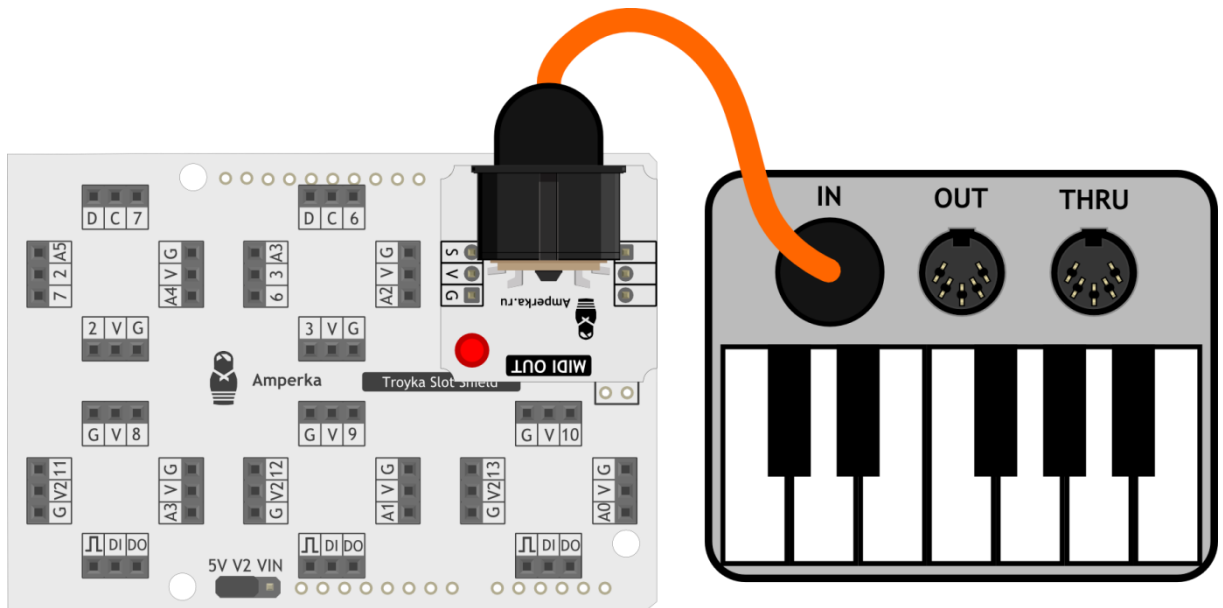


[midiSniffer.ino](#)

```
//Мы используем с единственным Serial-портом.  
//MIDI-вход находится на 10-м пине, на котором мы запустим  
SoftwareSerial  
SoftwareSerial mySerial(10, 11); // RX, TX  
  
void setup() {  
  //Настройка виртуального Serial-порта для подключения к компьютеру  
  Serial.begin(115200);  
  // Настройка SoftwareSerial-порта для подключения к MIDI-клавиатуре  
  mySerial(31250);  
}  
  
void loop() {  
  // Отправляем все сообщения с MIDI IN в монитор порта  
  if (mySerial.available()) {  
    Serial.println(mySerial.read());  
  }  
}
```

MIDI OUT

Тройка-модуль [MIDI OUT](#) подключается к пину TX микроконтроллера. Есть особенность при использовании модуля вместе с Troyka Slot Shield. В этом случае придётся подать питание на модуль через ножки микроконтроллера.



Мы используем Arduino Uno, поэтому в коде используется объект Serial. Если вы используете Arduino Leonardo или Iskra Neo, используйте Serial1.

[midiNotes.ino](#)

```

// Ножки микроконтроллера, которые мы используем для питания модуля:
#define VCC 5
#define GND 4

void setup() {
  // Устанавливаем скорость обмена
  Serial.begin(31250);

  // Организуем питание модуля через ножки микроконтроллера:
  pinMode(VCC, OUTPUT);
  pinMode(GND, OUTPUT);
  digitalWrite(VCC, HIGH);
  digitalWrite(GND, LOW);
}

void loop() {
  // Играем ноты от F#-0 (0x1E) до F#-5 (0x5A):
  for (int note = 0x1E; note < 0x5A; note++) {
    // Отправляем нажатие ноты на канале 1 (0x90), номер ноты (note),
    // средняя сила нажатия (0x45):
    noteOn(0x90, note, 0x45);
    delay(100);
    // Стандарт допускает отключение ноты (аналог отпускания клавиши
    // синтезатора)
    // при помощи отправки сообщения noteOn (0x90),
    // с номером той же ноты (note), и силой нажатия, равной нулю
    (0x00):
    noteOn(0x90, note, 0x00);

    // Пауза между нажатиями клавиш
    delay(100);
  }
}

// Отправка MIDI-ноты.
// Здесь отсутствует проверка валидности данных.
// MIDI-команда должна быть больше 127, а номер ноты и сила нажатия -
// меньше, либо равна 127

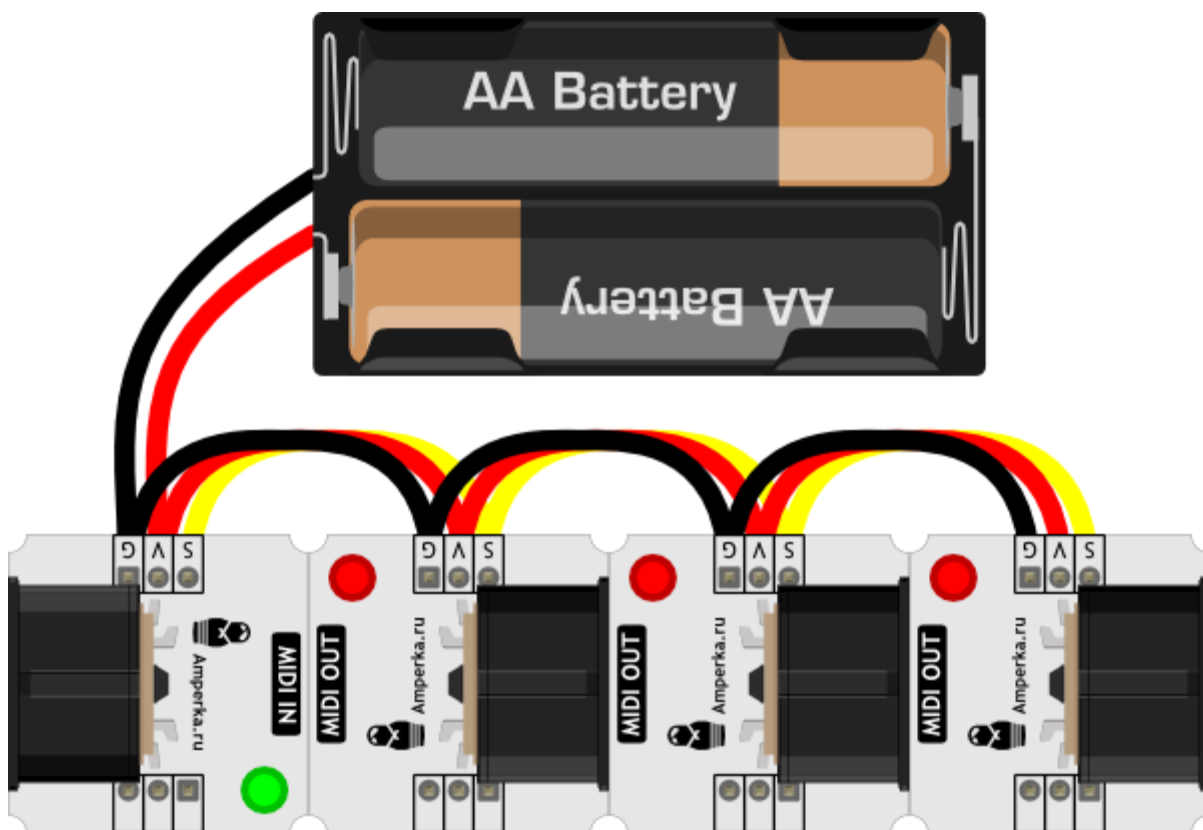
```

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

Проекты

MIDI-тройник

Некоторые синтезаторы обладают только одним или двумя MIDI-разъёмами, без MIDI THRU. Поэтому их не удаётся подключить классическим способом. Подключить же MIDI-входы параллельно друг другу не получится. В этом случае ток в токовой петле будет распределяться между входами непредсказуемым образом. Но проблему подключения таких синтезаторов можно решить, используя Тройка MIDI IN и горсть Тройка MIDI OUT. Для подключения модулей удобно использовать несколько [Тройка Pad 1×2](#) и [Power Bank](#).



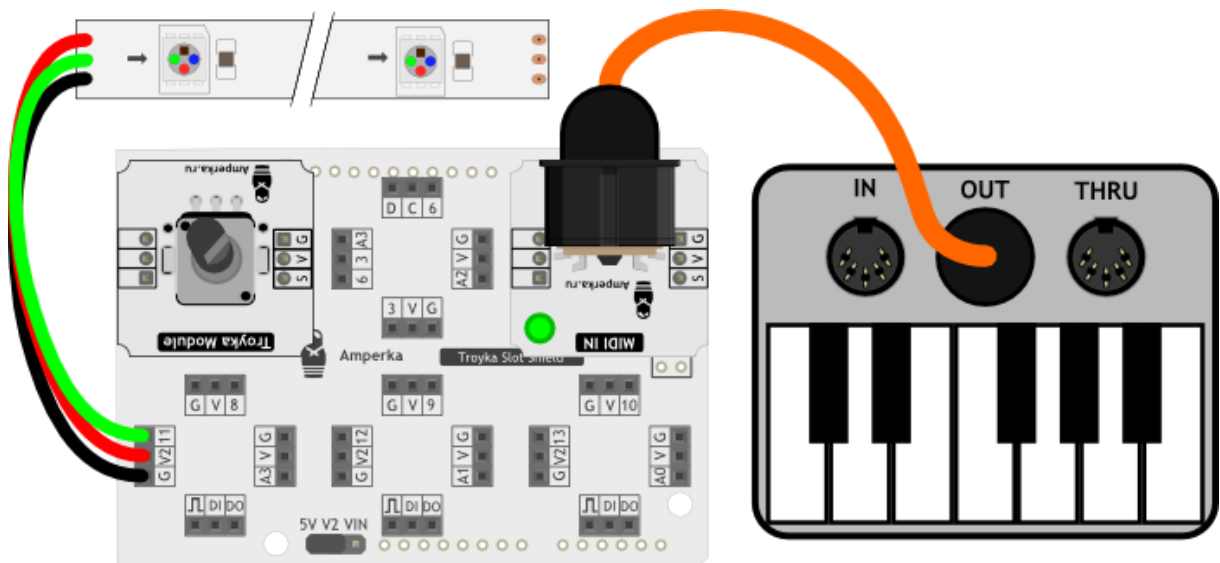
Индикатор нажатия клавиш

Используем цветную светодиодную ленту для индикации номера и силы нажатия клавиши синтезатора.

Понадобится:

- [Цветная адресуемая светодиодная лента](#)
- [Iskra Neo](#)
- [Тройка Slot Shield](#)
- [MIDI IN \(Тройка-модуль\)](#)

- [Потенциометр \(Тройка-модуль\)](#)
- [Блок питания на 12 В](#)



[midi-ledStrip.ino](#)

```
#include <MIDI.h>
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
#include <avr/power.h>
#endif

// Светодиодная лента подключена к пину 11
#define PIN 11

// Она состоит из 50 светодиодов
#define NUMPIXELS 50

// Создаём объект для управления лентой
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB +
NEO_KHZ800);

// Создаём объект для связи по MIDI через Serial1
MIDI_CREATE_INSTANCE(HardwareSerial, Serial1, myMidi);

// Вспомогательные переменные
int r, g, b, pitchPot;

// Функция, пересчитывающая оттенок в RGB.
// Подробности: http://wiki.amperka.ru/projects:christmastree
void f_HSV_to_RGB(int hue, int sat, int val)
{
  int base;
  if (sat == 0) {
    r = val;
    g = val;
    b = val;
  } else {
    base = ((255 - sat) * val) >> 8;
    switch (hue / 60) {
      case 0: {
        r = val;
        g = ((val - base) * hue) / 60 + base;
        b = base;
      }
    }
  }
}
```

```

        break;
    }
    case 1: {
        r = (((val - base) * (60 - (hue % 60))) / 60) + base;
        g = val;
        b = base;
        break;
    }
    case 2: {
        r = base;
        g = val;
        b = (((val - base) * (hue % 60)) / 60) + base;
        break;
    }
    case 3: {
        r = base;
        g = (((val - base) * (60 - (hue % 60))) / 60) + base;
        b = val;
        break;
    }
    case 4: {
        r = (((val - base) * (hue % 60)) / 60) + base;
        g = base;
        b = val;
        break;
    }
    case 5: {
        r = val;
        g = base;
        b = (((val - base) * (60 - (hue % 60))) / 60) + base;
        break;
    }
    case 6: {
        r = val;
        g = 0;
        b = base;
        break;
    }
}
}
}

// Обработчик события NoteOn. Эта функция будет вызвана автоматически
при
// при приёме сообщения NoteOn по MIDI-входу.
// Подробнее: http://arduinomidilib.fortyseveneffects.com/a00022.html
void handleNoteOn(byte channel, byte pitch, byte velocity)
{
    // Velocity имеет значение от 0 до 127
    // Чтобы не снижать диапазон значений оттенка цвета (0-255),
    // мы сдвигом влево умножаем силу нажатия на 2.
    byte doubleVelocity = velocity << 1;
    // Переводим оттенок в RGB
    f_HSV_to_RGB(doubleVelocity, 255, 255);

    // Номер ноты имеет значение от 0 до 127,
    // а светодиодов всего 50. Чтобы лента отображала ноты
    // в нужном интервале, подкрутим ленту под нужные ноты
    потенциометром.
    // Чтение значения с потенциометра происходит в loop();
    int activePitch = pitch - pitchPot;

    // Устанавливаем цвет светодиода в нужную позицию...

```

```

    pixels.setPixelColor(activePitch, pixels.Color(r, g, b));
    // ... и зажигаем ленту.
    pixels.show();
}
// Обработчик события NoteOff. Эта функция будет вызвана автоматически
при
// при приёме сообщения NoteOff по MIDI-входу.
// Подробнее: http://arduinomidilib.fortyseveneffects.com/a00022.html
void handleNoteOff(byte channel, byte pitch, byte velocity)
{
    // Гасим светодиод, который был зажжён от NoteOn
    int activePitch = pitch - pitchPot;
    pixels.setPixelColor(activePitch, pixels.Color(0, 0, 0));
    pixels.show();
}

void setup()
{
    // Инициализируем светодиодную ленту...
    pixels.begin();
    // ... и гасим все светодиоды
    pixels.show();

    // Присоединяем обработчик события NoteOn
    myMidi.setHandleNoteOn(handleNoteOn);

    // Присоединяем обработчик события NoteOff
    myMidi.setHandleNoteOff(handleNoteOff);

    // Инициализируем MIDI-интерфейс на прослушку всех
    // MIDI-каналов.
    myMidi.begin(MIDI_CHANNEL_OMNI);
}

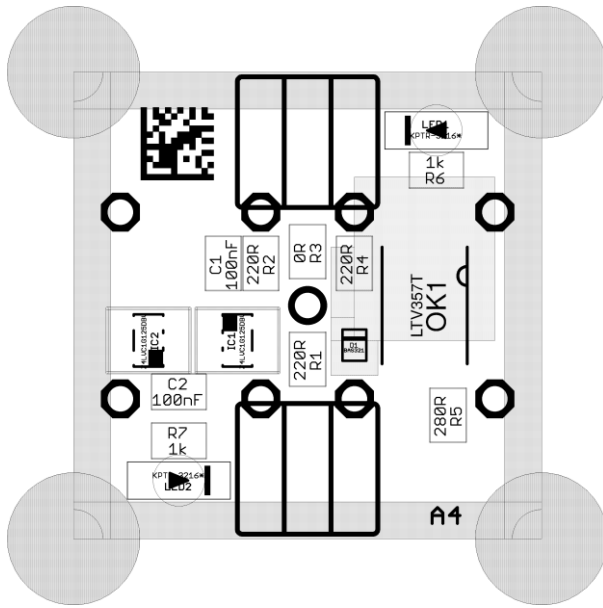
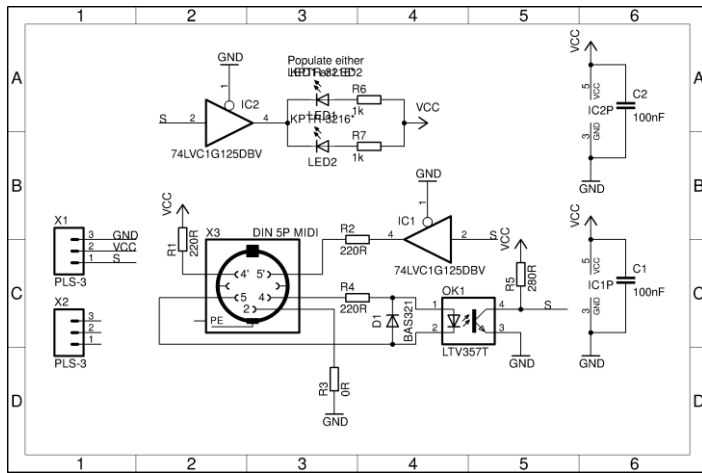
void loop()
{
    // Здесь мы просто вызываем MIDI.read,
    // остальное библиотека сделает сама
    myMidi.read();
    // Считаем значение с потенциометра.
    // Нам нужно значение в диапазоне (0-127),
    // его можно получить из диапазона (0-1024) сдвигом на 3 бита вправо
    pitchPot = analogRead(A4) >> 3;
}

```

Библиотеки

- [Библиотека для Arduino](#)
- [библиотека для IskraJS и Espruino](#)

Принципиальная и монтажная схемы



Характеристики

- Напряжение питания: 3,3–5 В
- Габариты: 25,4×25,4 мм